Vector Offset Operators

for

Deformable Organic Objects

Vassilios Hurmusiadis

A thesis submitted in partial fulfilment of the requirements of

Bournemouth University for the degree of Doctor of Philosophy

BOURNEMOUTH UNIVERSITY

December 1998

Dedicated

to Elena,

to my mother Irene and

to the memory of my father Harris.

"God exerts geometry"

Plato

Abstract

Many natural materials and most of living tissues exhibit complex deformable behaviours that may be characterised as organic. In computer animation, deformable organic material behaviour is needed for the development of characters and scenes based on living creatures and natural phenomena. This study addresses the problem of deformable organic material behaviour in computer animated objects. The focus of this study is concentrated on problems inherent in geometry based deformation techniques, such as non-intuitive interaction and difficulty in achieving realism. Further, the focus is concentrated on problems inherent in physically based deformation techniques, such as inefficiency and difficulty in enforcing spatial and temporal constraints. The main objective in this study is to find a general and efficient solution to interaction and animation of deformable 3D objects with natural organic material properties and constrainable behaviour. The solution must provide an interaction and animation framework suitable for the creation of animated deformable characters. An implementation of physical organic material properties such as plasticity, elasticity and viscoelasticity can provide the basis for an organic deformation model. An efficient approach to stress and strain control is introduced with a deformation tool named Vector Offset Operator. Stress / strain graphs control the elastoplastic behaviour of the model. Strain creep, stress relaxation and hysteresis graphs control the viscoelastic behaviour of the model. External forces may be applied using motion paths equipped with momentum / time graphs. Finally, spatial and temporal constraints are applied directly on vector operators. The suggested generic deformation tool introduces an intermediate layer between user interaction, deformation, elastoplastic and viscoelastic material behaviour and spatial and temporal constraints. This results in an efficient approach to deformation, frees object representation from deformation, facilitates the application of constraints and enables further development.

List of Contents

List of Illustrations

Chapter 1

Chapter 3

Chapter 5

Chapter 6

Chapter 7

Chapter 8

Chapter 9

Appendix A

List of Accompanying Material

This thesis is accompanied by a video tape containing several animation tests which demonstrate the various features of the suggested methodology. The format of the tape is VHS PAL and has a total duration of eight minutes and sixteen seconds. Time code has been printed on the tape in order to facilitate reference to specific scenes.

Title		Starting Time Code
Animation Test 1	Plastic VOODOO	10:04:05:00
Animation Test 2	Plastic VOODOO	10:04:19:00
Animation Test 3	Plastic VOODOO	10:04:32:00
Animation Test 4	Elastic VOODOO	10:04:46:00
Animation Test 5	Elastoplastic VOODOO	10:04:59:00
Animation Test 6	Elastoplastic VOODOO	10:05:12:00
Animation Test 7	Elastoplastic VOODOO	10:05:25:00
Animation Test 8	Viscoelastic VOODOO	10:05:38:00
Animation Test 9	Viscoelastic VOODOO	10:06:01:00
Animation Test 10	Viscoelastic VOODOO	10:06:14:00
Animation Test 11	Viscoelastic VOODOO	10:06:27:00
Animation Test 12	Plastic VOODOO with Motion Path Constraints	10:06:45:00
Animation Test 13	Plastic VOODOO with Articulated VOOs	10:06:58:00
Animation Test 14	Viscoelastic VOODOO with Motion Path Constraints and Articulated VOOs	10:07:11:00
Animation Test 15	Viscoelastic VOODOO with Plexus of VOOs	10:07:24:00
Animation Test 16	Walking Figure	10:07:37:00
Animation Test 17	Beach Ball	10:07:53:00

Animation Test 18	Giant Squid	10:08:24:00
Animation	Incredible Suckers	10:08:38:00
Animation	after birth (excerpts)	10:10:24:00
Animation	Gullaesthetics	10:11:16:00

Preface

My early attempts in producing computer animations based on living creature characters and natural phenomena brought me face to face with difficult animation problems. At first, I tried to ignore them. But soon, observing nature made me realise that these animation problems required an integrated physically based approach. One of the main reasons for undertaking this study was primarily to extend my understanding of the subject of organic modelling and animation. Another reason was to attempt to find simple solutions to organic animation problems and to provide versatile animation tools suitable for creating life like animated characters.

Acknowledgements

I would like to gratefully acknowledge the help, guidance and technical, philosophical and psychological support I received from my supervisors Professor Peter Comninos and Professor John Vince. Further, I wish to thank all the staff and students at the National Centre for Computer Animation for their helpful suggestions and constructive criticism. Last but not least, I thank Elena for her enthusiasm, encouragement, optimism and coffee making that kept me awake all these years.

Author's Declaration

I would like to draw attention to two research papers I have written and published in the last two years. The first paper, titled: "Vector Offset Operators for Dynamic Control of Deformable Objects", was published in EUROGRAPHICS UK Chapter Proceedings (Hurmusiadis 1996) and was based on the contents of chapter 5 and chapter 8. The second paper, titled: "Fast Implementation of Viscoelastic Creep Suitable for Manipulation of Organic Objects", was published in EDUGRAPHICS & COMPUGRAPHICS Combined Proceedings (Hurmusiadis 1997) and was based on the contents of chapter 6 and chapter 7.

1

Introduction

1.1 3D Computer Graphics

Computer Graphics (CG) has grown from being a specialist's tool in the 1980s, to become a general tool of everyday use in the 1990s. This is clearly indicated by the sheer amount of attention computer graphics receives from diverse areas such as education, medicine, the manufacturing, advertising, entertainment, military and other industries and the world of art. The notion of the three dimensional (3D) CG concept revolves around the process of looking through a virtual camera at 3D geometric data (see figure 1.1).

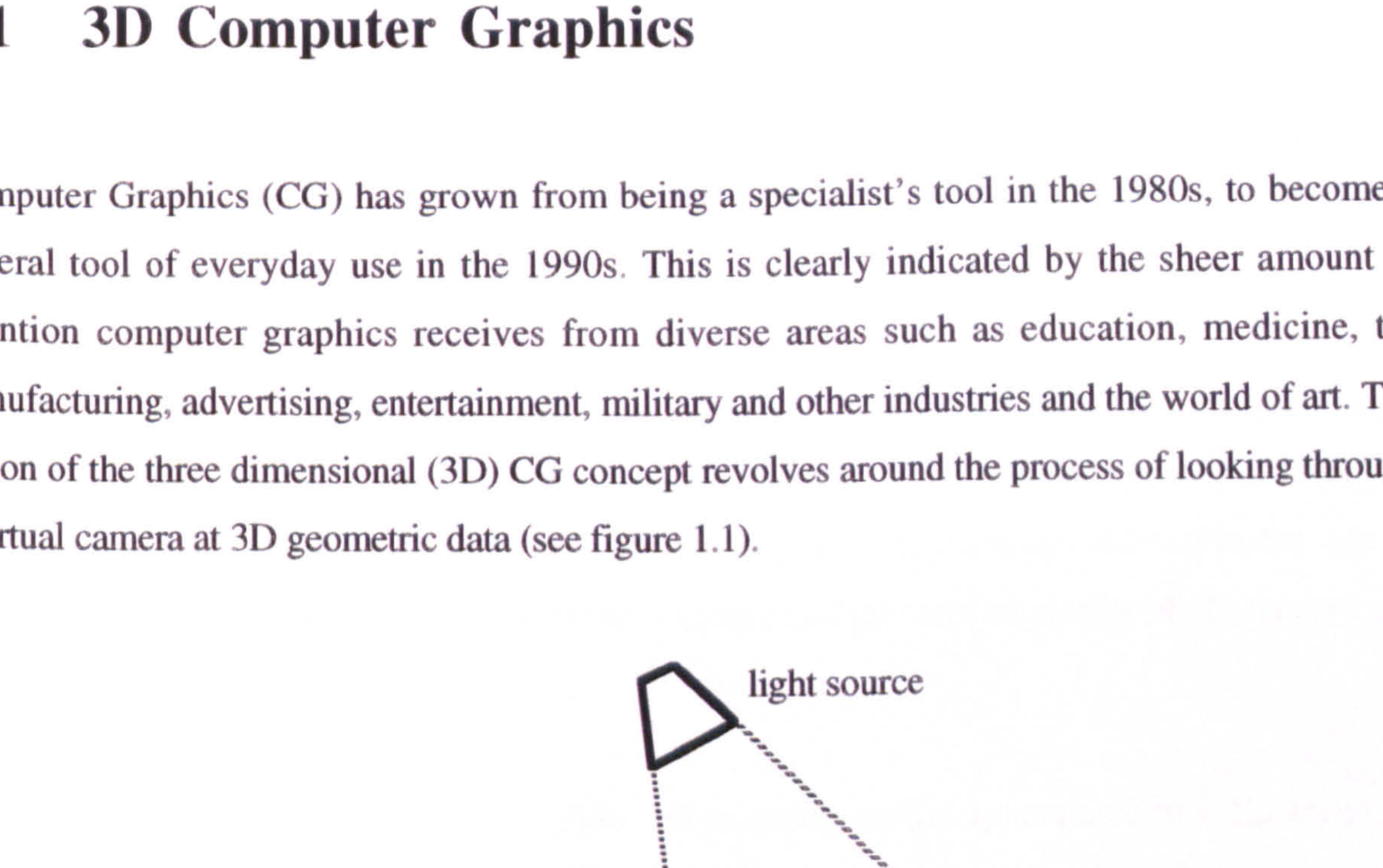


Figure 1.1 3D Computer Graphics.

The basic ingredients of the 3D CG concept consist of geometric data, a description of a virtual camera and the process of image synthesis.

Detailed geometric representations of real objects can be used to synthesise accurate models suitable for computer aided design and manufacturing (CAD, CAM) or medical applications. Alternatively, realistic or stylised representations of real or imaginary objects may be incorporated in virtual worlds which are designed for the production of animated sequences for entertainment applications. Motion inside virtual worlds is necessary for the visualisation of dynamic phenomena in science, medicine and engineering as well as for the creation of narrative in animation.

The process of image synthesis focuses on 2D image generation based on 3D data. Computer generated images can be photorealistic or non-photorealistic. Photorealism is indispensable for advertising and the production of special effects for film and video. Non-photorealistic synthetic images are equally important for scientific and other applications.

The rate of image generation may range from a fraction of a second to several hours per image. Real-time screen update is of crucial importance to certain applications encountered in areas like simulation, training and entertainment. In applications such as computer animation, the rate of image generation can become very slow since image quality and complexity of 3D scenery are of higher priority.

The ability to interact with virtual worlds and to influence the appearance of a 3D scene, at computer run-time, is crucial for the existence of a range of new applications. Interaction with virtual worlds, through direct manipulation of camera, objects and motion parameters, has become the key design feature for a new generation of applications that appear under the name "virtual reality".

Overall, the core CG concept can be extended and adapted to a powerful and versatile medium, to serve a great variety of purposes. It is becoming increasingly evident that 3D CG, apart from serving as a tool capable of providing new solutions to old problems, it has also spawned a plethora of new solutions to new problems. During the recent decades, 3D CG technology,

consisting of theory, specially implemented software and customised graphics hardware has been researched, developed and incorporated into industrial applications. On the other hand, this technology has also triggered the creation of new application areas such as the digital entertainment industry.

There are two major applications of 3D CG technology: Computer Animation (CA) and Interactive CG (ICG). This distinction is based on the level of interaction and the rate of image generation offered by an application. However, the majority of CG applications incorporate both animation and interactive manipulation to a certain extend.

1.1.1 3D Computer Animation

Although static pictures are a good means of communicating information, dynamically varying pictures are even better. Animation has traditionally been used as an art form, as well as a medium for information visualisation, for more than seven decades now. During this time animation has developed in a variety of forms depending on the materials and techniques used for the creation of picture sequences: cell animation, clay model animation, puppet animation, etc. The common element, though, in all forms of animation has always been a "frame-by-frame" production of the entire film. A sequence of pictures (frames) displayed on a screen at a rate of twelve frames per second or more can convey smooth motion. This happens because the human brain has the ability to blend the impression of two images which are being rapidly flashed in succession.

Model animation based on clay models, puppets, etc., is a traditional animation technique conceptually very close to 3D computer animation. In a conventional animation production process a miniature scene consisting of foreground characters and background objects is being slightly modified and a picture is then taken using a film or video stop-frame camera. The process is being repeated for the total number of frames required.

In computer animation a similar production process has been simulated using computer and 3D

CG technology. Shape, form, colour, light, motion and camera have been replaced by CG models which simulate their physical counterparts and in certain aspects enhance their capabilities. Shape, form and motion, for instance, have been implemented using 3D geometry. 3D CG objects can be moved in space, can be made to obey physical laws, can be dynamically deformed, can be forced to explode into numerous particles and still be reusable. Generally, 3D CG objects may be manipulated with a far greater degree of freedom than physical objects. Colour, material texture, light and camera effects have also been simulated using computer models which can produce photorealistic images capable of satisfying the demands of an experienced professional. Overall, a sophisticated computer animation system offers much more than a conventional animation production setup and it approaches in its functionality a complete "mini" film production studio.

The influence of computer animation in industry, education and art is surveyed in the following paragraphs.

1.1.1.1 Computer Animation in Entertainment Industry

In the last two decades, the entertainment industry has adopted computer animation as a mainstream content creation tool. This has fuelled an ever increasing appetite for visual effects in film, video and location based entertainment production.

One major and ever growing application area for computer animation is the production of special effects for film and video. Special effects have been traditionally based on scale models, animatronics, in-studio fire, smoke, explosion effects, etc. For some time now, computer graphics researchers have proved that most of these effects may be simulated and enhanced in flexibility, reusability and quality using computer animation models. However, until recently, the difficulty of implementation of such models and high production costs have kept digital effects away from mainstream film production. There have been, though, some pioneering exceptions in films such as "Star Wars - A New Hope" (1976), "TRON" (1982), etc.

In the 1990s it is, gradually, getting harder and harder to point out a major film production which does not involve digital effects in some sort. The digital effects industry has boomed in the last five years with a steep growth rate which is foreseen to continue for some time yet. The 3D computer animation effects produced for Steven Spielberg's feature film "Jurassic Park" (1993) by Industrial Light and Magic are being considered a landmark in the short history of digital effects. It was the first production ever, that involved long and complex close-up scenes of digitally generated realistic living creatures interacting with real actors. Another major production of great importance was the computer animated feature film "Toy Story" (1995) which was produced as a joint venture between Pixar and Walt Disney Feature Animation. The true contribution of this production is that it was the first full length feature film entirely produced using 3D computer animation.

Other recent feature films with high digital content are "Titanic" (1997), "Lost World" (1997), "Dragonheart" (1996), "Twister" (1996), "Casper" (1995), "Babe" (1996) and many more. Most of the recent Walt Disney Feature Animation productions involve some 3D computer animated scenes, as in "Hercules" (1997), "The Hunchback of Notre-Dame" (1995), "Pocahontas" (1994), "The Lion King" (1993), "Aladdin" (1992), "The Beauty and the Beast" (1991), etc.

A more widespread application area for computer animation effects is the advertising industry. The majority of commercial and television ident productions involve short computer animated inserts combined with live action or other filming techniques. There are numerous examples of state of the art computer animated short films produced by companies which are dedicated to the special needs of the advertising industry.

Computer animation is also used in the production of children's television programmes such as cartoons as in "Homer3" from Pacific Data Images, Inc. (1995) and in the production of music videos such as in Herbie Hancock's "Dis Is Da Drum" from Metrolight Studios, Inc. (1995), Peter Gabriel's "Steam" from Real World Productions (1993), etc.

Location Based Entertainment (LBE), also known as simulated rides, forms yet another significant section of the digital entertainment industry which attracts computer animation

expertise. In recent years, there have been several LBE productions which involved visuals almost entirely based on computer animation. A characteristic example is probably James Cameron's "Terminator 2 3D" (1995) from Digital Domain, located at Universal Studios Florida. This large scale location based venue involves high definition 65mm 3D composited film projection of digital and live footage combined with live stage action and other effects. Other computer generated simulated rides are "Krakken" from ExMachina (1995), "Seafari" from Rhythm & Hues Studios (1993), "The Volcano Mine Ride" (1995) and "Devil's Mine Ride" (1992) from New Wave Entertainment, etc.

1.1.1.2 Computer Animation in Simulation and Visualisation

Computer animated simulation is a form of automated animation. A simulation system generates motion based on some kind of input from the user ahead of time. The input usually consists of objectives and rules for making decisions and it is generally less specific than with animation. The user knows less about the resulting motion. The job of the simulator is to predict what would happen under certain circumstances and inform the user of the results.

Often, scientific simulation relies on computer animation for visual exploration of information data sets derived by various methods. An area where computer animation plays a vital role is the visualisation of data derived from space exploration. Satellite images help generate computer models and animation sequences which depict planet landscapes, solar activity, etc.

Frequently, the engineering practice incorporates computer animation for the visualisation of 3D space and form as in architecture and product design. Another application of computer animation is in the study of fluid mechanics, stress/strain analysis, etc.

A relatively new application area for computer animation is medicine, where CG models are used for the visualisation of physiological and pathological functions of the human body.

Sometimes simulation is capable of generating aesthetically interesting animation, such as in physical simulation (gravity, elasticity, etc.) or in simulation of natural phenomena (water,

wind, tornadoes, etc.). Properly constrained simulation models of physical laws can generate automated animations of moving objects which are made to behave very much like real objects. This in a way leads computer animation closer to live action filming and further away from traditional stop-frame animation techniques. For this reason physical simulation models are becoming very popular amongst animators and gradually find their way into commercial computer animation systems.

1.1.1.3 Computer Animation in Education and Art

Computer animation can be used for the visualisation of complex conceptual or physical, inherently three dimensional models, in a way suitable for education. From primary education to university level computer animation is in use as a tool for the development of visual course work material. Geometry, physics, astronomy and biology are only some of the subjects that have greatly benefited from the use of computer generated explanatory animations.

Traditional forms of animation have always had a strong presence as an art medium. Recently though, the ever decreasing cost of computer hardware and the widespread availability of 3D and 2D computer animation systems have allowed artists to gain access to this technology and adopt computer animation as yet another art medium. Computer animation, with its immense power to depict imaginary worlds, has offered the digital artist unprecedented opportunity and challenge. This has lead to the formation of a growing digital art community from around the world. This is evidenced by the plethora of computer animation art festivals and gallery shows worldwide. Nowadays, most major CG conference events are accompanied by art shows and galleries such as SIGGRAPH (USA), EUROGRAPHICS (Europe), IMAGINA (France), etc. On the other hand, long established and prestigious film and animation festivals have incorporated a computer animation category as in London Film Festival (UK), Ottawa Animation Festival (Canada), Annecy Animation Festival (France), Hiroshima Animation Festival (Japan), etc. Prix Ars Electronica (Austria) and the International Symposium of Electronic Arts are two events which specialise in digital arts and devote a major section to computer animation.

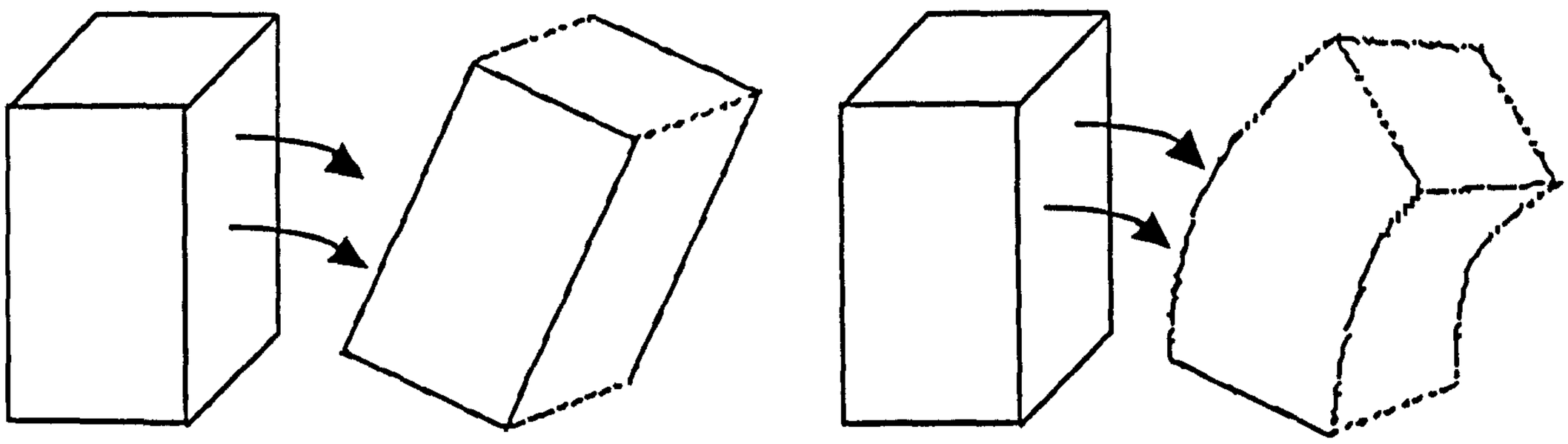
1.1.2 Interactive 3D Computer Graphics

The 1980s and the 1990s saw the expansion of high performance CG technology into new domains. Real-time interactive manipulation of 3D virtual worlds has been one of them. Interactive CG enables extensive and fast user computer interaction. This enhances the user's ability to understand data, perceive trends, visualise real or imaginary objects and to create virtual worlds that one can explore from arbitrary points of view. Direct geometric manipulation enables interactive scene composition through the specification of position and shape of geometric objects. Other parameters which influence the overall appearance of a scene, such as lighting, colour and texture may also be interactively manipulated. A system which enables direct manipulation of 3D data and is capable of real or even near real-time rate of screen updates forms a powerful framework suitable for a great variety of application areas in science, engineering, entertainment, education and art.

The interactive entertainment industry has expanded its domain by incorporating 3D CG technology and high performance customised hardware. This has lead to some very interesting results in the diversity and the quality of computer entertainment products. The "CINEMATRIX Kinoetic Adventure" is a remarkable example in interactive entertainment, which involves mass audience participation. The audience interacts with a virtual underwater world, projected in stereoscopic high definition television. It was presented at the SIGGRAPH '94 Electronic theatre by CINEMATRIX Interactive Entertainment Systems, Inc. and Evans & Sutherland Computer Corporation. Interactive CG technology is also present in television production where graphics characters are being generated and combined with live action in real time. Another major application area for interactive CG is the training industry. Interactive simulated environments are in use for flight training, army combat training, etc. Scientific simulations also rely on interactive CG for exploration of various dynamic phenomena, such as control of airflow in aircraft and car design, simulated surgery in medicine, etc.

1.2 Deformable Modelling in Computer Graphics

In nature, an object may be characterised as rigid or deformable according to its response to external forces. Rigid objects tend to retain their shape and form intact under a certain amount of external stress, whereas their position and orientation changes dynamically following Newtonian physics. Deformable materials, on the other hand, are generally submissive to relatively small external stresses and as a result their shape changes over time. Deformation is a dynamic process which depends on properties based on the molecular structure of a material.



a. b.

Figure 1.2 a. Rigid object, b. deformable object.

In computer graphics, objects may also be characterised as rigid or deformable (see figure 1.2). Rigid objects may be arbitrarily rotated and translated, but they always retain their original size and shape. The entire geometric description of a rigid object is subjected to the same amount of transformation. However, in deformable objects different parts of a geometric description may, at any given time, undergo different degrees of transformation. Thus, deformable objects may appear to bend, stretch, squash, shear and or twist. Generally, the shape of a deformable object appears to flow in time and space.

Recent trends in computer character animation lead towards more realistic animated characters. More often, such characters are based on living creatures which consist of soft organic parts. This is particularly evident in special effects for film and television where the majority of computer graphic scenes involve living creatures or natural phenomena. Hence, there is an

increasing demand for sophisticated deformation techniques which are efficient, easy to constrain and are capable of generating realistic deformable behaviour.

Simple deformations may develop inside the body of an object by applying standard animation techniques in conjunction with geometric deformation techniques. Shape metamorphosis, procedural deformation and animated free form deformation are all deformation techniques purely based on geometry. This geometric deformation approach provides powerful and expressive animation mechanisms, it is based on relatively simple theory and it is capable of achieving interactive rates on relatively low-cost hardware. However, interaction with such models is usually non-intuitive as it involves a lot of artificial elements, such as control points and other geometric parameters. In animations produced using the metamorphosis technique, realism is entirely dependent on the animator's skill. On the other hand, realism achieved by procedural models is entirely dependent on programming skill.

A more fundamental approach to deformable modelling involves physical simulation of force application onto real material objects. Stress and the effected strain can be simulated using conceptual force application tools which act directly on CG objects. Such models may be enriched with the physical properties of mass and elasticity and may act inside simulated environments with gravity, collision constraints, friction, etc. Physically based deformable modelling is based on complex theory of continuum mechanics which involves mathematics and numerical analysis. Such models are difficult to implement and often result in inefficient code that needs high-end specialised graphics workstations to achieve even remotely interactive rates. Considerable difficulties are also encountered in enforcing space and time constraints upon these models. It can become very complicated to predict their state in space and time, let alone to impose constraints. However, physical models can offer intuitive control because they react to external user applied forces as well as environmental forces like gravity and wind. The great advantage of such models is that they form inherently realistic simulations of natural behaviours. Thus, realism in animation produced with these models comes for free.

1.3 Focus of this Research

The focus of this research is concentrated on problems inherent in geometry based deformation techniques and on problems inherent in physically based deformation techniques (see figure 1.3).

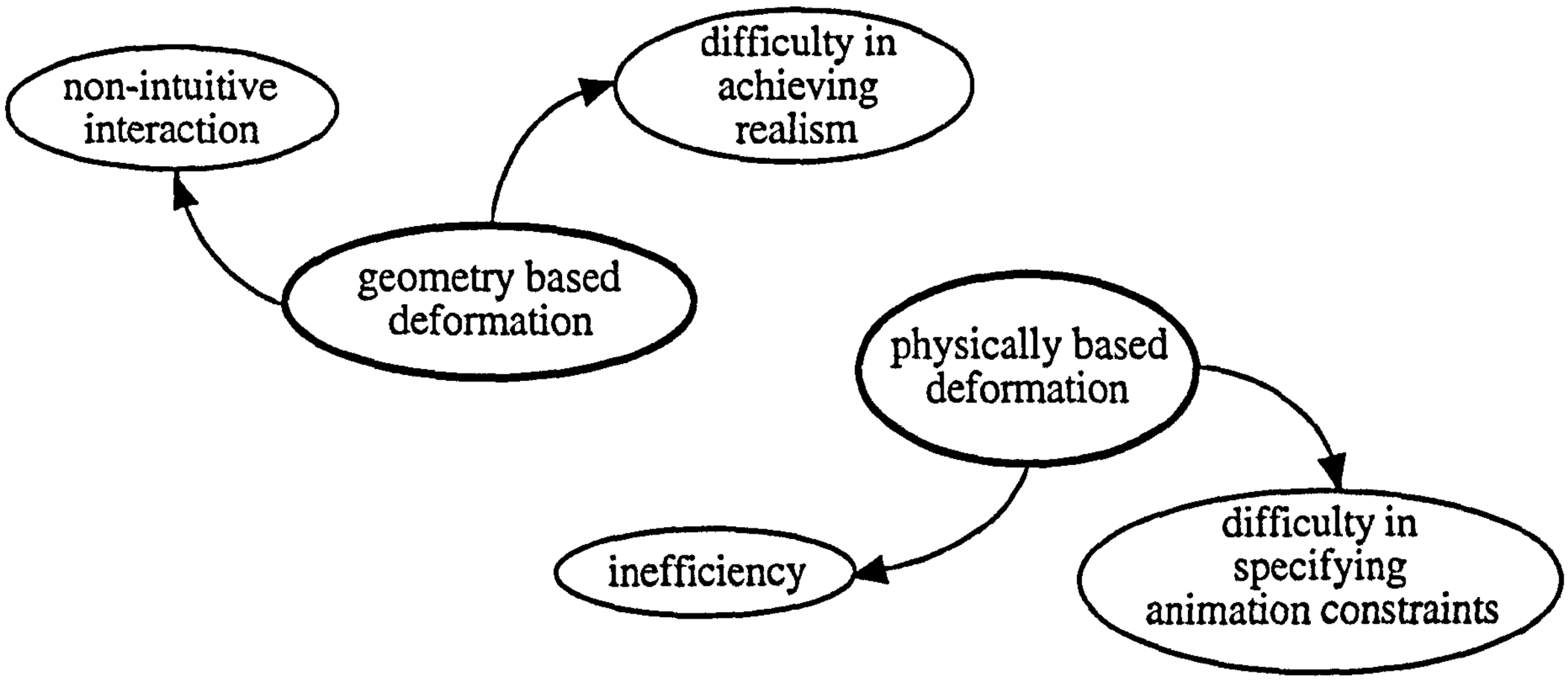


Figure 1.3 Problems in geometry based and physically based deformation techniques.

Problems inherent in geometry based deformation techniques are non-intuitive interaction and difficulty in achieving realism. Problems inherent in physically based techniques are inefficiency and difficulty in enforcing spatial and temporal animation constraints.

In computer animation, realistic organic material behaviour is necessary for the development of organic characters based on living creatures and natural phenomena. Therefore, an implementation of physical organic material properties such as plasticity, elasticity and viscoelasticity can form the basis for an organic deformation model suitable for animation. Animated deformable behaviour ought to be self-activated so animators may concentrate more on animation problems at a higher level such as story and plot. Additionally, a deformation model must be fast so as to allow comfortable interactive manipulation, as well as efficient animation testing and rendering. The model must also provide animation constraining techniques in order to help achieve fine control over the spatial and temporal behaviour of a deformable object.

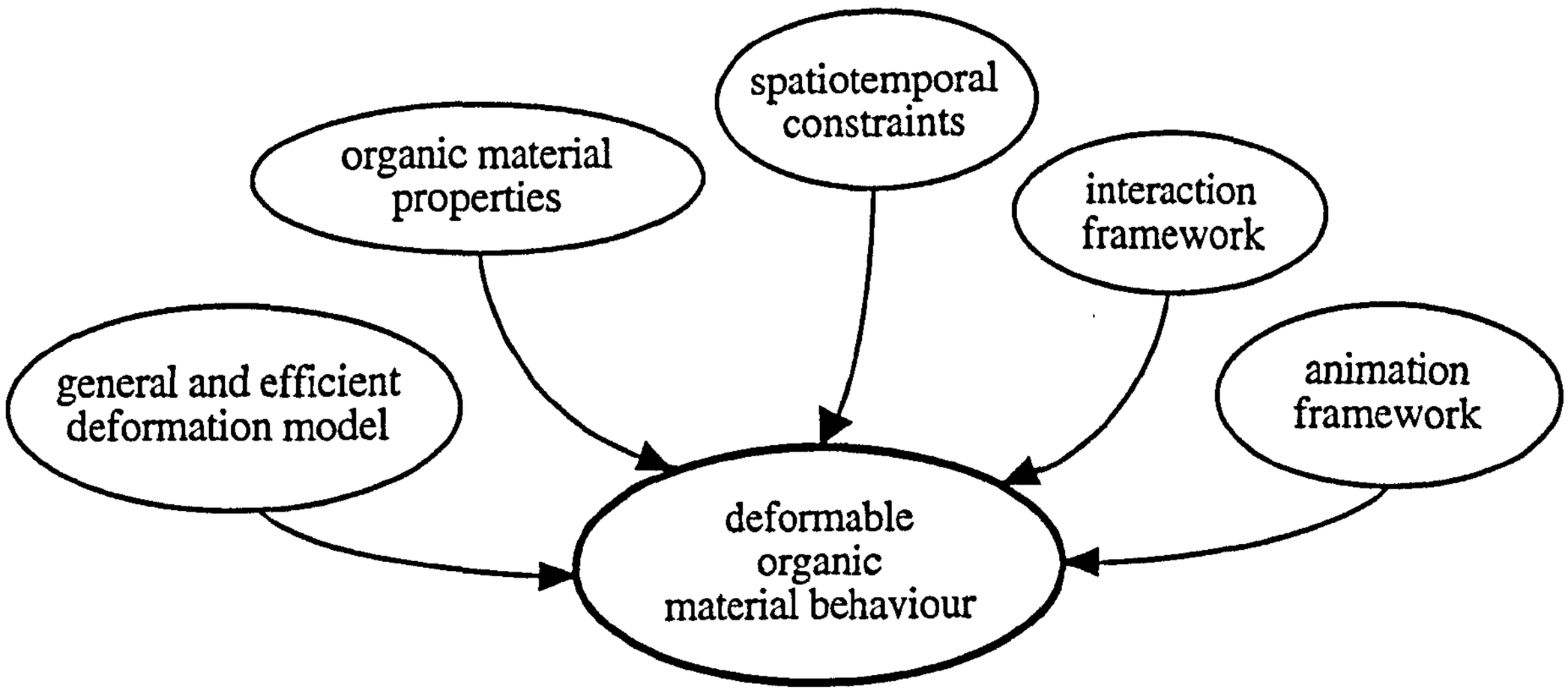


Figure 1.4 The objectives of this thesis.

The objectives of this study are (see figure 1.4):

- to develop a general and efficient deformation model,
- to equip the deformation model with organic material properties,
- to enable application of spatial and temporal constraints on organic objects,
- to enable interaction with organically deformable objects,
- to enable animation of organically deformable objects.

The main objective in this study is to find an efficient solution to interaction and animation of deformable 3D objects with natural organic material properties and visually correct behaviour. The solution must provide an animation and interaction framework based on direct control and dynamic motion path constraints suitable for the creation of animated deformable characters.

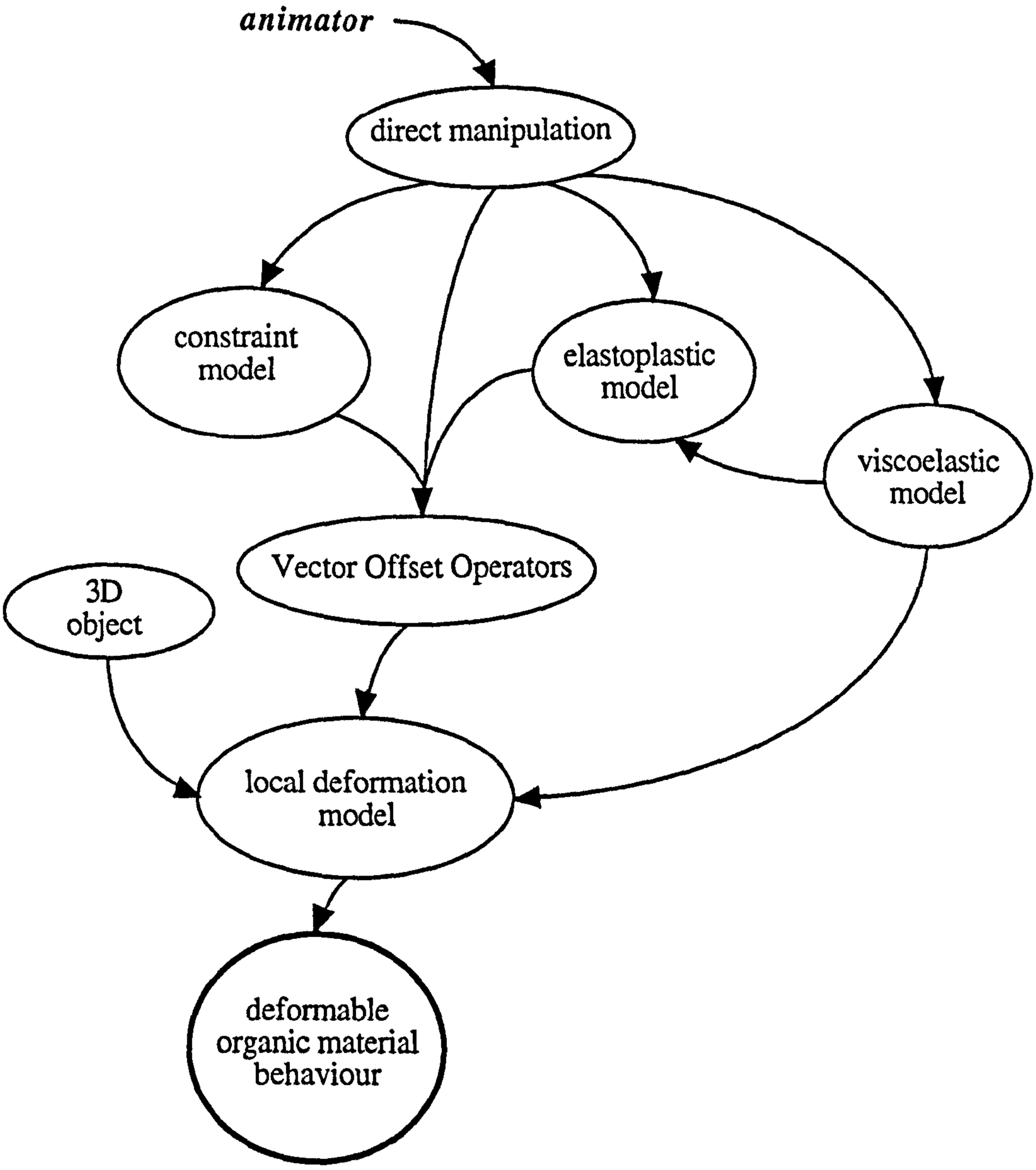


Figure 1.5 The structure of the conceptual model.

The diagram in figure 1.5 presents the structure of the conceptual model. The suggested structure is based upon a core deformation model. A Vector Offset Operator (VOO) is a strain application tool capable of specifying efficient local deformation in 3D space (see figure 1.6). Any 3D data that enter the deformed space of a VOO, experience an amount of local deformation. VOOs may be manipulated directly or through a dynamic constraint model. An elastoplastic model intervenes to influence the state of VOOs in space and time. A viscoelastic model intervenes to directly influence the state of the local deformation vector. The viscoelastic model may also affect the state of the elastoplastic model.

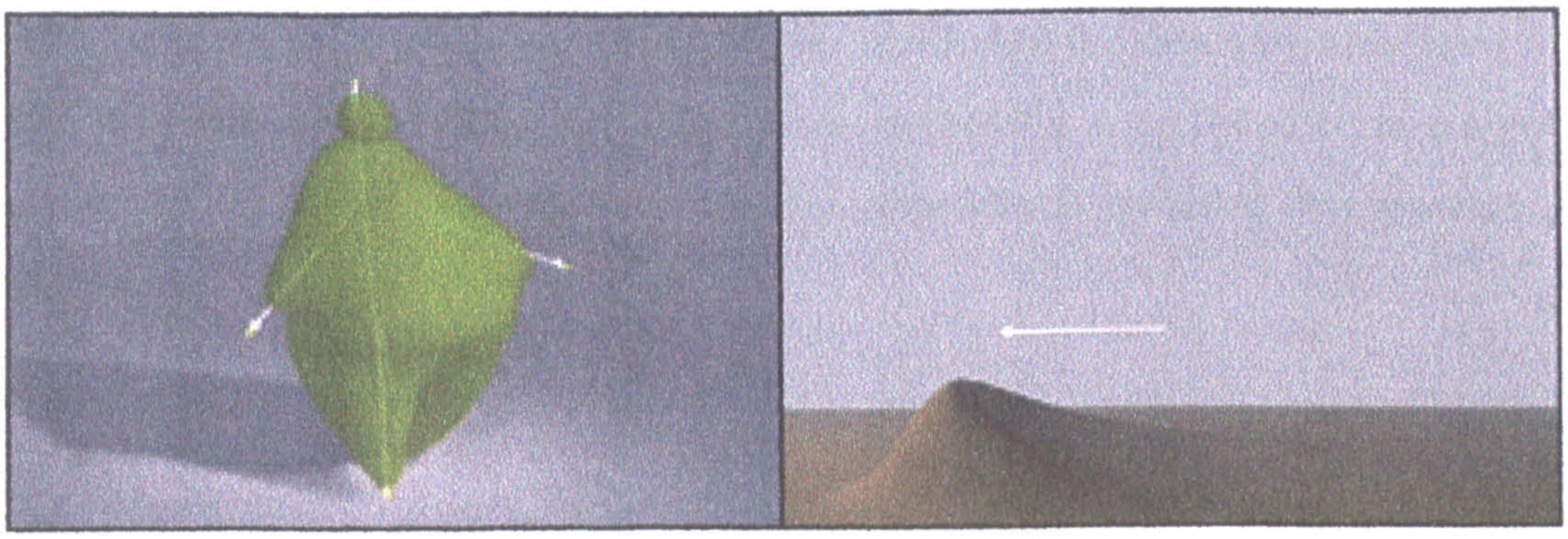


Figure 1.6 A VOO is a general purpose strain application tool.

The suggested model offers direct control upon VOOs, dynamic constraints and the elastoplastic and viscoelastic properties of a material. The merging of a local deformation model with the physical properties of elasticity, plasticity and viscoelasticity leads to a new generic tool which is capable of achieving efficient, realistic, yet controllable organic material behaviour. The suggested generic tool is based on a hybrid approach that leads to the implementation of a fast, general, reusable, extendible and independent of geometric representation model.

1.4 Outline of the Thesis

This thesis consists of ten chapters and one appendix. The first chapter has provided a general introduction to computer graphics followed by a presentation of the focus and the objectives of this research. The second chapter attempted a survey of existing literature on deformable modelling techniques. The third chapter reviewed some background theory of elasticity, plasticity and viscoelasticity and explored characteristic examples of experimental data of organic material behaviour. The fourth chapter reviewed certain 3D object representation techniques.The fifth chapter presented an overview of local deformations, introduced the concept of Vector Offset Operators and described the plastic deformation model. The sixth chapter described the elastoplastic deformation model. The seventh chapter extended the model to incorporate the property of viscoelasticity. The eighth chapter further extended the model to incorporate spatial and temporal constraints. The ninth chapter discussed applications and

results of the presented model. The tenth chapter concluded the thesis by presenting a thesis summary, conclusions and plans for future work. Appendix A included certain algorithms encountered in the thesis and appendix. Finally, appendix B included two research papers published by the candidate.

2

A Survey of 3D Modelling of Deformable Bodies

2.1 Introduction

Animation has always been a much loved entertainment and art medium. Increasingly through the years, animation has become a powerful tool in the visual communication of complex ideas to vast audiences. During the last decade, an exponential increase in computing machinery power and the rapid development of digital technology has acted as a catalyst in the bringing together of the elements of animation and computer. Contributing to this, there has been a tremendous research activity, worldwide, on the subject of computer animation. A great deal of such research has been particularly fruitful in a wide spectrum of animation practices. As a result a plethora of professional applications adopted computer animation as a means of visual output. The body of knowledge in computer animation theory and practice has outstandingly expanded and diversified in a relatively short period, which renders any attempt to thoroughly review all of the related literature a difficult task.

Computer animation literature may be categorised using an abstraction of the approach adopted for the animation, as an answer to the question: how to animate? This could lead to categories such as research in guiding-level animation, key-frame animation, animator-level animation, task-level animation, physically based animation or an integration of these (Zeltzer 1988). In guiding-level, a detailed script specifically controls every step of the animation. In key-frame animation, a set of animator defined key-states determine the advancement of the animation. In animator-level, the animator may direct and stage an animated character. In task-level, the animator may describe a set of rules to be obeyed and a set of goals to be achieved by the characters. Finally, in physically based animation certain laws of physics govern the state of the animated characters throughout the animation.

Further, the categorisation may proceed via the representation of the objective of animation, as an answer to the question: what to animate? This could help categorise research into the animation of natural or "unnatural" phenomena, mathematical, physical or scientific concepts, rigid bodies, flexible bodies, articulated structures, etc., or combinations thereof.

Finally, research in computer animation may be categorised by the implementation framework, as an answer to the question: which animation tool? Here, there are two distinct categories in computer graphics research in general: the realm of two dimensions (2D) and the realm of three dimensions (3D). Additionally, the relation between animation production and time may also play a vital role in the categorisation of animation research with regard to the implementation framework. This may lead to research in real or near real-time interactive computer graphics and in off-line computer animation frameworks.

The focus of the attempted literature review is on physically based animation of flexible bodies in a 3D computer animation framework. For the sake of completeness, some of the approaches taken by previous researchers in the area of geometric deformation modelling will be briefly examined. Then the survey will proceed with the examination of existing research on physically based modelling of flexible bodies. The standard elastic and inelastic models will be presented, as well as several variations of them. Then, some major application areas of flexible modelling like cloth simulation, biomechanical modelling and virtual actors will be reviewed. A survey of several interaction techniques with flexible bodies in real time will also be included followed by

a survey of certain techniques for constraining flexible bodies in space and time. A review of several commercial computer animation systems will complete this survey.

2.2 Geometric Deformation

A geometric deformation, as opposed to a physically based deformation, is a mathematical function which explicitly modifies the global coordinates of object elements in space. Methods for describing and modifying shape in 3D space are of central concern to computer graphics modelling. Geometry has proven useful for modelling stationary objects whose shapes do not change over time. On the other hand, geometric transformations combined with simple space parameter constraints can lead to a variety of geometric deformation methods. Several researchers have promoted a sculpting metaphor for geometric modelling, noting that it is a natural and familiar mode of thought for a designer. Very early, Parent (1977) discusses a computer graphics sculptor's studio for defining polygonal objects. At the same time, Brewer and Anderson (1977) describe a planar shaping tool for manipulating sculptured surfaces which are made of parametric surface patches.

2.2.1 Barr's Transformations

Barr (1984) suggested that deformations of solid primitives are important and highly intuitive operations which ease the control and rendering of large families of 3D geometric shapes. He presented a simple yet very effective way to treat solids as topological clay which may be bent, twisted, tapered, compressed or expanded. The normal or tangent vector of an arbitrarily deformed surface can be calculated from the surface normal or tangent of the undeformed surface and an affine transformation matrix. Thus, by applying a gradient of affine transformations over the whole or part of a solid, the shape of the solid is deformed smoothly. Bends and twists can be achieved with rotations, tapers with scalings and compressions or expansions with combinations of translations and scalings. Any arbitrary combination and succession of transformations may be applied to a solid, resulting in complex deformations.

The beauty of Barr's approach lies in its simplicity, efficiency and general applicability. Many researchers thereafter will refer to "Barr's Transformations" or "Global Deformations" as a standard upon which they will build more complex deformation models.

2.2.2 Free Form Deformation

A new technique for deforming solid geometric models in a free form manner was presented by (Sederberg and Parry 1986). Their Free Form Deformation (FFD) method may by described by the physical analogy of a parallelepiped of clear, flexible plastic lattice inside which is embedded an object or several objects. The object deforms along with the plastic lattice that surrounds it. The mathematical scheme in FFDs is based on trivariate Bernstein polynomials. The deformations are defined by such parametric functions whose values are determined by the location of control points. A common FFD user interface is based on direct manipulation of control points which are arranged inside the volume of the bounding parallelepiped. The interface to FFDs reflects the underlying mathematics of the modelling method and can be confusing and not at all intuitive for an inexperienced animator. The FFD approach is general and powerful, it is capable of deforming locally or globally and preserving the volume of the deformed solid.

An advancement to the FFD approach, the Extended Free Form Deformation method (EFFD) was introduced by (Coquillart 90). With EFFDs, the animator configures the initial lattice of control points to the approximate shape of the independent deformation, instead of starting with a parallelepiped of control points. EFFDs are quite effective for creating impressions, reliefs and other simple deformations. However, the animator must know the general shape of the deformation before starting to model.

A further extension to the FFD method was presented in (Hsu *et al.* 1992). Their method allows an animator to control a free form deformation of an object by directly manipulating the object. This offers better control of the deformation and a more intuitive interface. They developed a method of converting an animator action of the form "move this object point there"

into a set of FFD control point key positions that will effect this action. This technique can be used to mimic "Play-Doh®" material behaviour, which is a synthetic material with nearly perfect plastic behaviour.

A new technique was proposed by Chang and Rockwood (1994) that deforms an object by repeatedly applying affine transformations in space. The object warps along an animator defined curve. This approach reduces the definition of the free form deformation from a crowded set of control points to a single Bézier curve and a few affine maps.

Finally, completing the review of geometric deformations, in *"Advanced Animation and Rendering Techniques"* Watt and Watt (1992) include a section on soft object animation. It covers deformation of surface patches, nonlinear global deformations (Barr's transformations) and free form deformations (FFD and EFFD). Further, there is an introduction to "factor curves" that are functions of space and time. Factor curves can assist the modification of parameters, that control a global deformation, over time and space.

It must be noted that Barr's Transformations and the various FFD methods offer efficient and easy to control, purely geometric deformations. None of the methods introduced in the previous paragraph involve physics. As a consequence, any realism in the resulting shape of the deformation, depends entirely on the skill of the animator. However, both of these methods are considered as standard in flexible body modelling and can be found implemented in a variety of disguised forms in most of commercial modelling and animation systems.

2.3 Animation of Geometric Deformation

Several methods have been suggested for the extension of basic geometric deformation techniques towards animation. Some of these methods are based on space and time interpolation between key shapes which have been interactively designed by the user. Another approach to animated geometric deformation is based on procedural models which specify algorithmic deformation over time.

2.3.1 3D Metamorphosis

3D metamorphosis is the spatial and temporal transition of one shape into another. The careful selection of an initial and a final key shape and the specification of the transition technique in space and time can result in an interesting animated deformation of a shape. Additionally, a set of key shapes and interpolation functions that involve space and time parameters can lead to complex animated deformations. However, the realism in the resulting animation depends entirely on the skills of the animator. In a basic metamorphosis technique, also known as inbetweening, each element of the topology of the initial key shape is transformed to the respective element of the topology of the final key shape. This technique assumes that both initial and final objects share identical polygon connectivity and only differ in shape.

A simple 3D shape metamorphosis technique, suitable for polyhedral objects of different polygon connectivites, was presented in (Kent *et al.* 1992). Their algorithm merges the topological structure of a pair of 3D polyhedral models into a common vertex / edge / face network. Then the transformation from one object to another is achieved by interpolating between corresponding vertex positions.

More advanced metamorphosis techniques can be based on direct interpolation of deformation parameters in space and time. For instance, a metamorphosis technique based on Barr's Transformations need only interpolate between initial and final transformation parameter sets. Another technique based on FFDs need only interpolate between initial and final control lattice positions.

Although the present review clearly concerns 3D techniques, an exception will be made to introduce the work of Beier and Neely (1992). They presented a new technique for the metamorphosis of one digital image to another. 2D morphing is the combination of an image warping with a cross dissolve between image elements. Their morphing technique was based upon fields of influence surrounding 2D control primitives. The use of 2D control primitives

offers an intuitive interface to the process of describing which features of one image are to morph to which features of another. This technique is directly related to a much earlier 2D skeleton technique suggested by Burtnyk and Wein (1976). This 2D morphing technique was extended to a 3D metamorphosis technique by (Lerios *et al.* 1995). They applied their technique to 3D volume based representations of objects by adopting a similar approach to the one described in (Beier and Neely 1992). Their method enables fine animator control and achieves realistic looking intermediate objects by applying feature based 3D volume warping.

Rossignac and Kaul (1994) also presented a metamorphosis technique for 3D shapes. They introduced the concept of Bézier Interpolating Polyhedron (BIP), which provides a graphics representation of a deforming object, mathematically formulated, as a point describing a Bézier curve in space. BIPs are composed of Animated GRaphic ELements (AGRELs), which are faces with constant orientation, but with parametrised vertices represented by Bézier curves.

Coquillart and Jançene (1991) extended Extended FFDs towards animation. An animated FFD (AFFD) can be defined by an initial and a final control lattice. Intermediate lattices are then obtained by interpolating over time between the initial and final lattices. At each time step a key lattice deforms the 3D objects thus creating a deformed inbetweened instance of the objects.

Bechmann and Dubreuil (1992) introduced a new technique to animate deformable objects through time and space. They introduced the concept of interactive manipulation and animation design of 4D deformable objects, with the fourth dimension referring to time. The state of a deforming element at a specific time is conveyed by its position in space, velocity and acceleration vectors.

2.3.2 Procedural Deformation

A procedural deformation model is generally based on an algorithm that relates space to time. It is usually implemented using a guided animation system and written in a scripting animation language like CGAL (Comninos 1980, 1986). An interesting approach to the animation of

geometric deformations may be based on procedural algorithmic models that mimic certain physical phenomena.

A procedural approach was adopted in (Fournier and Reeves 1986) and in (Peachey 1986), who both attempted to simulate ocean waves. Fournier and Reeves based their wave model on Gerstner's oceanographic model, according to which particles of water are assumed to describe circular or elliptical stationary orbits. This model can easily produce realistic wave shapes which are varied according to the parameters of the orbits. The surface of the ocean floor affects the refraction and the breaking of waves on the shore. Later, Ts'o and Barsky (1987) extended the wave model towards wave tracing using beta-splines, reflective and refractive texture mapping to achieve realistic simulations of water surface.

A procedural deformation model, similar to the one introduced by Fournier and Reeves, can be applied as a more general deformation method in an attempt to mimic cyclic movement in nature, other than water. Some animal movement is cyclic (periodic) in its nature, such as bird flapping, undulatory movement in fish or aquatic mammals, swimming of micro-organisms, etc. A generalised form of a wave model was applied in bird flapping by (Hurmusiadis 1991).

The present review, has so far considered conventional methods for computer graphics modelling and animation of flexible bodies. These methods are geometric and kinematic in their representation. The models produced by such methods, may be thought of as passive because they do not interact with each other or with external forces. In most methods, the motion of the deforming parts of the flexible bodies is conveyed by geometric interpolation between key positions or key parameters. In some methods, though, time is incorporated in the equations and thus velocity and acceleration vectors play a vital role in shaping the deformation and producing more realistic motion.

2.4 Physically Based Deformation

Physically based modelling is by its nature a cross-disciplinary field involving elements of Newtonian physics, applied mathematics, numerical analysis, software engineering and computer graphics (Barzel 1992). Computer animation has recently been able to achieve astonishing realism by exploiting concepts from physics. In physically based animation numerical procedures compute the motion of models in simulated worlds in accordance with Newtonian mechanics. Active models which are based on principles of physics, react to applied forces, to constraints, to ambient media or to impenetrable obstacles as one would expect real physical objects to react. Deformable models approximate non-rigid material behaviour under diverse environmental conditions. Such models must be capable of exhibiting a wide variety of natural behaviours like stretchability, flexibility, resiliency, fragility, fluid like behaviour, etc. Implementations of active deformable models are based on laws of continuum mechanics, which are usually expressed in the form of dynamic differential equations. There is a variety of numerical algorithms that may be used to solve systems of differential equations of motion. Unavoidably, active deformable models are computationally more expensive than geometric and free form deformation models. Moreover, physically based models, in general, present the additional difficulty of being constrained in space and time. However, the outcome of implementing physically based deformable models is manifold, as they offer a higher level of realism in animation and are suitable for a variety of interactive applications.

In *"Physically Based Modelling for Computer Graphics"* Barzel (1992) presents a structured approach towards modelling incorporating applied mathematics, numerical analysis and computational physics into computer graphics. A clear separation between the abstraction, the representation and the implementation of a model is the key to the development of reusable, modular and correct models with high degrees of complexity. According to Barzel a hybrid approach between the top-down and bottom-up techniques starts with a top level fixed goal. Then building upon knowledge of low-level techniques that are available, the top and the bottom are linked, creating intermediate structures. This so called annealing design technique is suitable for research on physically based modelling for computer graphics. Reviews on physically based deformable models can be found in *"Computer Graphics: Principles and*

Practice" (Foley *et al.* 1990), *"3D Computer Animation"* (Vince 1992) and in *"Virtual Reality Systems"* (Vince 1995). Various physically based deformable model formulations and applications are surveyed in the following sections.

2.4.1 Elastic Modelling

Elastic objects "spring" back to their original shape, as soon as external forces are removed. Elastic behaviour is visually very interesting and can be used to simulate a wide spectrum of physical objects such as synthetic elastic objects, cloth, organic living tissue, etc.

In (Terzopoulos *et al.* 1987) elasticity theory from classical mechanics is employed to construct differential equations that model the behaviour of non-rigid curves, surfaces and solids as a function of time. They suggested a new class of active elastically deformable models, based on simplifications of elasticity theory (Hooke's Law) (Fung 1965). These models can be made to behave like string, rubber, cloth, paper, flexible metal, etc. Their method incorporates physical properties such as mass and damping. The motion of a typical mass element inside an elastic model is conveyed by Lagrange's equation of motion:

$$\frac{\partial}{\partial t}(\mu \frac{\partial \mathbf{r}}{\partial t}) + \gamma \frac{\partial \mathbf{r}}{\partial t} + \frac{\delta \mathrm{E}(\mathbf{r})}{\delta t} = \mathrm{f}(\mathbf{r}, t) \qquad [2.1]$$

where, $\mathbf{r}$ is the position of the element at time t, μ is the mass of the element, γ is the damping factor, $\mathrm{f}(\mathbf{r},t)$ is the sum of external forces applied on the element and $\mathrm{E}(\mathbf{r})$ is the potential energy of elasticity. Their simulation proceeds by numerically solving the partial differential equations that govern the evolving shape of the deformable object (Press *et al.* 1986) (Zienkiewicz 1977). The shape of a body is determined by the Euclidian distances between nearby points. As the body is forced to deform, these distances change. The potential energy of the elastic deformation should be zero when the model is in its natural state. This energy should grow larger as the model gets increasingly deformed, away from its natural state. So, this energy of elasticity generates internal forces that react to external forces and constantly attempt

to restore the rest shape of the object. The simulation of Newtonian mechanics and the numerical solution that this method is based upon, lead to a substantial computational overhead.

Witkin and Welch (1990) described a fast method for physically based animation of flexible bodies. Their approach treats motion control of non-rigid bodies with attachment constraints. To achieve non-rigid body behaviour they apply global deformations (Barr 1984) on objects with mass.Their formulation can produce objects that deform in a geometrically simple but physically correct way. Attachment constraints are achieved by calculating constraint forces that counter any external or internal forces that tend to pull objects apart. Motion control is achieved by attaching a point on an arbitrary trajectory. In other words this method can produce articulated puppets whose parts are made of jelly like material, with specified points under full control of the animator. The puppet moves with correct passive dynamics by moving these control points as functions of time. The key advantages in this approach, in contrast to the previous Newtonian mechanics simulation, is simplicity and efficiency.

Pentland and Williams (1989) described a system which uses polynomial deformation mappings to couple a vibration mode representation of object dynamics together with volumetric models of object geometry. Miller (1988) has incorporated elastic muscle actuators into discrete deformable objects to synthesise self-locomoting snakes and worms. Finally, Kass & Miller (1990) used an elastically deformable surface with some additional constraints in order to model the surfaces of pools of water.

2.4.2 Cloth Modelling

Modelling the behaviour of cloth has become an important subject in the computer animation research community and has attracted the attention of many researchers in the recent years. Cloth models have been used for the design of garments in the fashion industry, to dress virtual actors, to animate flags, curtains, tablecloths, etc. Such items exhibit a complex behaviour, particularly when they come in contact with rigid bodies or under the influence of environmental forces like wind and gravity.

Several authors have explored modelling and animation of cloth using physically based elastic surface models. Aono (1990) developed a physics based non-rigid object model for the behaviour of cloth and its simulation with given forces and boundary conditions. His work involved elasticity theory and incorporated physical properties such as viscoelasticity, anisotropy and inhomogeneity. By doing so, he managed to implement physical cloth parameters like Poisson's ratio, Lame's constant and Young's modulus which help simulate the behaviour of real cloth objects with distinct fibre characteristics.

Kunii and Gotoda (1990) incorporated both kinetic and geometric properties for generating garment wrinkles. In (Lafleur *et al.* 1991) there is a discussion on the problem of detecting collisions of very flexible objects like cloth with almost rigid bodies like human limbs. They use a very thin force field around the rigid obstacle surface to avoid collision with the soft cloth surface. Further, in (Carignan *et al.* 1992) a cloth modelling technique is adapted to garment design for virtual actors. 3D cloth models are attached around a synthetic actor and when the actor moves in space, external forces due to gravity, wind and collision response, are being applied to the cloth. Finally, Yang *et al.* (1992) customised the previous cloth modelling technique to develop a CAD tool for garment design for use in the fashion industry. Their tool enables the visual examination of a 3D garment on a moving human body, before the garment is actually manufactured.

2.4.3 Inelastic Modelling

Inelastic models attempt to simulate plastic, viscoelastic and fragile natural materials. A perfectly plastic material yields under any stress and the resulting strain (deformation) is permanent (Mendelson 1968). There are no elastic energies to restore to its initial shape. A characteristic plastic material is modelling clay. A viscoelastic material, under certain stresses, flows like a viscous fluid and returns back to its initial shape when the stress is removed (Christensen 1982). A typical synthetic viscoelastic material is silicon putty. Fragile materials develop discontinuities in the structure when the external stress exceeds a maximum value (Fung 1965). In practice, all natural or synthetic materials are fragile and do break under a

certain stress. Most natural or synthetic materials, under various stress conditions, exhibit a complex behaviour which is a mixture of elasticity, plasticity, viscoelasticity and fracture.

Terzopoulos and Fleischer (1988) focused on the simulation of inelastically deformable models. They attempted to simulate all three common inelastic behaviours: viscoelasticity, plasticity and fracture. They extended the numerical algorithms that they developed for the simulation of their elastic model (Terzopoulos *et al.* 1987), to incorporate the inelastic deformation units. To implement viscoelasticity they suggested that this behaviour may be approximated with the combined use of an elastic unit (a spring) and a viscous unit (a dash pot). The elastic unit obeys Hooke's law:

$$f = ke \qquad [2.2]$$

where f is the external force, k is the spring constant and e is the deformation. In the viscous unit the rate of deformation propagation is proportional to the external force:

$$f = \nu \frac{de}{dt} \qquad [2.3]$$

where f is the external force, ν is the viscosity constant and (de/dt) is the rate of deformation propagation. The plasticity unit was implemented by introducing the magnitude of permanent deformation under an external stress application. Finally, fracture was based on the introduction of discontinuities in the object topology when the deformation exceeded a certain limit. The Terzopoulos *et al.* method results in large systems of simultaneous ordinary differential equations which are numerically solved. This approach to elastic and inelastic behaviour offers a thorough simulation framework which is more suitable for visualisation rather than animation. The computational overhead could become a prohibitive factor for the implementation of this method in fast, interactive animation design applications.

2.4.4 Particle Based Inelastic Modelling

Several researchers approached non-rigid modelling using physically based particle systems. Commonly, each particle is considered as a concentration of mass that is connected to other mass particles via elastic springs or viscoelastic dash pots.

Miller and Pearce (1989) implemented a dynamic particle system for modelling fluids and powders. Their particle system involved pair-wise inter-particle forces. A particle system was also proposed in (Wu *et al.* 1995) for the purpose of animating a deformable surface as realistically as using a continuous system. They used approximate methods to calculate the stress and strain on an elastoplastic model whose behaviour was described by a stress/strain graph. They applied their method on animating skin with wrinkles, cloth, paper, rubber and other surfaces.

Holton and Alexander (1995) extended the notion of the particle based deformable system to a cellular structure. In their approach, a cell consists of a tetrahedral element with four mass points, one on each vertex and six connecting springs, one on each edge. They used such cells as the minimum volumetric unit to construct volumetric 3D objects. They avoided the conventional spring and mass differential equation approach (Terzopoulos *et al.* 1987). Their algorithm incorporates an elasticity constant and a viscosity constant in a linear equation of motion which conveys the state of each mass point inside the cell structure. Their model is derived from the Maxwell mechanical viscoelastic model which is based on a spring and a dashpot connected in series. Thus, they achieved a fast implementation of an approximation of viscoelastic behaviour. The real advantage of their approach lies with the fact that the cellular structure lends itself to efficient fracture techniques.

A more complex inelastic behaviour can be achieved by thermoelastic models. Such models feature non-rigid dynamics governed by Lagrangian equations of motion and conductive heat transfer. In its solid state, the model is composed of thermoelastic elements which interconnect particles (much like springs in elastic models). As the temperature increases the stiffness of a thermoelastic element decreases and the unit fuses when the temperature exceeds the melting

point. In the molten state the model behaves like a viscous fluid. An interesting thermoelastic model approach was introduced by (Tonnesen 1991). His thermal particle technique simulated deformable solids that can melt into fluids when heated and resolidify when cooled. Later, Szeliski and Tonnesen (1992) developed a surface modelling technique based on oriented particles. They built deformable surfaces which enabled interactive moulding. Their surfaces become more mouldable as temperature increases. What is even more interesting is that new particles are generated wherever the moulding process results in a more complex feature. The surfaces can be cut and joined at arbitrary locations.

2.4.5 Biomechanical Modelling

Biomechanics aim to explain the mechanics of life (Fung 1990). From molecules to organisms to vertebrate animals, every living creature, to a certain extend, obeys some laws of mechanics. Furthermore, biomechanics is a tool for the design of real devices or conceptual computer models that simulate certain aspects of life. An area of great importance and interest in computer animation is that of living creature modelling. In recent years, there has been a growing interest in the development of idealised biomechanical models of human or other animal bodies. These models may be used to produce realistic animations for the entertainment industry. Accurate models may be used in experimental simulation for medical surgery and in artificial limb prosthesis. Biomechanical models of the human figure, may also be used in the car industry as computer simulated crash dummies or in fashion design as synthetic fashion models.

Chen and Zeltzer (1992) adopted a radical approach by simulating a biomechanical model of a human arm. Since muscle is the fundamental motor that drives all animal motion, it seems more natural and more challenging to simulate a muscle that drives the skeleton rather than a motion specified skeleton that drives the muscle. So they proposed a method where changes in shape of moving character figures will be accurately reproduced by simulating the muscle action and resulting forces that propel these figures. The geometry and underlying material properties of their muscle model are captured using finite element analysis. They incorporated an existing biomechanical model of muscle function to apply non-linear forces to the finite element mesh

nodes. This method is suitable for animators wishing to create anatomically based character animations. The accuracy of the simulation, however, makes the muscle model also suitable for biomechanical study of muscle function.

Perhaps the most intriguing and most complex part of a living creature is, undoubtedly, the face. Our everyday familiarity with faces and facial expression helps us communicate with other people and animals but also makes us very critical towards any attempt to synthetically reconstruct a face. One of the hardest challenges is to develop computational models of the face capable of synthesising facial expression quickly and convincingly. There is a growing need for such models, in computer character animation, plastic surgery, teleconferencing, etc. Waters (1987 and 1988) developed a face model which included two types of muscle: linear muscles that pull and sphincter muscles that squeeze. He used a simple spring and mass model for the skin and muscles. Each muscle has a zone of influence and the effect of a particular muscle is reduced as a function of radial distance from the muscle point. His muscles are independent of the underlying bone structure which makes the muscle model independent of specific face topology.

Terzopoulos and Waters (1990) proposed a physics based facial modelling approach which was based on Waters' earlier work on facial expressions (Waters 1987). Their model is capable of animating facial expressions by contracting synthetic muscles. The muscles are embedded in an anatomically motivated model of skin, composed of spring-mass layers. The physical simulation propagates the muscle forces which deform the skin to produce facial expressions. This model is computationally efficient and produces realistic expressions.

Kalra *et al.* (1991) suggested a methodology for specifying facial animation, based on a multi-layered approach. Each successive layer defines entities from a more abstract point of view, starting with simple muscle actions connected to phonemes and working up through words, sentences, expressions and emotions. The highest level layer consists of a scripting language that allows the manipulation of these entities, ensuring synchronisation of the eye motion with emotions and word flow of a sentence.

Lee, Terzopoulos and Waters (1995) have proposed an accurate biomechanical model for facial

animation of individuals. They developed an algorithm to estimate the skull structure from acquired range data and prevent the synthetic facial skin from penetrating the skull. Their model includes teeth, an articulated neck and jaw and animated eyes and eyelids. Finally, to conclude the survey on facial modelling, Koch *et al.* (1996) developed a model for the simulation of facial surgery. Their goal was to predict the facial shape after standard procedures in facial surgery. Several other researchers have attempted to develop specialised models that focus on other important parts of a human model like hair (Anjyo *et al.* 1992), (Daldegan *et al.* 1993) and human hand grasping (Gourret *et al.* 1989).

2.4.6 Virtual Actors

An integration of several physically based computer animation models with behavioural models and a layered approach lead some researchers to the development of complete virtual actor systems suitable for computer animation and much more. Such actors may have a realistic human look or a character animation look. The implementations are usually based on an amalgamation of deformable models and rigid articulated structure models. Some virtual actor models can be interactively controlled by an animator. Some, however, more advanced models are equipped with artificial intelligence, which renders them autonomous, exhibiting a behaviour of their own and obeying specific orders.

Magnenat-Thalmann and Thalmann (1987) introduced the concept of Joint dependent Local Deformation (JLD) operators, which are specified local deformations (a.k.a. Barr's Transformations, see §2.2.1) depending upon the nature of the joints. These JLD operators control the evolution of the surfaces which are being mapped around an articulated skeleton. Using JLDs and an underlying articulated structure of limbs and joints, they constructed synthetic actor models of Marilyn Monroe and Humphrey Bogart, which featured in the computer animation film "Rendez Vous à Montréal". They developed the Human Factory system, which allows animators to build realistic models of individual human beings. A detailed presentation of the JLD method may be found in *"Making Them Move"* (Badler *et al.* 1991).

Chadwick *et al.* (1989) proposed an integrated, layered methodology for creating and animating computer generated characters. Their Critter system is based on a four layers approach. The first layer consists of the definition of a motion foundation which consists of an articulated armature. Then they overlay a deformable layer of muscle and fatty tissue which is conveyed by FFDs. A surface description covers the previous two layers, to simulate skin, fur or cloth. Finally, the fourth layer consists of motion specification which defines the behaviour of the armature. This approach attempts a uniform character representation which can be tuned and tweaked by the animator to meet precise expressive qualities. Their emphasis is placed on constructing the character, so that less effort is needed in actually scripting an animation. However, they do not attempt to replace animators by algorithmic models, but intend to provide them with a more powerful medium for artistic creativity. A detailed presentation of the Critter system may be found in *"Advanced Animation and Rendering Techniques"* (Watt and Watt 1992).

Turner and Thalmann (1993) also described a model for creating 3D animated characters. This model is based on a simulated, elastically deformable skin surface which is wrapped around a traditional kinematic articulated figure. The skin surface is free to slide along the underlying structure and is finally constrained by reaction forces, which push the surface out and spring forces which pull the surface in. By tuning the parameters of the physically based model, a variety of surface shapes and behaviours can be obtained, such as skin wrinkles and folds around the joints. Later, in (Turner 1995), the elastic skin model was further developed into the LEMAN system for interactively and intuitively constructing and animating layered elastic characters.

Magnenat-Thalmann and Thalmann (1993) further concentrated their efforts towards the visualisation of the behaviour of realistic human beings. They suggested behavioural techniques for automating high-level control of virtual actors. In their approach, they introduced the concept of autonomous actors, which react to their environment and are capable of taking decisions based on perception systems, memory and reasoning.

Similarly, however focusing on a totally different ecosystem, Tu and Terzopoulos (1994) proposed a framework for the realistic simulation of schools of tropical fish. In their approach,

each fish is modelled holistically as an autonomous agent situated in its physical world. An artificial fish consists of a spring-mass physically based model, which can swim hydrodynamically in simulated water through the motor control of internal muscles that motivate its fins. In their physics based virtual marine world the detailed motions of fish are not entirely predictable because they are not scripted.

In *"Simulating Humans"* Badler *et al.* (1993) seek to build computational models of human like figures which, though they may not trick our senses into believing they are alive, nonetheless manifest animacy and convincing behaviour. An interactive computer graphics model is endowed with reasonable biomechanical properties and is provided with human like behaviours. This simulated figure is used as an agent to effect changes in its world. A set of natural language instructions may describe and guide its tasks.

2.4.7 Interacting with Deformable Models

Several researchers have suggested methods for interacting with physically based deformable models in real time using immersive or non-immersive systems. Such applications require dedicated hardware with optimised computer graphics microcode.

Hilton and Egbert (1994) have developed an interactive tool for physically based 3D particle systems which is also extendible to soft objects. Their tool consists of bounded dynamic vector fields which apply forces, accelerations or velocities onto particles or object elements. Their method offers flexibility in animation, modelling and simulation and can solve a wide range of problems in computer animation such as rigid and non-rigid body motion, natural phenomena, modelling, image processing, simulation of gases, etc.

A method of virtual manipulation of elastic objects was introduced in (Miyazaki *et al.* 1995). They employed the conventional structure of mass points connected with elastic springs, to approximate the micro structure of an elastic object. As a virtual manipulation tool, they employed the model of a rigid cylinder with a hemispherical tip. This stress application tool

which was directly controlled by a 3D mouse, was used for virtual pushing and hitting the elastic objects.

Dias and Galli (1995) defined and implemented a class of physically based deformable objects for the object oriented graphics platform: Open Inventor 3D (Werneck 1994). They proposed a model for the simulation of deformable surfaces using an optimised version of a mixed dynamic energy method comprised of a discrete structural model based on Newtonian physics and a model based on energy minimisation principles. Their model may be dynamically loaded at run-time by a VRML browser, providing the capability of virtual walkthroughs in 3D scenarios populated with soft objects exhibiting realistic behaviours.

Finally, Impelluso (1996) presented a distributed system for a virtual environment which allows for haptic manipulation of deformable virtual objects. The distributed system consisted of a CrayT3D parallel computer for the numerical calculations of the deformations, an SGI-Onyx computer for the immersive display and the head and hand tracking system and a PC computer to drive the force feedback haptic device.

2.5 Constraining Deformation

Physically based deformable bodies pose severe and challenging problems in interactive control and animation design. Physical properties can generate a more natural behaviour but make it really difficult for the animator to control the location and the shape of an object in space and time. Many researchers recognised the need for the development of specialised constraint methods that forcibly confine the outcome of physical behaviour. There are two main categories in constraint methodology: the specification of spatial constraints and temporal constraints. Spatial constraint methods deal with issues like following predefined trajectories, collision detection and avoidance, inter-penetration and self-penetration avoidance, structural attachments, hierarchical structuring, etc. Temporal constraint methods deal with initial and final value problems of time dependent parameters like duration, velocity, acceleration, etc.

Witkin and Kass (1988) proposed a general space time constraint methodology. They adopted a formulation whose central characteristic is the numerical solution of the character's motion and time varying muscle forces over the entire time interval of interest. Their method optimises certain functions, such as energy consumption, that specify how the motion should be performed in terms of efficiency, smoothness, etc. Solving the constrained optimisation problem yields optimal physically valid motion that achieves the animator specified goals. They applied this method on rigid articulated bodies with uniform mass and frictionless joints.

Platt and Barr (1988) presented a framework for flexible body constraints. They introduced two types of constraints. Reaction Constraints (RCs) allow fast computation of collisions of flexible models with polygonal models. RCs also, allow the models to be pushed and pulled under the control of the animator. Augmented Lagrangian Constraints (ALCs) create effects such as volume preserving squashing and stretching. By adding physical constraints to elastic models, they achieve a compromise between a complete specification of the motion on one hand and a pure numerical simulation solution on the other.

Cohen (1992) proposed an interactive framework for the space time constraints introduced in (Witkin and Kass 1988). Cohen's approach allows the animator to interactively specify constraints and objectives for the motion and to guide the numerical solution of the optimisation problem. Constraints and objectives are defined over space time, referring to the set of all forces and positions of all of a creature's degrees of freedom, from the beginning to the end of an animation sequence. Constraints may be expressions that should be true at a particular point in time. Also, constraints may express a relationship that should be true at all times or over a limited range of time.

Baraff and Witkin (1992) presented yet another model for the dynamic simulation of flexible bodies, subject to non-penetration constraints. They implemented global deformations due to forces such as gravity, viscous damping, contact forces and restoring forces.

Metaxas and Terzopoulos (1992) developed a systematic approach to deriving dynamic models from parametrically defined solid primitives, global geometric deformations and local finite element deformations. Their models behave in a physically correct way with prescribed mass

distributions and elasticities at interactive rates. They also proposed efficient constraint methods for connecting the dynamic primitives together to make articulated models.

Thomas-Ngo and Marks (1993) suggested a genetic algorithmic approach for the development of a global search model that is capable of generating multiple trajectories for space time constraint problems. The key element of their research is a method for encoding trajectories as behaviours and a genetic search algorithm for choosing behaviour parameters.

Bruderlin and Williams (1995) introduced multi-resolution motion filtering, multi-target motion interpolation with dynamic time warping, wave shaping and motion displacement mapping. Their techniques are well suited for reuse and adaptation of existing motion data such as joint angles, joint coordinates or higher level motion parameters of articulated figures. This approach is complementary to key framing, motion capture and procedural animation.

Witkin and Popoviç (1995) described a similar technique for editing captured or key framed animation. Their method allows interactive definition of constraints which smoothly deform the original motion. They also achieve smooth transitions between motions with motion blends. Finally, Snibbe (1995) presented a new set of interface techniques for visualising and editing animation directly in a 3D scene. Snibbe suggested the separation of spatial and temporal control of position by using two curves for each animated object: the motion path which describes the 3D spatial path along which an object travels and the motion graph, a function describing the distance travelled along this curve over time.

2.6 3D Computer Animation Systems

Most commercial 3D computer animation systems have incorporated techniques for the creation and control of flexible models. The two most dominant, in the computer animation industry, applications are Alias/Wavefront Power Animator v8 and Maya from Silicon Graphics, Inc (Alias / Wavefront 1997) and Softimage 3D from Microsoft Corporation (Softimage 3D 1997).

Alias/Wavefront Power Animator v8 and Maya support "Mold and Sculpt", an intuitive free from sculpting tool that allows the animator to interactively shape a 3D parametric surface object. It also supports "Blobby Objects", a feature that enables the creation of organic objects using implicit surfaces. "Deformation" is a set of tools for global deformation application like bending, twisting, shearing, squashing and stretching. "Character Builder" is a tool for the modelling of articulated figures, which can be animated by muscles and dressed with a deformable skin layer. Alias/Wavefront is implemented for Silicon Graphics workstations which incorporate specialised graphics hardware. It is a powerful modelling system capable of accurate modelling suitable for the product design industry. It is, equally, a powerful animation system geared towards the production of realistic animation for film special effects.

Softimage 3D supports "Meta-Clay Modelling", a density based modelling tool for organic, sculptured 3D parametric surface objects. "Deformation" of models can be achieved via cluster, control point, effector, lattice, patch, spline or vertex manipulation. "Q-Stretch" is a tool for automatic squash and stretch deformation based on speed and acceleration. "Skin" is a tool for automatic local and global weighted envelopes. Lastly, "Dynamics" is a tool for the specification of density, mass, elasticity, force, etc. Softimage/3D Extreme is implemented for Silicon Graphics workstations and for high configuration personal computers. It is also a powerful system focused on fast and high quality animation production and it dominates the advertising industry.

Another established 3D computer animation application with large user base is 3D Studio MAX from Kinetix Ltd (Studio 3D 1996). Apart from the basic global deformation techniques for bending and stretching of 3D polygonal objects, the system offers the ability to be extended with interesting third party plug-ins. Such extensions include procedural animation models like ripples, waves, particle systems, gravity, wind, etc. Another important tool that also comes as an extension to 3D Studio is the "MetaReyes" modeller from 4Dvision Ltd, which uses the metaball modelling technique to help easily build polygonal objects with complex organic look. This modelling tool becomes really useful with the "Bones Pro" extension from Digimation Ltd. This last extension provides with a tool for the specification of underlying articulated structures which can deform the surrounding polygonal surfaces. 3D Studio MAX has, for some time, dominated the low-end advertising and computer graphics industry.

2.7 Summary and Conclusion

Modelling of 3D deformable bodies has attracted research and development efforts for more than a decade now. The subject is young and the progress is rapid. Unavoidably, the present survey is by no means complete. Only a review of the most representative approaches has been attempted.

Two standard geometric deformation methods were described. Barr's global deformation (Barr 1984) and free form deformation techniques (Sederberg and Parry 1986) form the basis for most conventional 3D sculpting tools. Geometric deformation methods have been be extended to animation through metamorphosis (Lerios *et al.* 1995), (Coquillart and Jançene 1991) and procedural models (Fournier and Reeves 1986). Geometric deformation models have, also, been linked to physically based models to provide more sophisticated, hybrid deformable models suitable for animation (Witkin and Welch 1990), (Chadwick *et al.* 1989), (Magnenat-Thalmann and Thalmann 1993).

Pure physically based deformable models like the elastic spring-mass formulation (Terzopoulos *et al.* 1987) have formed the basis for a new school of thought in flexible body modelling. Major research efforts have been directed towards cloth modelling (Aono 1990), facial muscle modelling (Waters 1987), and skin, muscle and bone layered modelling (Turner 1995). The study of inelastic material behaviour has lead several researchers towards the development of viscoelastic, plastic and fracture models (Terzopoulos and Fleischer 1988). Such models were extended towards a particle based approach to inelastic material behaviour (Tonnesen 1991).

Integrated systems for the construction and animation of autonomous virtual actors were reviewed (Magnenat-Thalmann and Thalmann 1993), (Tu and Terzopoulos 1994). Several interactive manipulation and stress application techniques for physically based flexible bodies, in real time, were introduced (Miyazaki *et al.* 1995), (Impelluso 1996). Finally, there was a brief survey of techniques for constraining flexible bodies in space and time (Witkin and Kass

1988), (Platt and Barr 1988).

There have been some attempts towards the development of hybrid models that are based on geometric deformations and simplifications of laws of physics (Witkin and Welch 1990), (Chadwick *et al.* 1989), (Magnenat-Thalmann and Thalmann 1993).

Geometric deformation models pose the problem of non-intuitive interaction and rely heavily on animator and programming skill. Their great advantage, though, is speed of execution on medium to low-cost hardware. Physically based deformation models, on the other hand, pose the problem of space and time constraints and demand highly configured specialised graphics workstations. Important benefits, though, are intuitive direct manipulation of 3D objects and automated natural looking animation. Hybrid deformation models utilise the speed in shape deformation, offered by geometric models and improve the natural look by incorporating physically based elements. Hybrid models have, so far, achieved satisfactory results in the idealised areas of perfectly plastic and perfectly elastic material behaviour. There have not been found any attempts to merge these two together and, further, combine them with the physical property of viscoelasticity.

This research work discusses the problem of interactive control and animation of deformable organic objects. The focus of this research is concentrated on the development of a hybrid deformation model that incorporates the physical properties of plasticity, elasticity and viscoelasticity in one continuous medium.

The following chapter examines the theoretic basis along with some experimental data of organic material behaviour.

3

Organic Material Behaviour

3.1 Introduction

Living creature movement is characterised by organic material behaviour of soft tissues such as skin, fat and muscle and of hard tissue such as bone. The mechanical properties of biological tissues have some features similar to those of familiar engineering materials and other features very different from them. In general, biological tissues exhibit a complex deformable response towards external stresses. Elasticity, plasticity and viscoelasticity are physical properties encountered in the deformation behaviour of the majority of living tissues.

The objectives of this chapter are:

- to introduce basic background theory of elasticity, plasticity and viscoelasticity,
- to examine the deformation characteristics of organic tissues.

3.2 Elasticity

Elasticity is the property that makes objects resume their original shape after being deformed. Elastic material behaviour is characterised by internal energies which develop inside the body of an object as soon as a deformation appears and they grow monotonically as the magnitude of the deformation increases. These elastic energies generate elastic forces which tend to restore the deformation completely immediately after the removal of the external force. A rubber object can be stretched but when it is released it springs back to its original length. A wooden bow can be bent but straightens again when released. Elasticity is most obviously, a property of synthetic materials which can be deformed considerably without breaking. However, to a certain extent, elasticity is a property of all solids. Wood, concrete, steel, most organic tissues, all exhibit elastic behaviour when subjected to a certain range of external stresses.

3.2.1 Definition of Elasticity Terms

Some terms will be defined by describing an imaginary specimen which is perfectly regular and behaves in an ideal way.

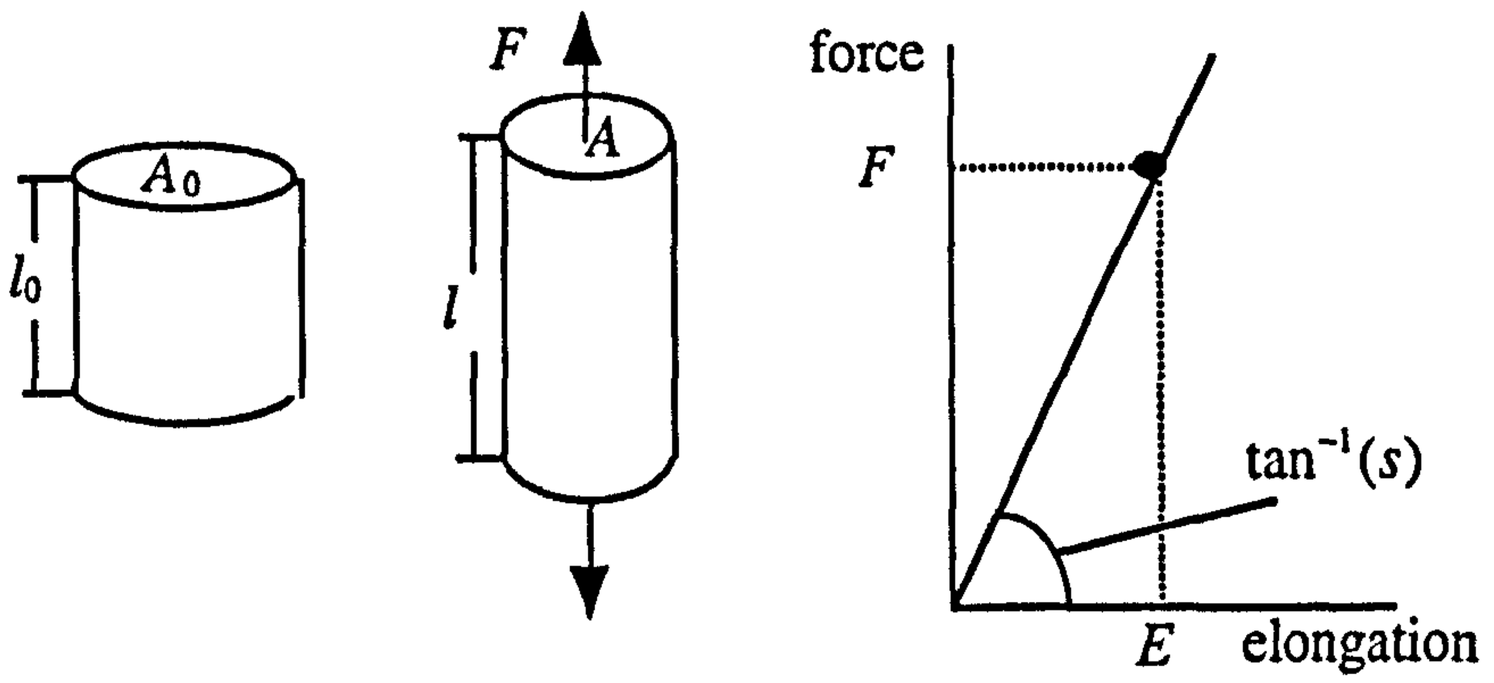


Figure 3.1 a. An ideally elastic specimen at rest. b. Stretched by force F.
c. The graph shows the linear relationship between force and elongation.

This specimen is initially a cylinder of length, l_0, and uniform cross-sectional area, A_0. It is

stretched by a force, F, acting on its ends, to a new length, l. The absolute elongation of the specimen is $E = l - l_0$. As it stretches it gets thinner and the cross-sectional area of the stretched specimen becomes A (see figure 3.1a and b) (Lianis 1981). A graph of F against E is a straight line through the origin (see figure 3.1c).This ideal material obeys Hooke's law, according to which, force is directly proportional to absolute elongation.

$$F = sE \qquad [3.1]$$

The gradient of the graph is called the tensile stiffness, s, of the specimen. Hooke's law can also be expressed using the concepts of stress and strain. The nominal stress, σ, is defined as the load, F, divided by the original cross-sectional area, A_0.

$$\sigma = \frac{F}{A_0} \qquad [3.2]$$

The nominal strain, e, is defined as the increase in length per unit original length.

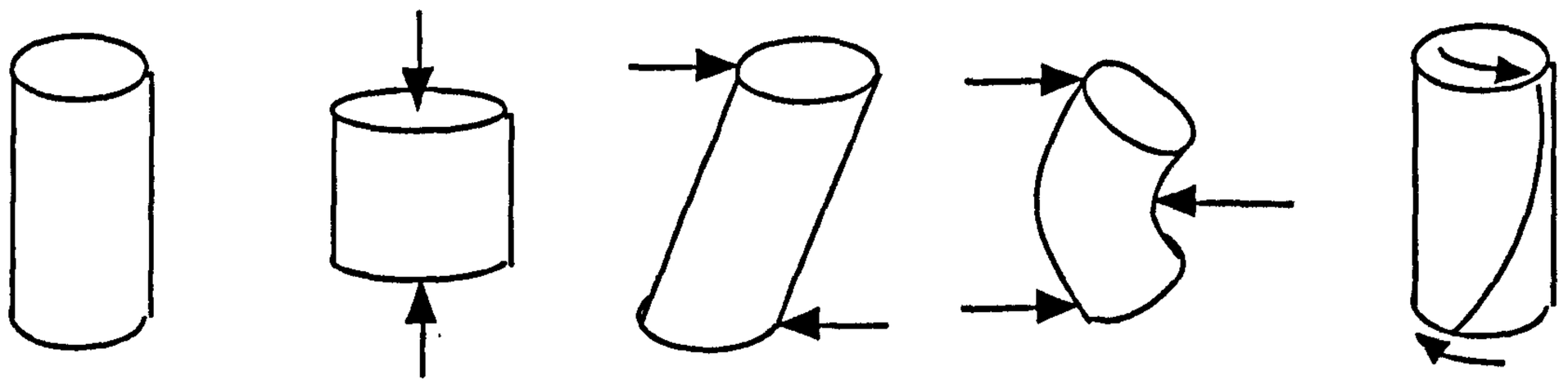
$$e = \frac{E}{l_0} = \frac{(l - l_0)}{l_0} \qquad [3.3]$$

So Hooke's law becomes:

$$\sigma = \mathcal{E}e \qquad [3.4]$$

where $\mathcal{E}$ is called Young's modulus and describes the elastic property of the material of the specimen. Although nominal stress and strain are not very accurate, as opposed to true stress and strain, they are frequently used in experiments because they are easier to measure. Elongation is measured in metres and elongation strain is dimensionless, because it is derived as the ratio of two lengths. Forces are measured in newtons (N) and stresses are measured in newtons per square metre also called pascals (Pa). Young's modulus is also measured in pascals.

3.2.2 Types of Strain



a. **b.** **c.** **d.**

Figure 3.2 Various types of strain: a. squashing, b. shearing, c. bending and d. twisting.

Stretching is not the only way an object can be deformed. Different types of strain may be defined depending on the direction of the strain in relation to the shape of the specimen. Squashing, shearing, bending and twisting are also common types of strain (see figure 3.2).

3.2.3 Elastic Strain Energy

A certain amount of work is needed for the application of any strain to an elastic specimen. In the stretching example (see figure 3.1), the work needed to stretch the specimen by a small additional amount Δl (too small to involve an additional increase of force), is $F\Delta l$. This work, $F\Delta l$, is represented by the narrow stippled area, in figure 3.3.

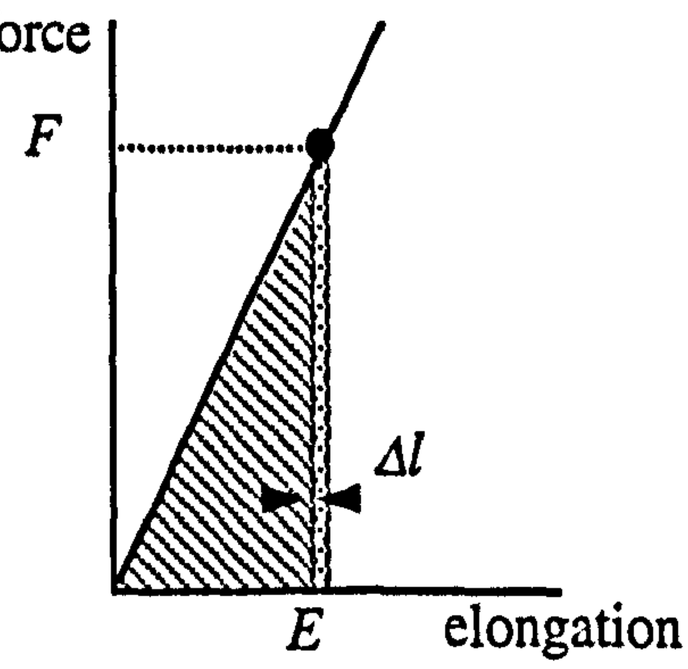


Figure 3.3 The striped area represents the total elastic strain energy stored inside the body of the strained specimen.

The total work done as the force increases from 0 to F, is represented by the striped area under the graph. This work is converted to elastic strain energy U, which is stored inside the body of the strained specimen.

$$U = \frac{1}{2} FE = \frac{1}{2} sE^2 \qquad [3.5]$$

This energy will be converted back to work during the elastic recoil, when the external force is removed.

3.3 Plasticity

3.3.1 Perfectly Plastic Material

A plastic object yields under the influence of relatively small stresses. The deformation of a plastic object is permanent, since there are no internal elastic forces to restore the initial shape. The entire amount of work spent for a plastic deformation is absorbed by the shape deformation and cannot be retrieved. A characteristic plastic material is modelling clay which is well known for its mouldability and it is commonly used in traditional modelling animation.

3.3.2 Elastoplastic Material

Most materials yield to plastic strain, only when the stress exceeds a certain limit. Under this stress limit these materials behave elastically. A typical elastoplastic material obeys Hooke's law, only under a limited range of external stresses (see figure 3.4). Inelastic deformations occur in real materials, for temperatures and forces exceeding certain limiting values, above which, irreversible dislocations at the atomic level can no longer be neglected.

The graph in figure 3.4, represents a simplified stress / strain relationship of a typical

elastoplastic solid material. In fact, many materials exhibit a much more complex mechanical behaviour depending on the direction of stress application, frequency of stress application, temperature and other factors.

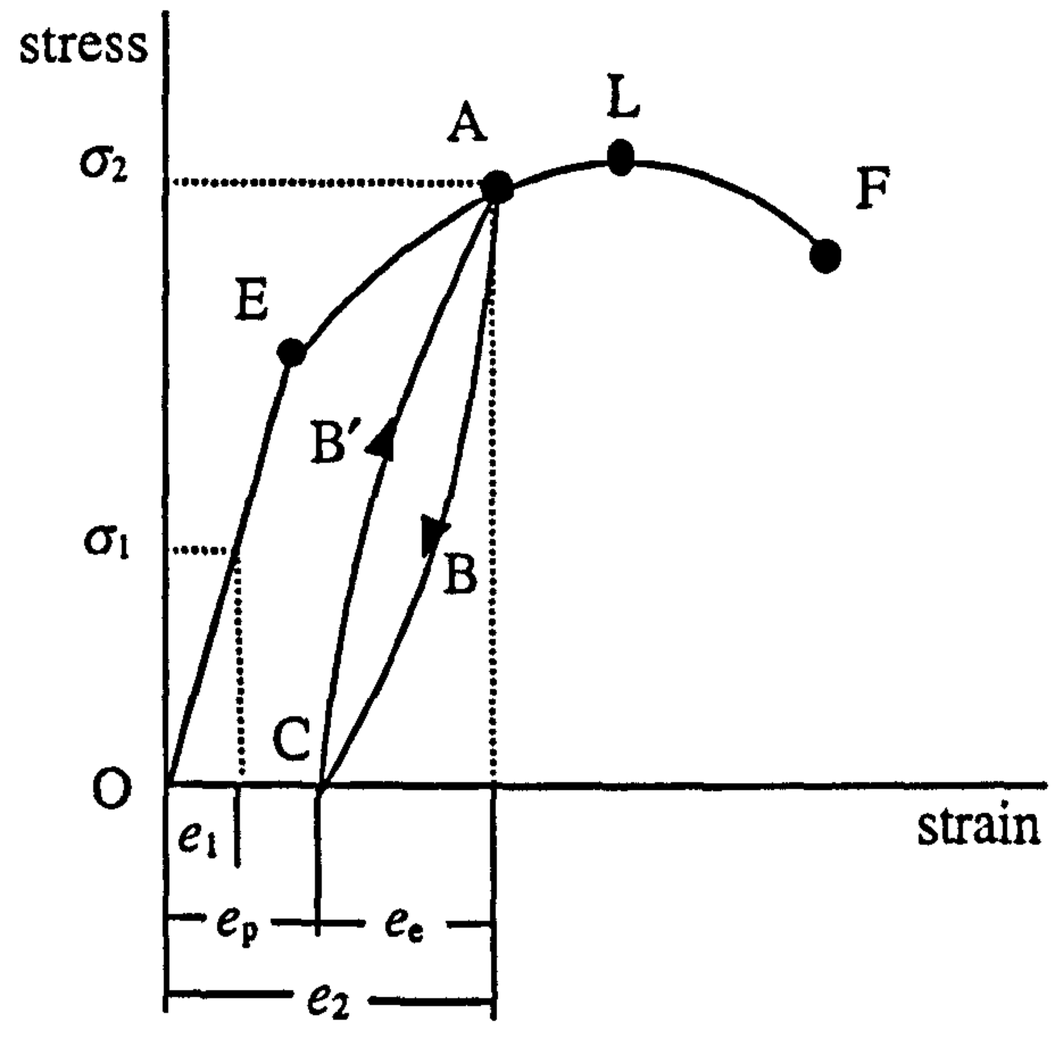


Figure 3.4 A graph of stress against strain for a typical elastic material (Mendelson 1968).

For a small stress, σ_1, in the range OE (see figure 3.4), the stress is linearly proportional to the strain, e_1. Upon removal of the stress the specimen returns to its original shape. When the stress exceeds the elastic limit, E, the strain increases at a greater rate. For a large stress, σ_2, beyond the elastic limit, permanent plastic deformation takes place. Upon removal of this stress a portion e_e, of the total strain e_2, is elastically restored following the curve ABC, which is approximately parallel to line OE. A portion, e_p, of the strain remains as plastic deformation. When reloading the specimen a slightly different curve is traced, CB′A. This phenomenon is called hysteresis. In some materials the hysteresis loop, ABCB′, is of considerable size but in most it is so thin that it may be neglected. When the stress exceeds the point of maximum load, L, the material starts to flow with no extra increase in stress until it finally fractures at point F.

3.4 Viscoelasticity

In some materials, stress and strain are related to time in such a way, that the exhibited mechanical behaviour may be characterised as elastic and viscous at the same time. As mentioned in section 3.2, the stress / strain relationship in a perfectly elastic material may be described by Hooke's law:

$$\sigma = Ee \qquad [3.6]$$

In a Newtonian viscous fluid, however, stress application results to strain rate (i.e. strain velocity):

$$\sigma = v\dot{e} \qquad [3.7]$$

where v is the viscosity coefficient of the fluid and the dot denotes first derivative of time.

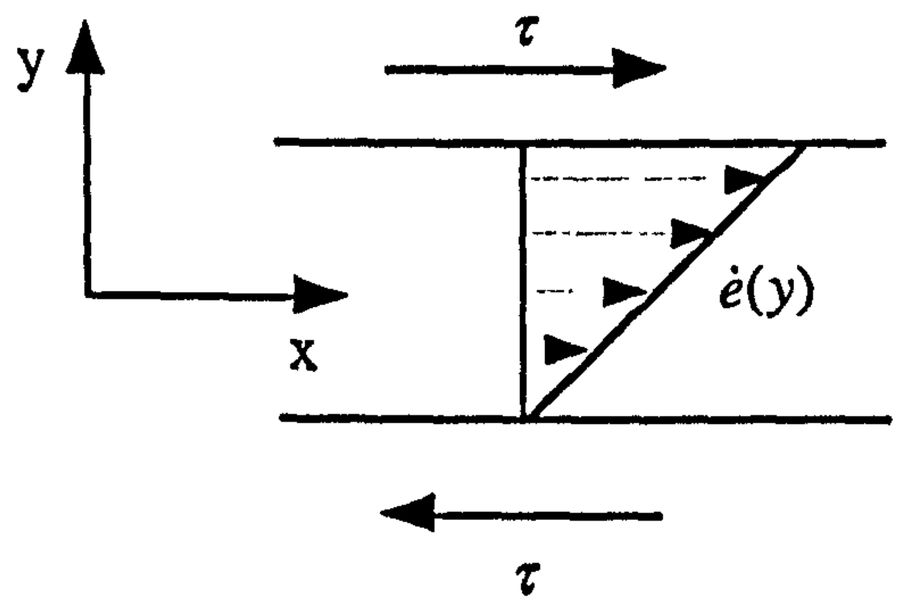


Figure 3.5 Velocity gradient in a Newtonian viscous fluid.

Viscosity is the measure of internal friction that occurs between different layers of a fluid moving with different velocities. When a stress is applied to a viscous fluid the fluid motion is characterised by a gradient of decreasing velocity with increasing distance from the point of stress application (see figure 3.5):

$$\tau = v\frac{d\dot{e}(y)}{dy} \qquad [3.8]$$

When a viscoelastic material is being subjected to a certain stress the caused strain flows like in a viscous fluid and it develops a strain velocity over time. As soon as the stress is removed elastic restoration takes place and the material gradually returns to its initial shape like an elastic solid. Viscoelastic material behaviour is encountered in some synthetic materials such as high polymers.

Most biological tissues are approximately viscoelastic. In the biological world atoms and molecules of diverse nature are organised into complicated structures of cells, tissues and organs. As a result the mechanical behaviour of a part of a living creature is also very complicated. Animal muscle, skin, fatty tissue, plant tissue, etc., all exhibit approximately viscoelastic material behaviour of great complexity.

3.4.1 Features of Viscoelasticity

Viscoelasticity is a material property, which is manifested with the following characteristic features of stress and strain behaviours: strain creep, stress relaxation and hysteresis.

3.4.1.1 Strain Creep

If a viscoelastic body is suddenly stressed and the stress is maintained constant afterward, the body continues to deform for some time.

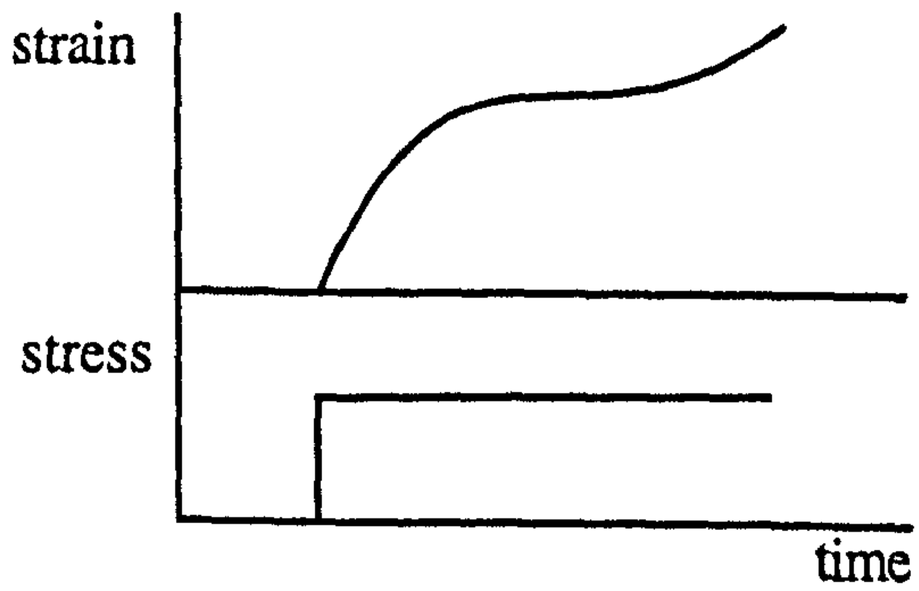


Figure 3.6 Strain creeps over time while stress is maintained constant.

This phenomenon is called strain creep (see figure 3.6). Strain creep occurs in viscoelastic materials, where intermolecular connections are very loose. As a result, different parts of an object respond to the same external stress, with different timing. In fact, there is spatial gradient of strain velocity around the stress application area. So, strain develops slower in parts of an object that lie further away from the centre of stress application.

3.4.1.2 Stress Relaxation

When a viscoelastic body is suddenly strained and the strain is maintained constant afterward, the corresponding stresses, induced inside the body, decrease with time. This phenomenon is called stress relaxation (see figure 3.7).

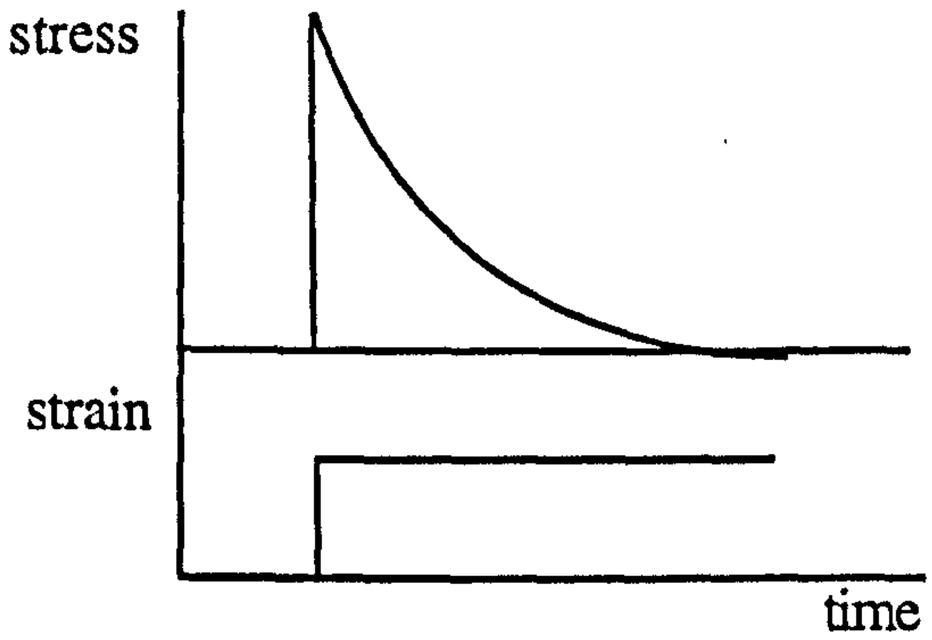


Figure 3.7 Stress relaxes over time while strain is maintained constant.

Stress relaxation is another characteristic phenomenon encountered in viscoelastic materials. The amount of elastic energy stored inside a viscoelastic object is gradually being converted to permanent deformation. So, if the external stress is removed after a considerable amount of time, the object having lost its elastic energy, does not experience elastic restoration.

3.4.1.3 Hysteresis

If a viscoelastic body is subjected to cyclic loading the stress / strain relationship in the loading process is usually somewhat different from that in the unloading process. This phenomenon is

called hysteresis (see figure 3.8).

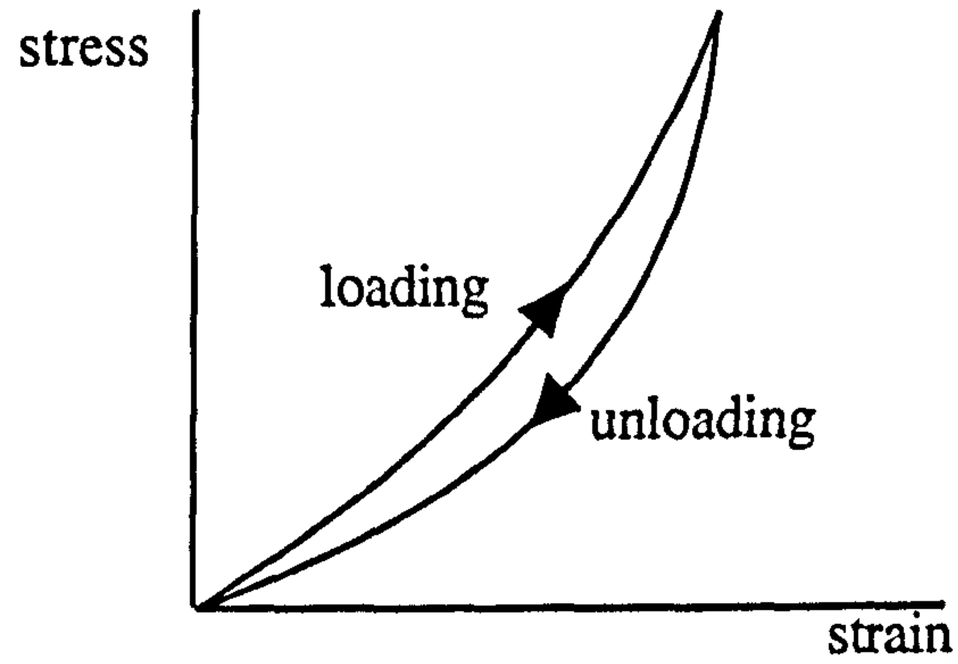


Figure 3.8 Hysteresis loop in a stress / strain relationship.

The stress / strain relationship in viscoelastic materials is generally non-linear and the value of Young's modulus ($\mathcal{E} = \sigma / e$) is not constant (see figure 3.8). Graphs, such as the ones in figures 3.6, 3.7 and 3.8, may be derived through laboratory experiments. In reality, the curves are more complicated and each viscoelastic material exhibits mechanical behaviour which differs substantially from others. Thus, it becomes very difficult to establish rules and to define a theory of viscoelasticity that is satisfied by most viscoelastic materials.

3.4.2 Mechanical Viscoelastic Models

An idealisation of viscoelastic behaviour can be achieved by constructing mechanisms, which consist of springs and dashpots, connected to each other. Such mechanical models have often been used to approximate viscoelastic behaviour in computer graphics (Terzopoulos and Fleischer 1988). A linear spring with spring constant, s, is supposed to produce an instantaneous deformation which is proportional to the load (see figure 3.9a). Thus, if F is the force acting on the spring and E is the extension then:

$$F = sE \qquad [3.9]$$

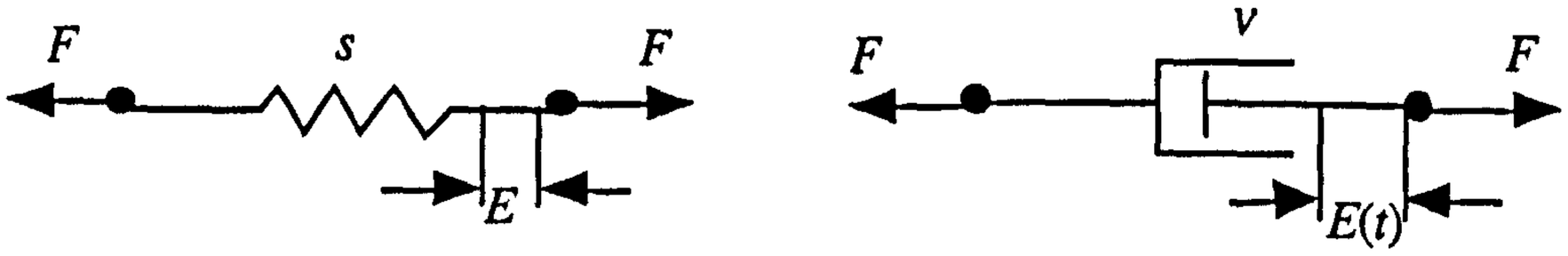


Figure 3.9 a. Linear spring, b. linear dashpot.

A linear dashpot, with coefficient of viscosity v, is supposed to produce a velocity proportional to the load at any instant (see figure 3.9b). Thus, if the force F acts on a dashpot it will produce a velocity of deformation:

$$F = v\dot{E} \qquad [3.10]$$

Common mechanical viscoelastic models are the Maxwell model, the Voigt model and the Kelvin model.

3.4.2.1 The Maxwell Model

The Maxwell (James Clerk Maxwell 1831-1879) mechanical viscoelastic model consists of a spring and a dashpot connected in series (see figure 3.10a).

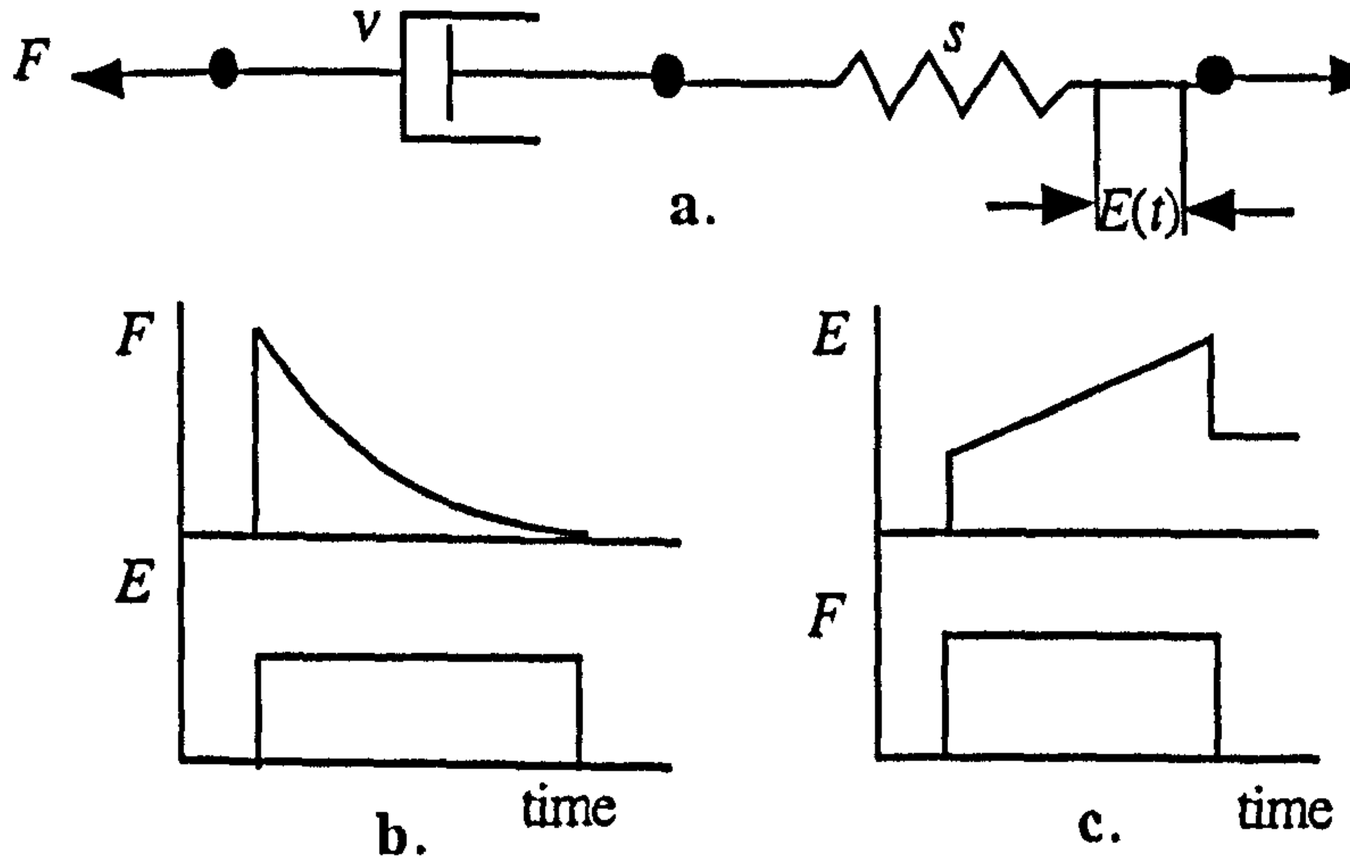


Figure 3.10 a. The Maxwell viscoelastic model.
b. Stress relaxation graph, c. strain creep graph (Fung 1981).

A constant deformation of the model produces an immediate reaction by the spring, which is followed by a gradual stress relaxation (see figure 3.10b). On the other hand, a constant application of a load induces an immediate deformation of the elastic spring which is followed by a linear strain creep of the dashpot (see figure 3.10c).

3.4.2.2 The Voigt Model

The Voigt (Woldemar Voigt 1850-1919) model consists of a spring and a dashpot connected in parallel (see figure 3.11a).

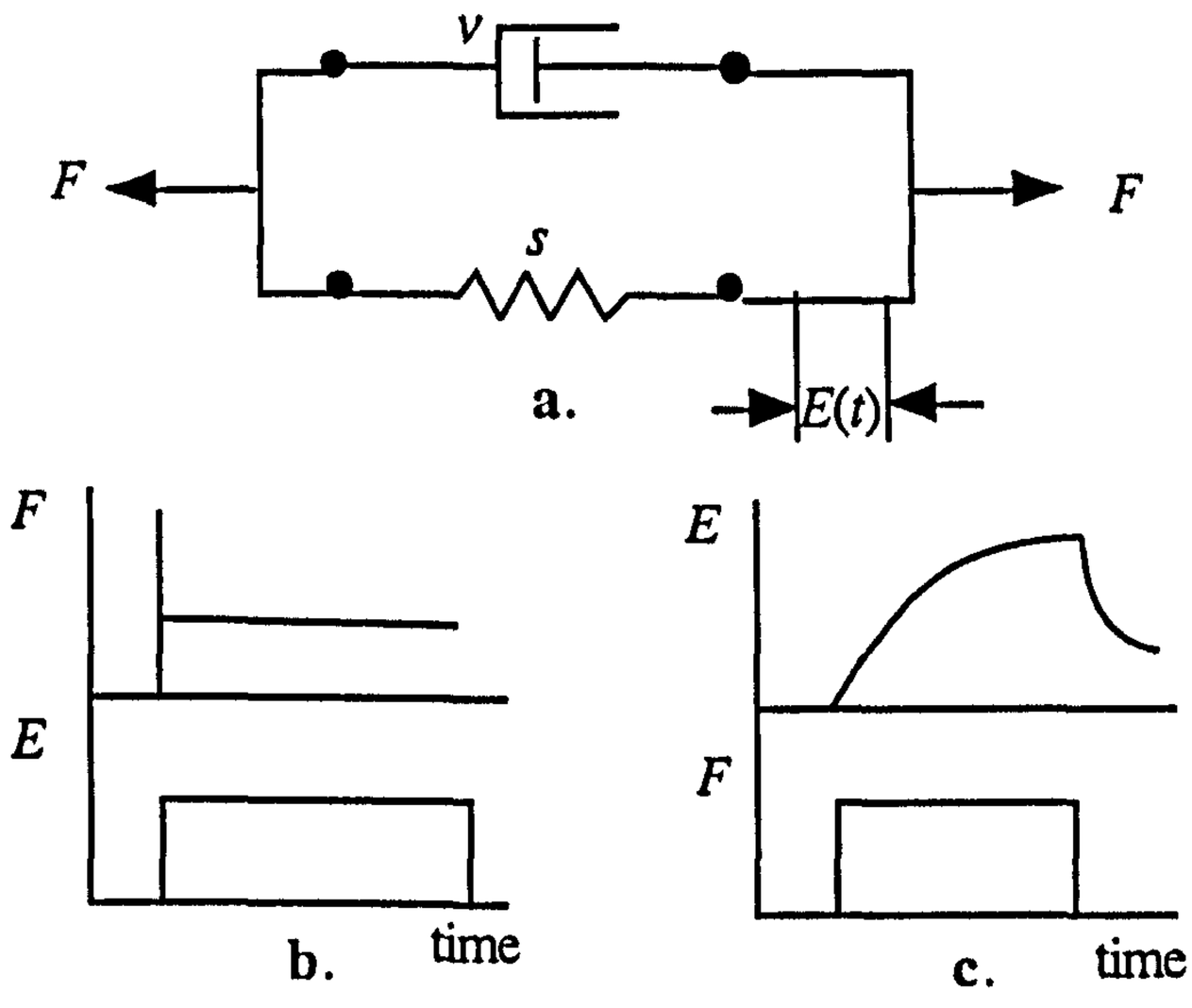


Figure 3.11 a. The Voigt viscoelastic model.
b. Stress relaxation graph, c. strain creep graph (Fung 1981).

A constant deformation of the model produces an immediate reaction by the spring, which is followed by an instantaneous stress relaxation (see figure 3.11b). A constant application of a load will produce no immediate deformation, because the dashpot will not move instantaneously. Instead, deformation will develop gradually, as the spring takes a greater and greater share of the load (see figure 3.11c).

3.4.2.3 Kelvin Model

The Kelvin (Lord Kelvin 1824-1907) model consists of a dashpot and a spring connected in series and a second spring connected in parallel to the other two (see figure 3.12a). The stress relaxation and strain creep graphs appear to be a combination of the respective graphs from the Maxwell and the Voigt models (see figure 3.12b and c).

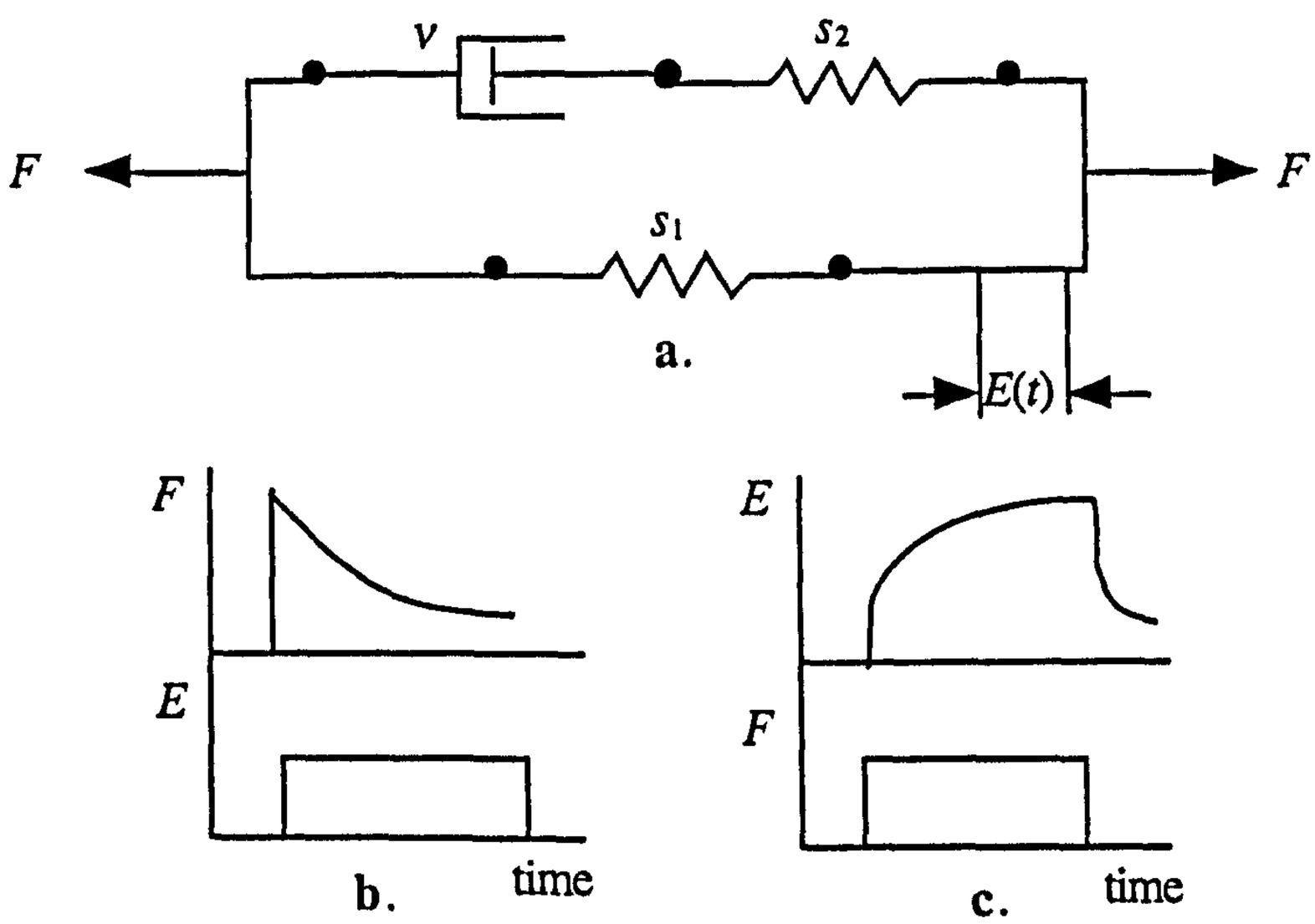


Figure 3.12 a. The Kelvin viscoelastic model.
b. Stress relaxation graph, c. strain creep graph (Fung 1981).

These mechanical models exhibit a form of idealised viscoelastic behaviour. However, in the following section it is shown that organic viscoelastic materials exhibit a more complex behaviour.

3.5 Material Behaviour of Organic Tissues

In this paragraph, the elastic and viscoelastic nature of characteristic samples of biological tissue is presented. Several graphs based on experimental data illustrate the peculiarities often encountered in the mechanical behaviour of organic tissues.

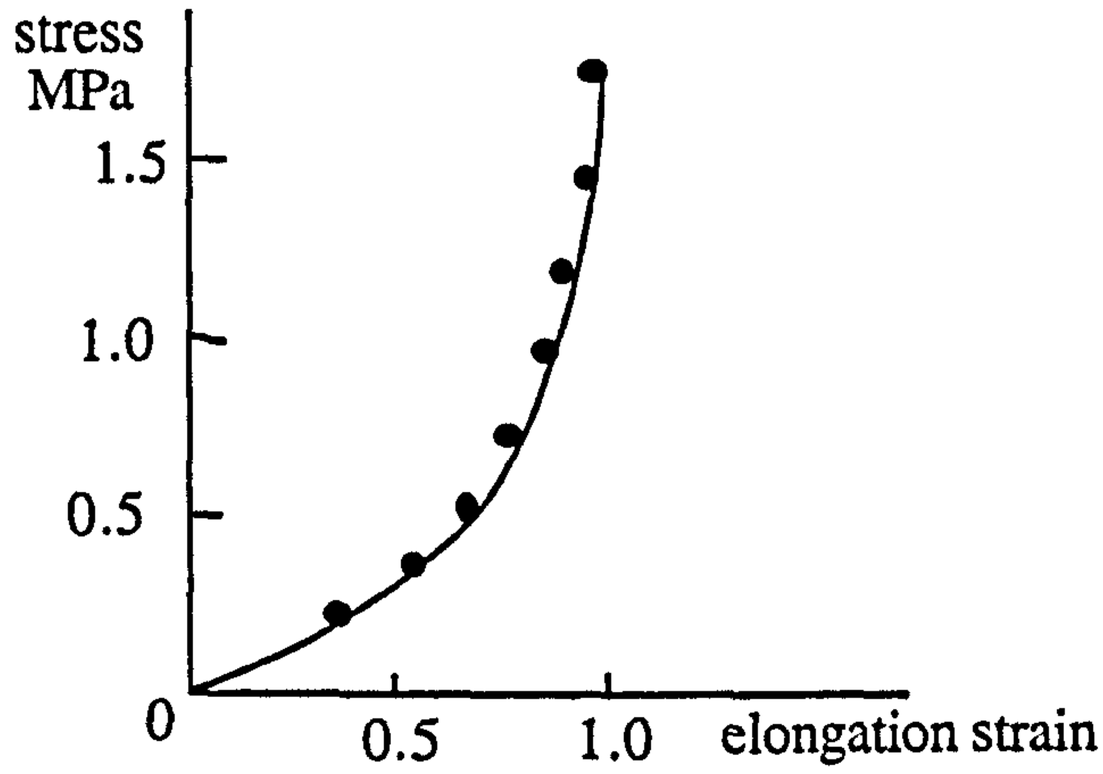


Figure 3.13 A graph of stress against strain obtained from experimental data.
The specimen used for the experiment was the ligament of a deer (Dimery *et al.* 1985).

Dimery *et al.* (1985) performed simple stress experiments on ligament specimens of sheep and deer. Thus, graphs of stress against strain were obtained from the experimental data. In figure 3.13, the stress / strain graph is approximated by a curve. If Hooke's law, of linear elasticity, were obeyed, the graph would be a straight line (see figure 3.1).

Sea anemones and jellyfish have a jelly-like layer of mesogloea in their body walls. Alexander (1962) and Koehl (1977) made another simple experiment. Instead of applying different stresses and measuring the resulting strains, they applied a constant stress and observed the strain changing over time.

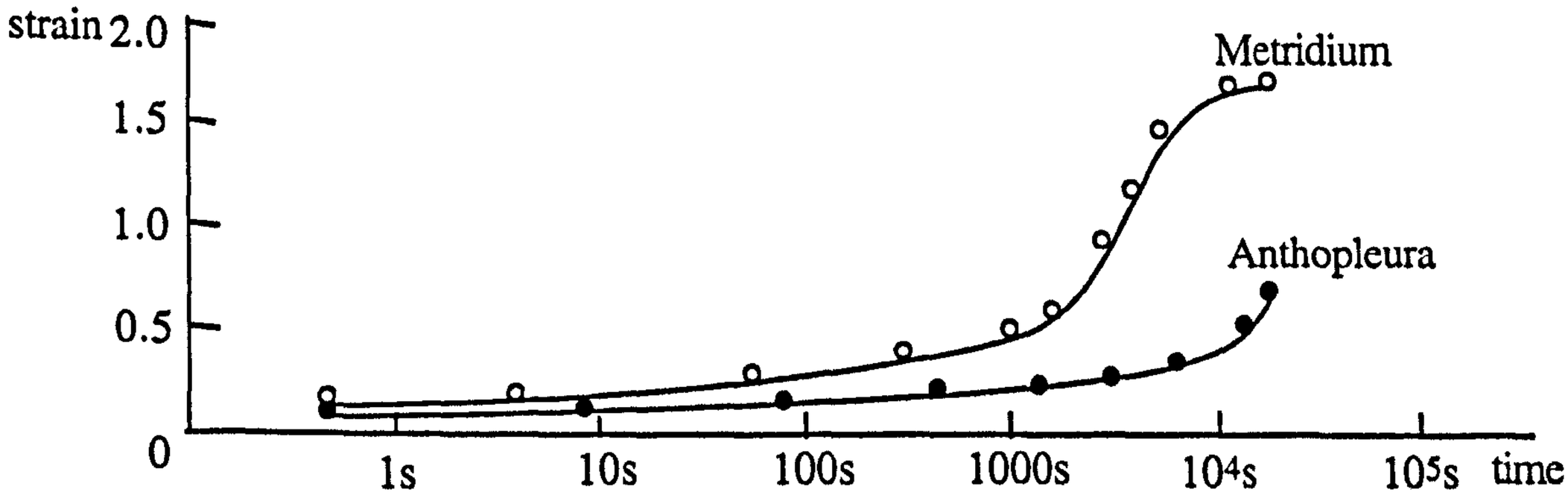


Figure 3.14 Graphs of strain against time for strips of sea anemone mesogloea, stretched by constant stresses (Koehl 1977) (metridium and anthopleura are common sea anemones).

The specimen stretched gradually over a long period of time (see figure 3.14). If the stress was removed after the experiment the specimen returned gradually to its initial length. The change in length, though slow, was perfectly elastic. Sea anemones are not peculiar in exhibiting these time dependent properties, but they show them very clearly.

The biological strain creep curve (see figure 3.14) seems to differ considerably from the respective strain creep curves of the mechanical viscoelastic models (see figures 3.10, 3.11 and 3.12).

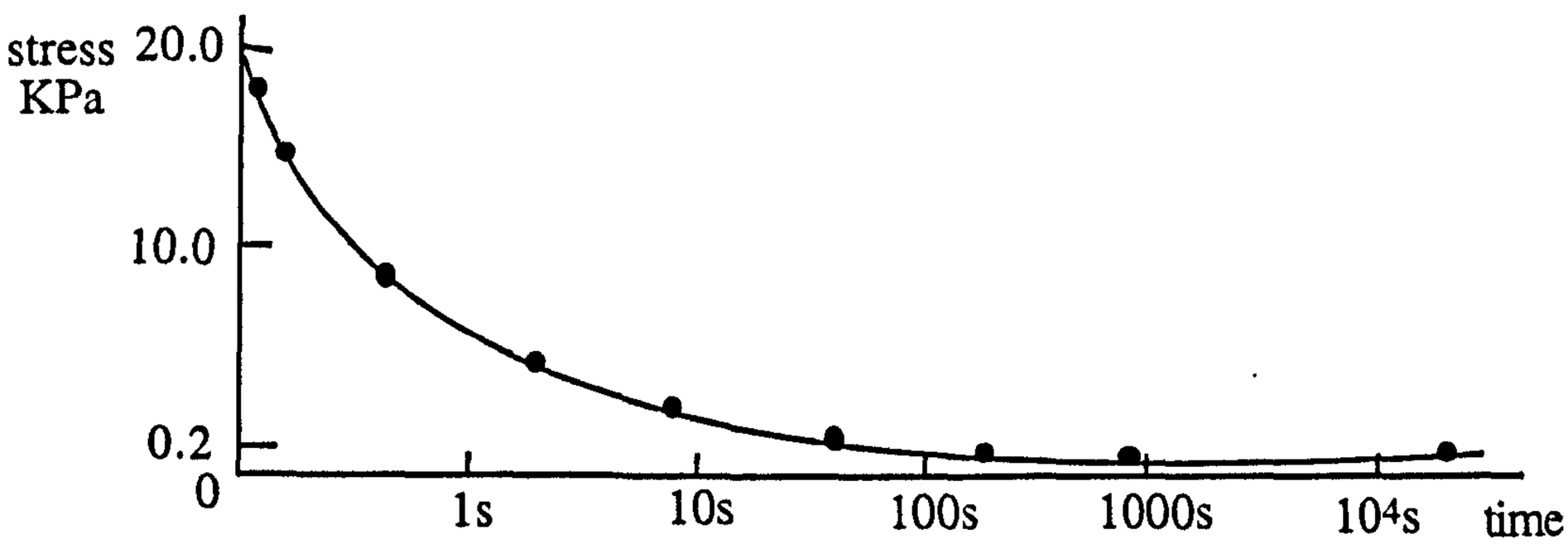


Figure 3.15 Stress / time graph for strips of sea anemone mesogloea (Cosline 1971).

Similar tests (Cosline 1971) showed that an organic specimen stretched at a constant length for a long time, caused the stress to gradually decline, showing the phenomenon of stress relaxation (see figure 3.15).

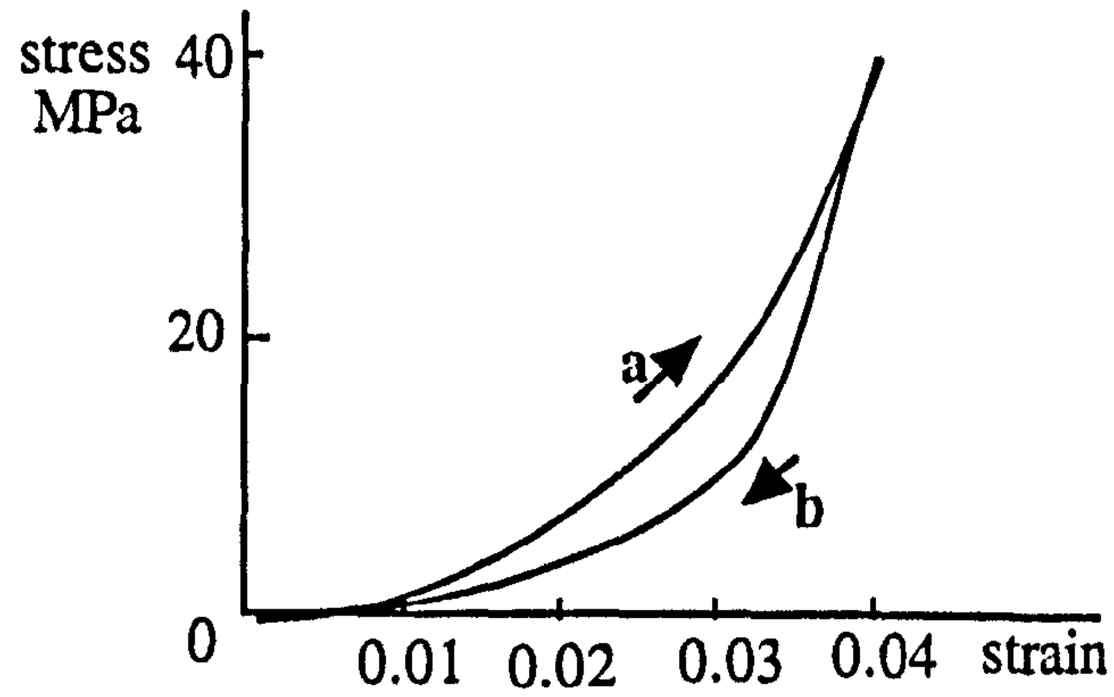


Figure 3.16 Graph of stress against strain, during a. loading and b. unloading a specimen of tendon of a kangaroo (Ker *et al.* 1986).

Ker *et al.* (1986) applied cyclic loading by stretching a tendon specimen and then allowing it to recoil. The data gathered from this experiment lead to the graph in figure 3.16. Notice that the graph forms a loop. The upper line (a) represents the stretching process of the organic specimen and the lower (b) represents the recoil process. A cyclic loading process causes the phenomenon of hysteresis. Stress / strain hysteresis is present to all materials to some extend, but is particularly obvious in soft organic tissues.

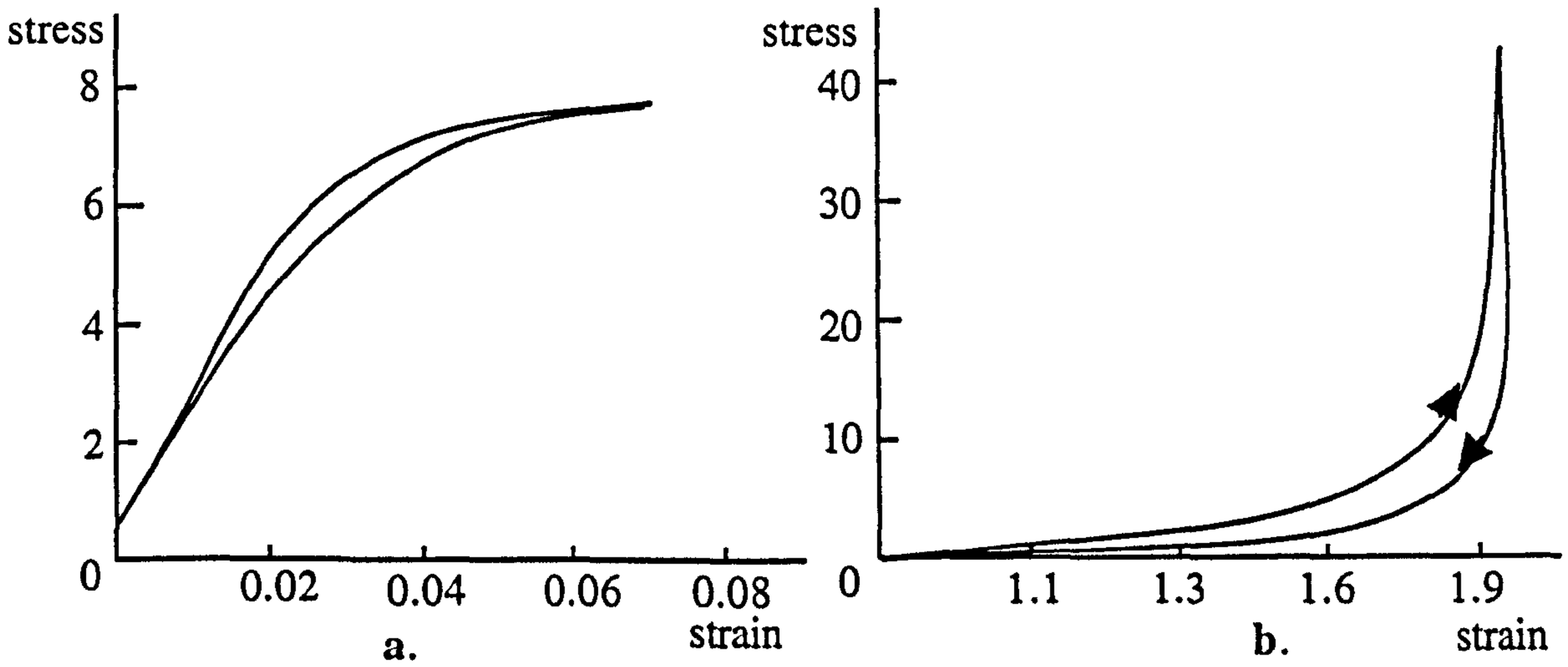


Figure 3.17 Stress / strain graphs of a. human bone and b. human skin (Yamada 1970).

Yamada (1970) performed similar stress and strain experiments on human bone and soft skin (see figure 3.17).

All biological materials show creep, stress relaxation and hysteresis phenomena to some extent. In soft tissues, in the physiological state, the stress generally increases exponentially with increasing strain. But when stress exceeds the physiological range the tissue yields and breaks. Biological stress / strain, strain creep, stress relaxation and hysteresis curves are complex to the extent that they do not abide by the theories of linear elasticity and viscoelasticity. Some synthetic materials such as high polymers (rubber, plastics, silicon, etc.) exhibit material behaviour which is very similar to that of biological tissues and can also be characterised as viscoelastic.

3.6 Summary and Conclusion

This chapter presented basic theory of elasticity and viscoelasticity. Hookean linear elasticity is an idealised theory, obeyed by very few real materials and only when they are being subjected to a limited range of stresses. When the stress exceeds a certain limit most real materials yield to plastic deformation and finally break. Viscoelasticity is the combination of elasticity and viscosity in the same material. Some synthetic materials and most of biological materials, in physiological state, are viscoelastic. The phenomena of stress relaxation, strain creep and hysteresis are the characteristic features of viscoelastic behaviour. An idealised expression of these phenomena was introduced using simple mechanisms which consist of elastic springs and viscous dashpots connected to each other. Further, it was shown that the material behaviour of living tissues, although viscoelastic, differs considerably from the linear behaviour of mechanical models.

All living creatures consist of organic parts, most of which contribute actively or passively to external or internal body movement. Most animals exploit the elastic and viscoelastic properties of parts of their bodies for performing various tasks. They use muscles as spring antagonists (mammal breathing) or as spring energy stores (running, galloping, jumping, swimming, flying, hovering, etc.). They use bones as structural components (skeletons) and as tools (horns, tusks). Fatty tissue, skin and organs apart from their internal motion, they participate passively to external body movements. The material properties of these soft tissues greatly influence the overall appearance of a moving living creature. The combination of rigid skeletal movement with the viscoelastic response of soft tissues characterises most vertebrate animal movement (e.g. in mammal walking). Moreover, in invertebrates the difference in viscoelastic properties between layered soft tissues generates characteristic organic movement (e.g. in jellyfish swimming).

In 3D computer animation synthesis of realistic human and animal movement is a vital ingredient since the majority of computer generated character animations contain deformable life like characters. Living form movement poses, perhaps, one of the most difficult problems in animation. Undoubtedly, animator skill is and will always be crucial for realistic reproduction

of even simple animal movements. However, a self-activated model for organic material behaviour could replace a great amount of repetitive animation work and may let the animator concentrate on other higher level problems. The role of such model becomes more important particularly in the animation of soft tissue invertebrates such as octopodes, squid, cuttlefish, jellyfish, etc.

In the following chapter, there will be a discussion on various 3D object representation approaches and on their suitability for organic modelling.

4

3D Object Representation

4.1 Introduction

A 3D CG object consists of the topological and geometric description of a real or imaginary entity. The objective of this chapter is to introduce a representation model for 3D CG objects. Topology and geometry may be represented in a variety of ways, each suitable for different purposes and each requiring a different approach in the geometric modelling process. The most frequently encountered approaches in CG practice are parametric surface representations, polygonal representations and volume based representations.

4.2 Overview

A volumetric model is an entity that can provide a value at each point (voxel) in 3D space (Wang and Kaufman 1995). Volumetric data can be acquired directly, using computerised tomography scanners (CT), through voxelisation of surface data or using a procedural definition. Usually, volume based representation results in accurate description of real objects and is mostly used in medical applications. Volumetric models require large storage space and long rendering times.

Parametric surfaces such as B-spline or Bézier surfaces are higher-degree approximations to real surfaces and are capable of accurate representation of complex 3D surface boundaries (see figure 4.1a) (Mortenson 1985). They are particularly versatile in representing smooth curved objects. Detailed 3D models constructed by combining several surface patches can be interactively manipulated through a handful of control points and require limited storage space. Ease of interactive manipulation enables user-friendly modelling and animation and makes parametric surfaces very popular among computer graphics practitioners. Many CG applications incorporate parametric surfaces for the development of detailed product design models, suitable for manufacturing. Parametric representation is also suitable for the development of flexible cartoon characters and generally anthropomorphised animated objects (e.g. walking cubes and spheres).

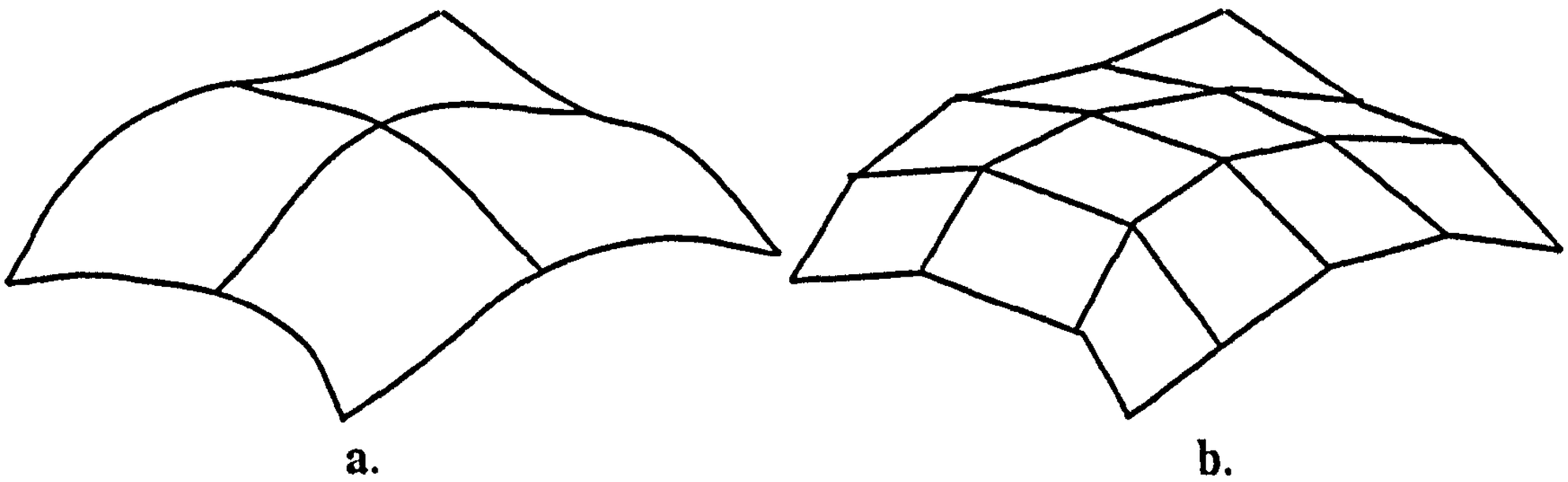


Figure 4.1 a. Parametric, b. polygonal representation of a surface.

However, parametric surfaces are normally defined by a regular quadrilateral or triangular grid of control points which renders them not very suitable for representation of complex 3D organic

objects.

Polygonal representation offers a simpler and more basic approach to 3D shape construction (see figure 4.1b). Polygons are first degree linear approximations to surfaces. So, to accurately represent smooth surfaces usually a large number of polygons is required. This results to large storage space requirements for 3D polygonal objects. However, accuracy of representation can be increased or decreased accordingly using polygon count reduction techniques (Turk 1992), (Algori and Schmitt 1996). Polygonal objects are widely used in CG applications because they offer computational efficiency rather than modelling accuracy. Contributing to this is a wide availability of graphics hardware customised for fast rendering of triangles. Virtual reality and interactive entertainment application are mostly based on such hardware and use reduced polygon count object descriptions as a compromise in order to achieve real-time performance. Many of the algorithms involved in image synthesis such as visible surface determination, shading calculations, etc. are optimised for triangulated polygonal data. Algorithms that deal with object collision detection and object penetration avoidance are also more efficient using triangulated data. Polygonal objects are easier to convert between different file formats and are more portable from system to system. Recently, laser based digitising technology has enabled the direct transfer of scaled or even life size real world models, into polygonal meshes. 3D objects obtained in this way inherit the natural detail from the real model. This level of detail is very difficult to achieve using conventional modelling techniques with parametric surfaces. Although sometimes inefficient, digitised objects are very useful for realistic organic character modelling. Deforming polygonal data is algorithmically simpler as opposed to parametric data. However, it is much more difficult to achieve the same quality of smooth deformation, as one has to deform every vertex of the polygonal object, as opposed to a small number of control points in parametric surfaces.

The proposed deformation methodology, being independent of geometric representation, can easily be adapted to operate either on parametric control points, on voxels or directly on polygon vertices. For the sake of brevity, the discussion on 3D object representation will be concentrated on a general description of polygon meshes and polygonal solids.

4.3 Polygonal Representation

The building blocks of polygonal representation are simple topological elements such as vertices, edges and faces (see figure 4.2).

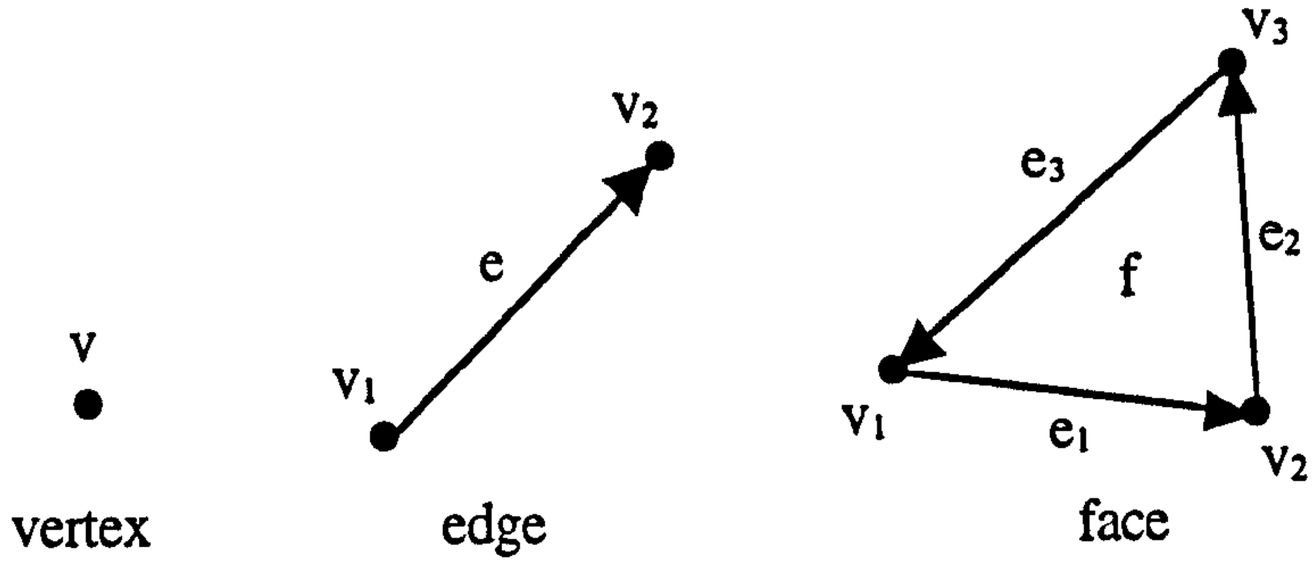


Figure 4.2 Topological elements.

A vertex is represented by a point in 3D space. An edge is an orientated straight line segment, defined by an ordered pair of vertices. A face is a planar polygon which consists of a closed sequence of edges. A minimum of three edges is needed for the construction of a face. However, faces with more than three edges may be used to represent concave shaped faces or even faces with holes.

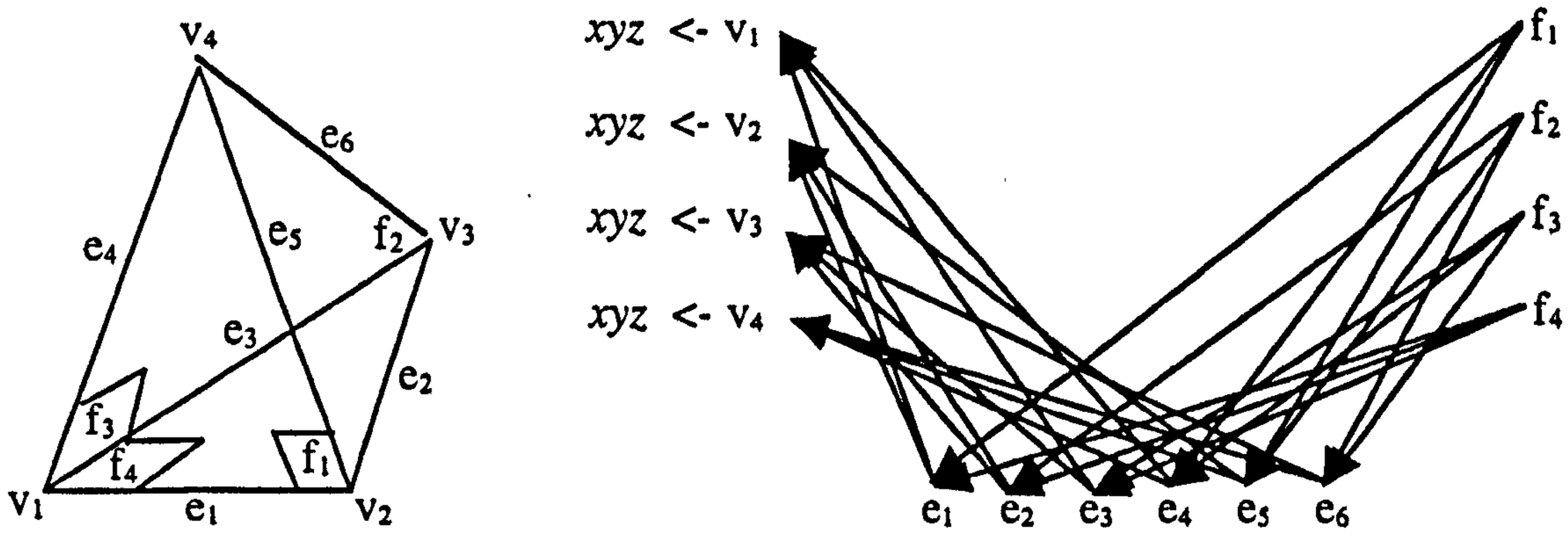


Figure 4.3 Polygonal connectivity of a tetrahedron.

The polygonal description of a 3D object is the set of data that describe how vertices, edges and faces are related to each other. This set of data usually consists of a list of vertices, a list of

edges and a list of faces interconnected to each other. Each vertex in the vertex list points to a triplet of x, y, z coordinates. Each edge in the edge list points to its two vertices. Each face in the face list points to an ordered loop of all the edges and or all the vertices of its contour (see figure 4.3).

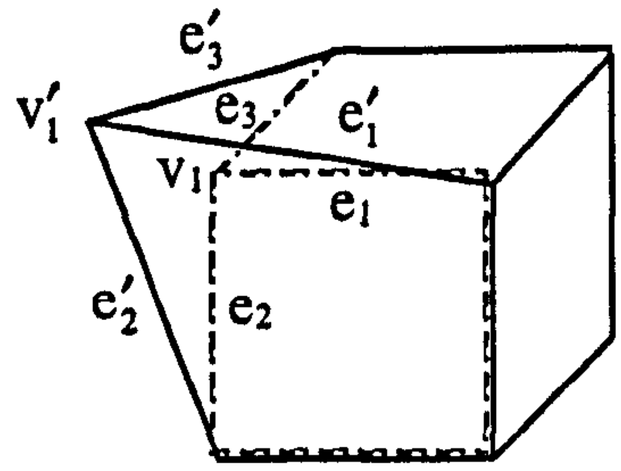


Figure 4.4 Change of edge lengths and vertex coordinates results in object deformation.

During a deformation, vertex coordinates and edge lengths of an object may change but the polygonal connectivity remains intact (see figure 4.4).

4.3.1 The Polygon Mesh

In a polygon mesh vertices, edges and faces are connected in such a way that the following rules are obeyed (see figure 4.5):

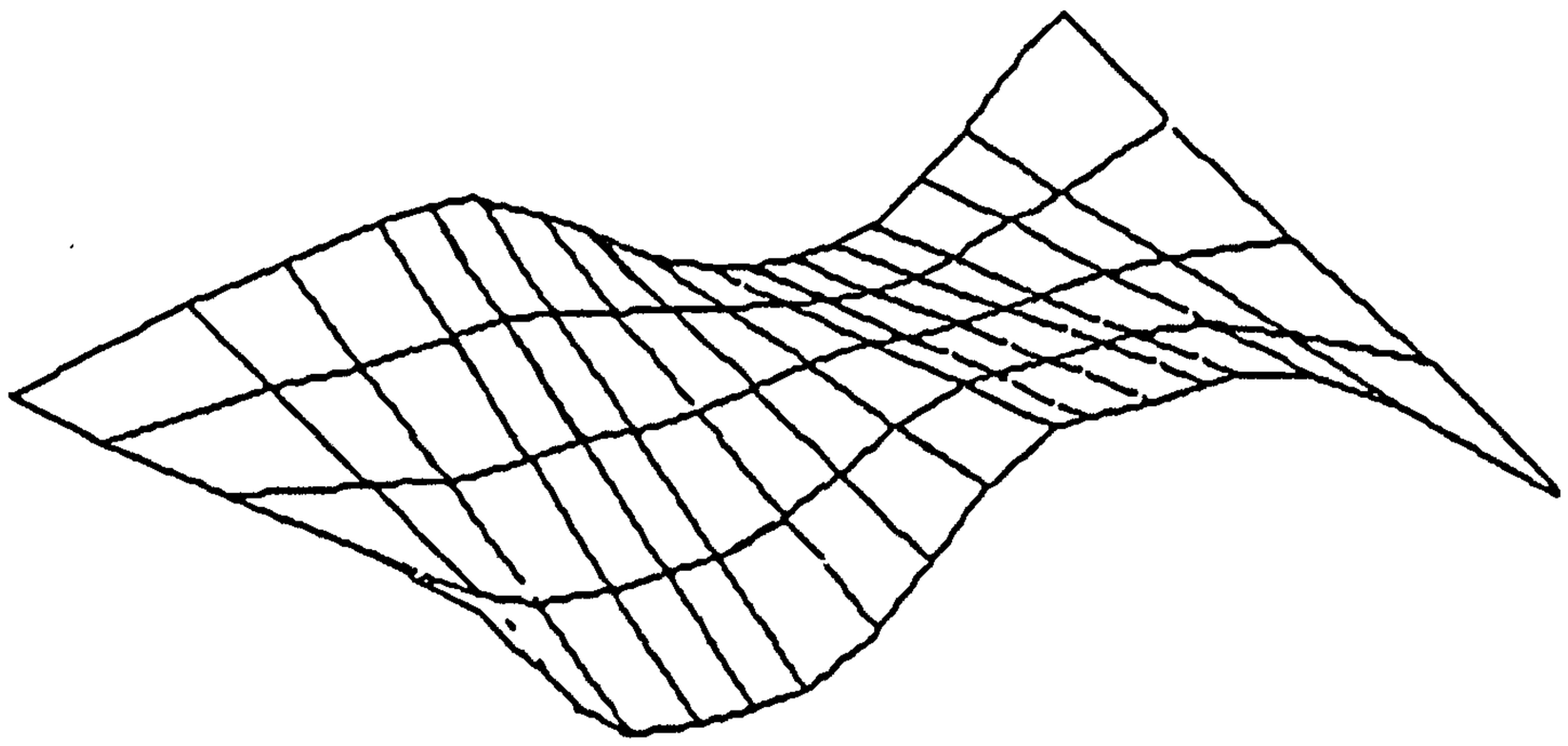
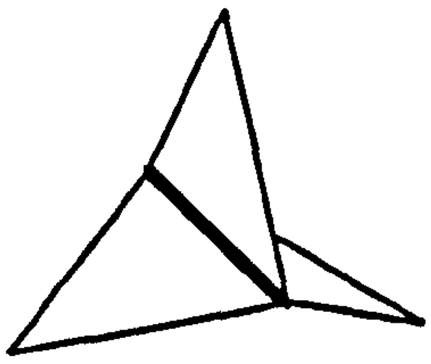


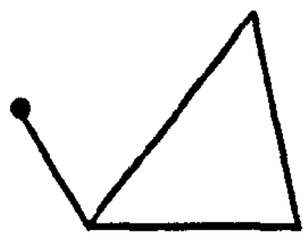
Figure 4.5 A simple polygon mesh.

- an edge must be shared by at most two adjacent faces,

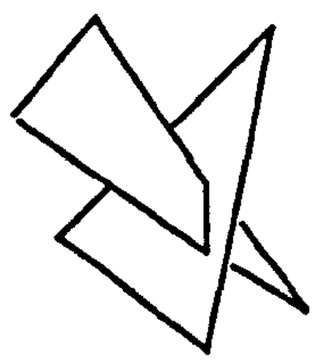
- a vertex must be shared by at least two edges,
- all faces must be closed planar polygons,
- faces must not interpenetrate.



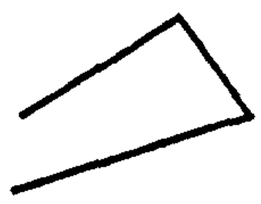
a. An edge is shared by three faces.



b. A vertex belongs to only one edge.



c. Two faces interpenetrate.



d. An open polygon.

Figure 4.6 Four cases of invalid polygonal connectivity.

Polygon meshes offer an efficient and versatile approach to 3D object representation and are suitable for modelling a variety of objects such as cloth, paper, leaves, water surface, etc.

4.3.2 The Polygonal Solid

A polygonal solid (or polyhedron) is an object that is completely bounded by a set of planar polygons (see figure 4.7). Restrictions must also be enforced:

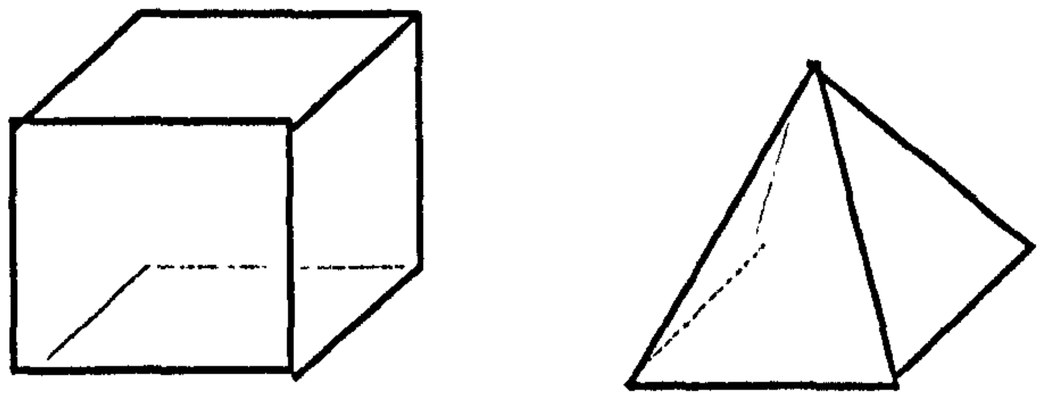


Figure 4.7 Simple polygonal solids.

- each edge must connect two vertices,
- each edge must be shared by exactly two faces,
- at least three edges must meet at each vertex,
- faces must be closed polygons,
- faces must not interpenetrate.

Euler's formula expresses a relationship among the number of vertices, edges, faces and holes on faces of a polygonal solid:

$$V - E + F - H = 2(C - G) \qquad [4.1]$$

where V is the number of vertices, E is the number of edges, F is the number of faces, H is number of holes in faces, C is the number of separate components of the object and G is the genus of the object (equal to the number of holes that pass entirely through the object as in a torus) (see figure 4.8).

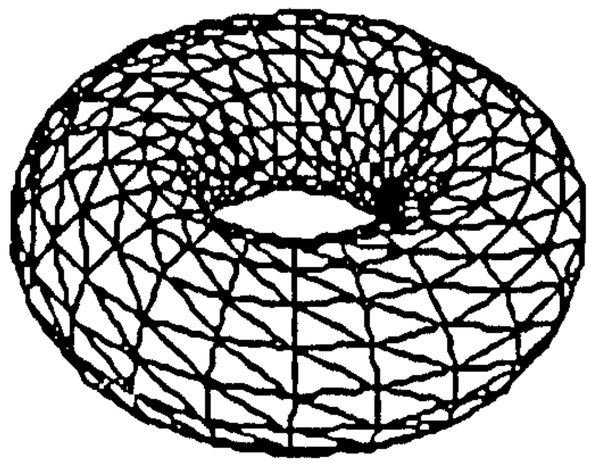


Figure 4.8 A polyhedron of genus 1.

Polygonal solids and polygon meshes may be constructed using various geometric modelling techniques, such as extrusion of 2D polygons, skinning of curves, solid rotation, sweep along curves, etc. Several polygonal objects may be combined to produce detailed representations of complex 3D characters (see figure 4.9a and b). Polygonal data can also be obtained using 3D digitising devices (see figure 4.9c).

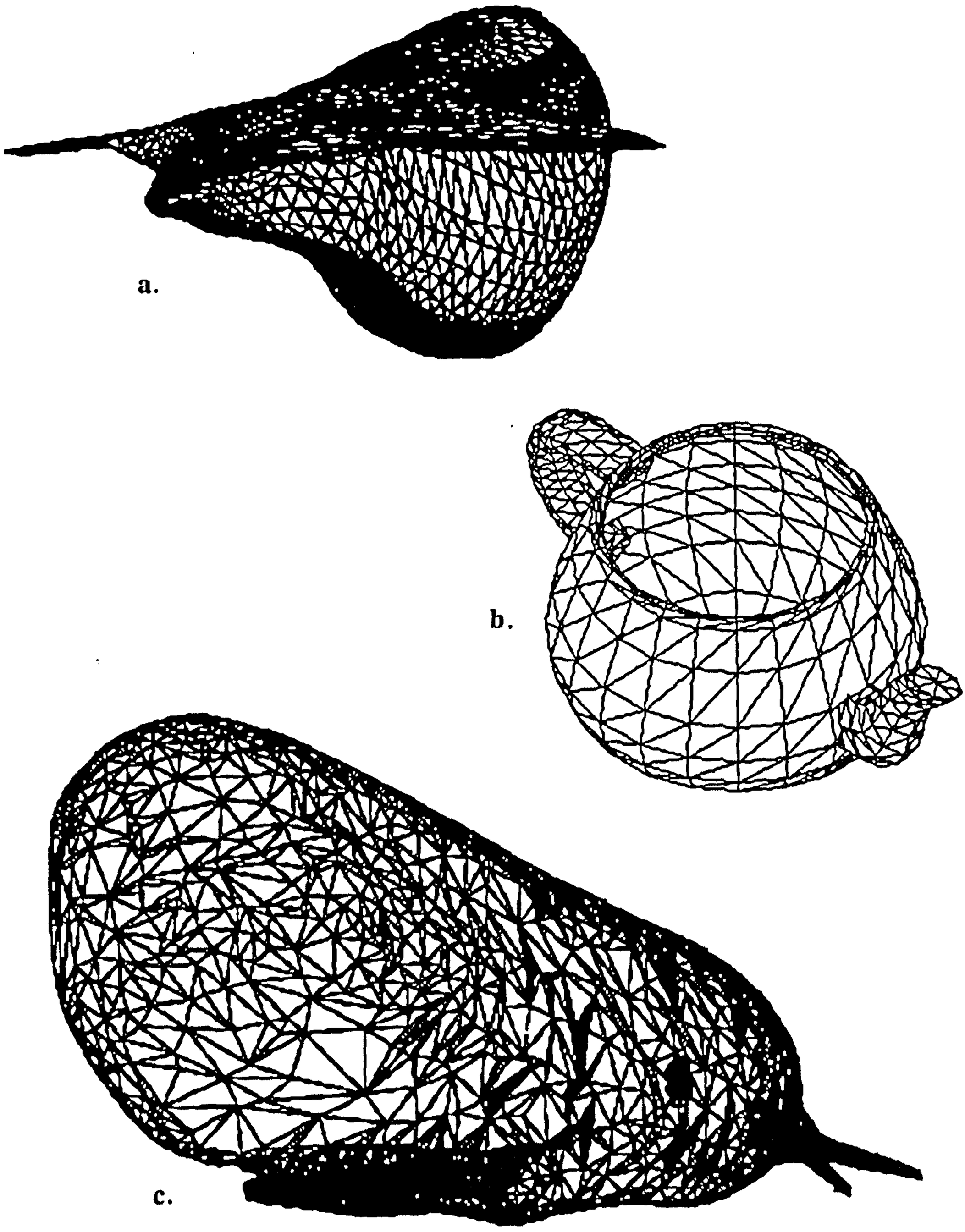
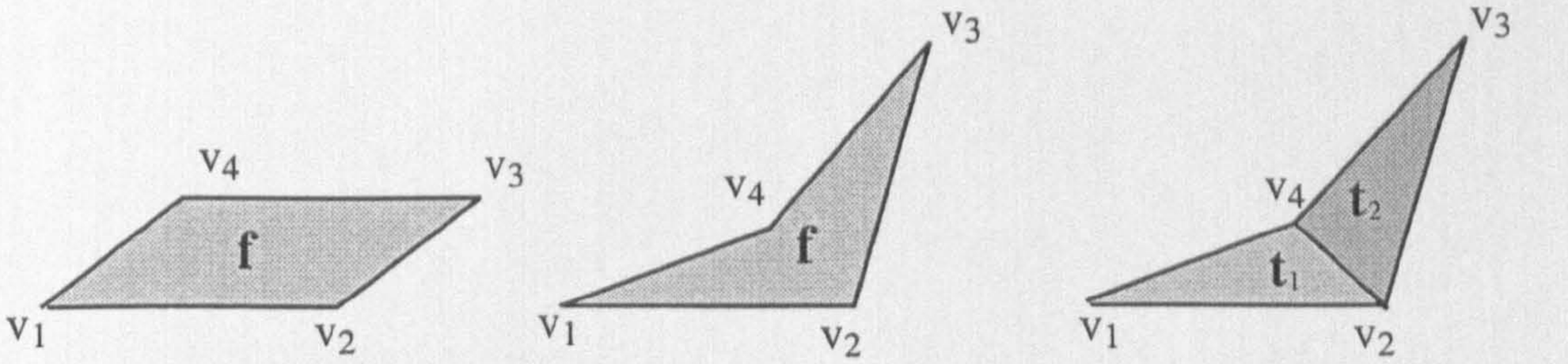


Figure 4.9 Complex polygonal objects.

4.4 Triangulation

Faces in polygonal representation may consist of closed polygons of three or more vertices. It is assumed, though, that all the vertices of a plane are coplanar. In the process of a deformation the shape of a face may be distorted and its vertices, if there are more than three, may become non-coplanar (see figure 4.10).



a. undeformed face **b**. deformed face **c**. face split into two triangles

Figure 4.10 A deformed face with more than three vertices.

A deformed face such as the one in figure 4.10b, is likely to cause problems in CG algorithms that make use of the plane equation and the plane normal of a face. In image synthesis, for instance the shading calculation process may resort to an averaged plane equation with all four vertices contributing to the equation. This may cause inaccurate shading. However, if the face were split into two triangles (see figure 4.10c) this problem would not occur. A triangle is immune to problems arising from deformation simply because its three vertices are always coplanar (as long as the three vertices are prevented from becoming collinear or coincident).

Triangulated objects are frequently used in CG applications which incorporate deformation techniques, collision detection techniques, etc. Most of these algorithms have been optimised for triangles and use triangles as the basic building block for 3D objects. Most CG applications that use parametric surface representation for modelling 3D objects, convert surface data into triangulated polygonal data, just before the rendering process. This is done to take advantage of accuracy in modelling offered by parametric surfaces and of rendering efficiency offered by triangulated polygonal objects.

A simple triangulation algorithm was developed by the candidate (see appendix A). This algorithm is not the most efficient one, neither it produces optimised triangulations. Triangulation, has been the focus of many research efforts in the past (Fournier and Montuno 1984), (Chazelle 1991), (Ronfard and Rossignac 1994). Some of them attempt to optimise the total number of triangles or the average area or edge lengths of the triangles, where others focus on algorithmic efficiency.

4.5 Summary and Conclusion

Parametric surfaces are suitable for representing smooth curved objects, can easily be manipulated through few control points and require limited storage space. Parametric surfaces, though, pose problems in representing complex 3D organic objects and require computationally expensive implementations. Triangulated polygonal objects require more storage space and pose problems in the development of smooth deformations. Their real advantage, though, is that they are based on algorithmically simple concepts which lead to efficient implementations suitable for real-time CG applications. Modelling of organic characters based on living creatures can be achieved by digitising real world models into polygonal data. 3D objects can, thus, inherit their natural detail from real models which is otherwise difficult if not impossible to achieve. This facilitates the development of realistic looking characters and shifts the emphasis towards the animation process.

The following chapter introduces a local deformation model.

5

Vector Offset Operators for Local Space Deformation

5.1 Introduction

In this chapter a new method for manipulating deformable objects is presented. The proposed methodology is structured around a core local space deformation model. An interactive strain application tool is used to control and constrain local deformation in 3D space.

The objectives of this chapter are:

- to introduce basic background theory of local deformations,
- to develop a general and efficient local deformation model,
- to enable intuitive interaction with deformable objects,
- to enable application of spatial constraints on deformable objects.

Key design features for the development of the deformation model are efficiency, generality, intuitive interaction and spatial constrainability. Speed of deformation is essential for effective interactive manipulation and efficient animation testing and rendering of complex deformable objects. Spatial constraints are useful for applying detailed deformation control on complex 3D objects. Additionally, the model must be general in the sense that it is not dedicated to a specific object representation approach. This ensures portability and extendibility and, thus, renders the model suitable for further modular development.

5.2 Local Space Deformation

3D deformation is the process that results to modification of the shape of an object. For instance, when deforming a polygonal object vertex coordinates, edge lengths and face shapes are modified.

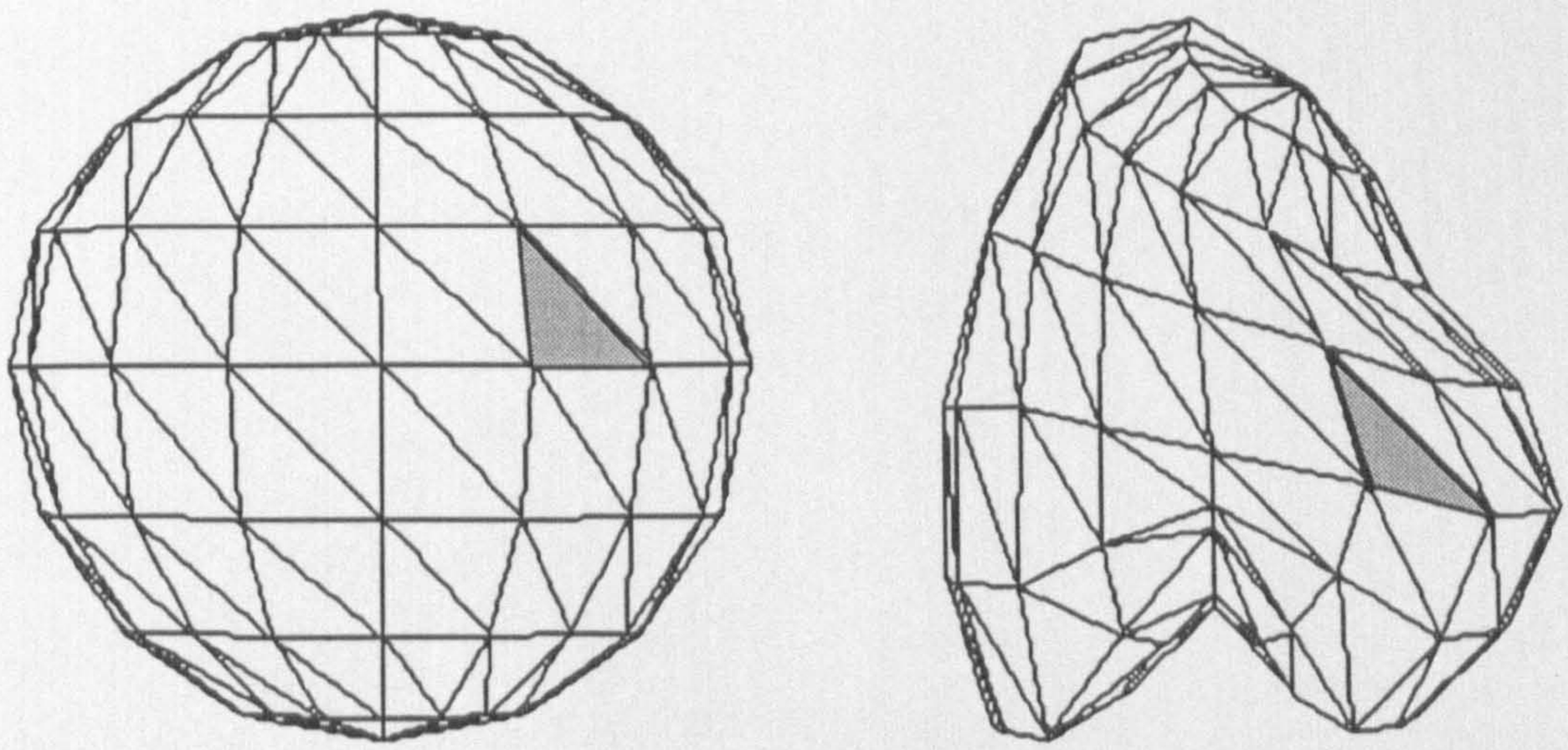


Figure 5.1 Deforming a triangulated object.
Each triangle is subjected to alterations of vertex positions and edge lengths.

The shape and geometry of the whole object will change but the connectivity among topological elements will remain unaltered (see figure 5.1). Space deformation may be applied to polygonal, parametric or volumetric data. Polygonal data may be directly deformed by modifying vertex coordinates whereas parametric data can be deformed indirectly by modifying control point coordinates (Mortenson 1985). Volumetric data can be similarly deformed by modifying voxel positions in space (Lerios *et al.* 1995).

5.2.1 Global Frame of Reference

In the case of polygonal objects vertices, edges and faces are usually defined using a 3D Cartesian coordinate system as a global frame of reference. A 3D Cartesian coordinate system is defined by three mutually perpendicular unit vectors **i**, **j** and **k** and an origin point, O. Each

vertex v_i of the object (where subscript $_i$ denotes vertex number) is defined by a position vector, $\mathbf{r}_i$, which starts at the origin, O, of the global coordinate system and ends at vertex v_i, (see figure 5.2).

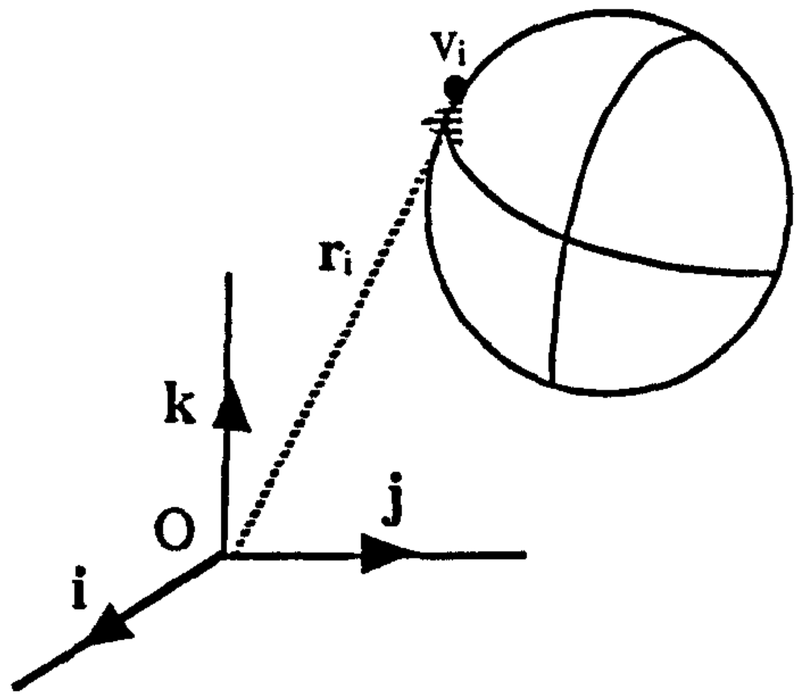


Figure 5.2 An object defined using a global frame of reference.

The components of vector $\mathbf{r}_i$, are also the global coordinates of vertex v_i:

$$\mathbf{v}_i = (v_{xi}, v_{yi}, v_{zi}), \quad \mathbf{r}_i = (r_{xi}\ r_{yi}\ r_{zi}) \qquad [5.1]$$

where

$$v_{xi} = r_{xi},\ v_{yi} = r_{yi},\ v_{zi} = r_{zi}$$

5.2.2 Local Frame of Reference

A local frame of reference may be used to define local space. For this purpose, a local coordinate system is defined using three mutually perpendicular unit vectors **u**, **v** and **w** and the local origin, L (see figure 5.3).

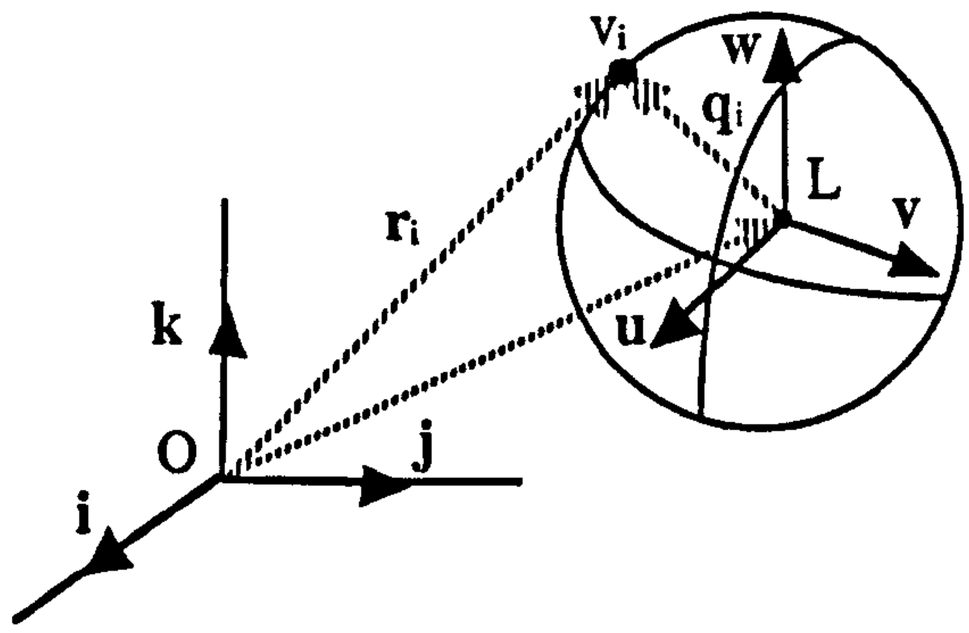


Figure 5.3 An object defined using a local frame of reference.

An object may be mapped from global to local space. So, the same vertex v_i can be defined by a global position vector $\mathbf{r}_i$, as well as by a local position vector $\mathbf{q}_i$ (see figure 5.3). The latter vector $\mathbf{q}_i$ starts at the local origin, L and ends at vertex v_i. Vector $\mathbf{q}_i$ is given by:

$$\mathbf{q}_i = \mathbf{r}_i - \mathbf{OL} \qquad [5.2]$$

where **OL** is the global position vector of the local origin, L.

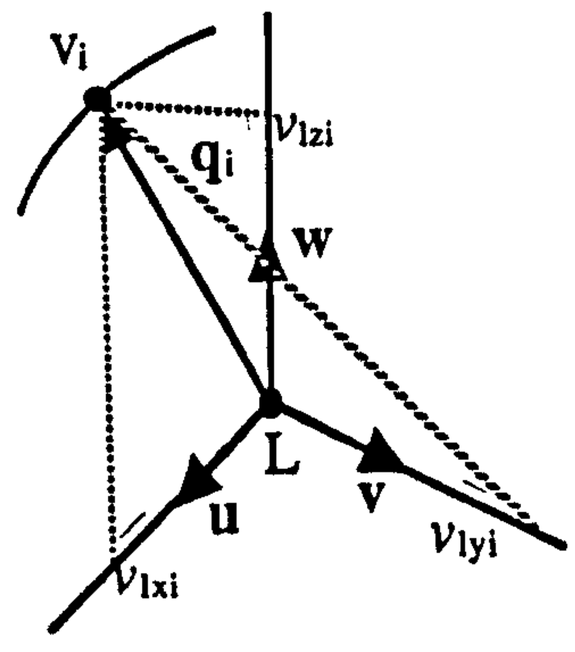


Figure 5.4 Local position vector $\mathbf{q}_i$ and vertex v_i in local coordinates.

Vector $\mathbf{q}_i$ must then be projected onto vectors **u**, **v** and **w** of the local frame in order to produce the local coordinates of vertex v_i (see figure 5.4):

$$\mathbf{v}_i = (v_{lxi},\ v_{lyi},\ v_{lzi})$$

where

$$v_{lxi} = \mathbf{q}_i \circ \mathbf{u}, \quad v_{lyi} = \mathbf{q}_i \circ \mathbf{v}, \quad v_{lzi} = \mathbf{q}_i \circ \mathbf{w} \qquad [5.3]$$

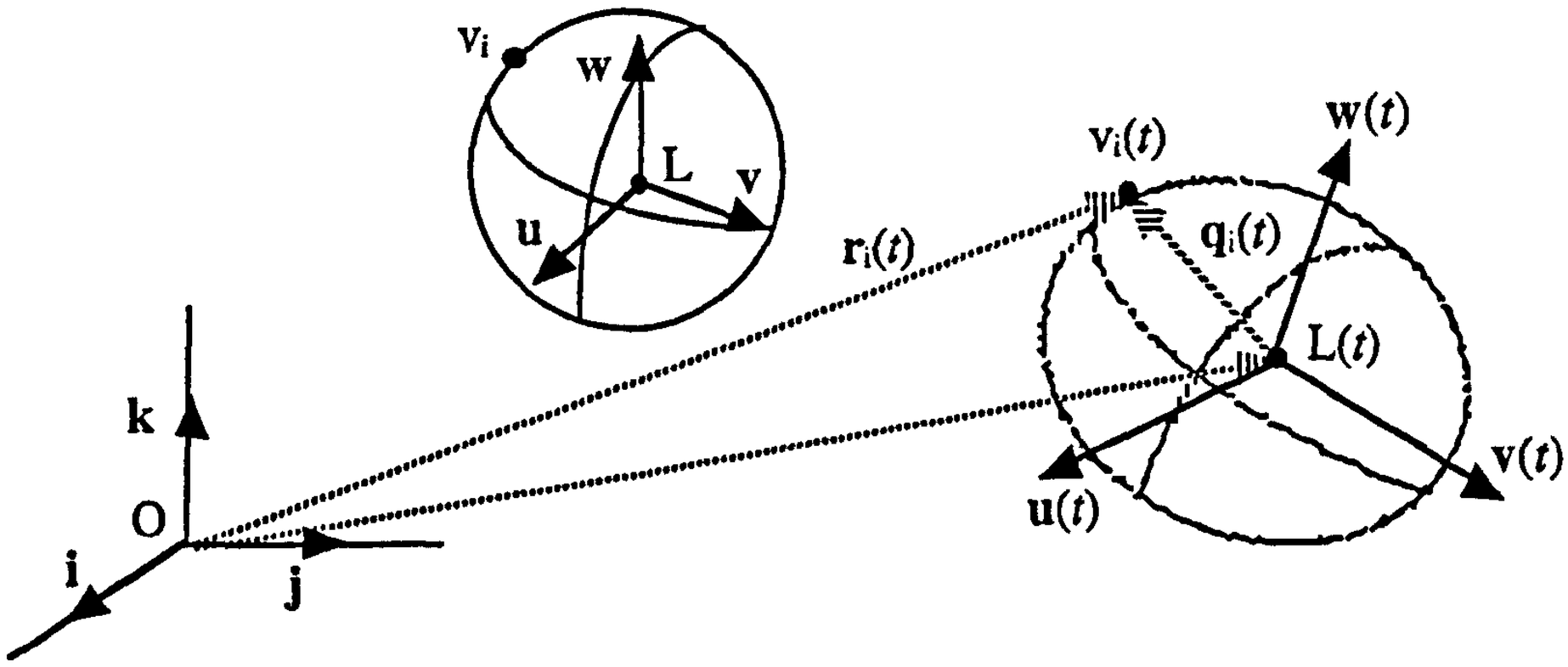


Figure 5.5 An object at rest state and at time *t*.

Local space may undergo geometric transformations such as translation, scaling and rotation. Transformations may develop over time and vectors **u**, **v**, **w** and the local origin, L, become time dependent: **u**(*t*), **v**(*t*), **w**(*t*) and L(*t*) (see figure 5.5). Local space transformations may be applied to objects. A vertex v_i can be made to follow an animated local frame of reference and inherit local transformations becoming $v_i(t)$. The local position vector $\mathbf{q}_i$ also becomes time dependent, $\mathbf{q}_i(t)$ (see figure 5.5):

$$\mathbf{q}_i(t) = \mathbf{r}_i(t) - \mathbf{OL}(t) \qquad [5.4]$$

5.2.3 Transformation Vector

In figure 5.6 v_i and $v_i(t)$ represent the same vertex at rest state and at time *t* respectively. The vector that starts at v_i and ends at $v_i(t)$ represents the total transformation that v_i has received at time *t*.

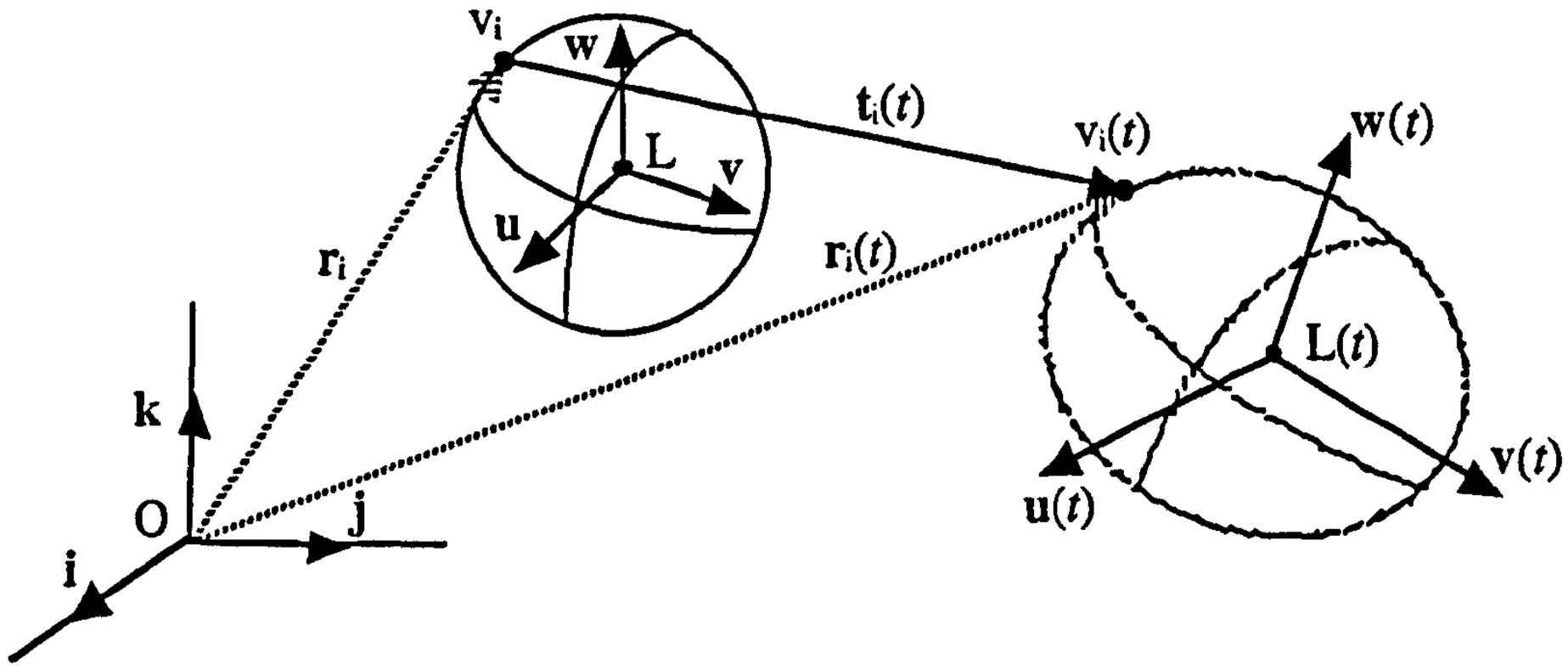


Figure 5.6 The total transformation vector $\mathbf{t}_i(t)$ for vertex, v_i.

Vector $\mathbf{t}_i(t)$ may be calculated as the difference between the two global position vectors $\mathbf{r}_i$ and $\mathbf{r}_i(t)$:

$$\mathbf{t}_i(t) = \mathbf{r}_i(t) - \mathbf{r}_i \qquad [5.5]$$

Combining equations [5.2] and [5.4], equation [5.5] becomes:

$$\mathbf{t}_i(t) = (\mathbf{OL}(t) + \mathbf{q}_i(t)) - (\mathbf{OL} + \mathbf{q}_i)$$

$$\Rightarrow \quad \mathbf{t}_i(t) = (\mathbf{OL}(t) - \mathbf{OL}) + (\mathbf{q}_i(t) - \mathbf{q}_i)$$

$$\Rightarrow \quad \mathbf{t}_i(t) = \mathbf{T}(t) + \mathbf{Q}_i(t) \qquad [5.6]$$

where

$$\mathbf{T}(t) = \mathbf{OL}(t) - \mathbf{OL} \qquad [5.7]$$

$$\mathbf{Q}_i(t) = \mathbf{q}_i(t) - \mathbf{q}_i \qquad [5.8]$$

Vector $\mathbf{Q}_i(t)$, in equation [5.8], is equal to the difference between the two local position vectors $\mathbf{q}_i$ and $\mathbf{q}_i(t)$. This vector represents a composition of rotation and scaling that is applied to vertex v_i in local space. In equation [5.7], vector $\mathbf{T}(t)$ represents translation in global space. This

translation vector is applied equally to all the vertices of the object and it, thus, causes rigid body motion. It should be noted that rigid body motion usually consists of a mixture of translation and rotation in local or global space. Therefore, a complete study of rigid body dynamics ought to include linear and angular velocity and momentum. However, since the emphasis of this thesis lies on a deformable model with physical properties such as elasticity and viscoelasticity, a complete and in depth analysis of rigid body dynamics will not be carried out.

5.2.4 Strain Vector

The magnitude and direction of vector $\mathbf{Q}_i(t)$, may by varied from vertex to vertex inside the same object, thus, producing an arbitrary deformation. Particularly, if a gradient of transformation amounts were applied to vector $\mathbf{Q}_i(t)$ the result would be a smooth deformation of the object (see figure 5.7).

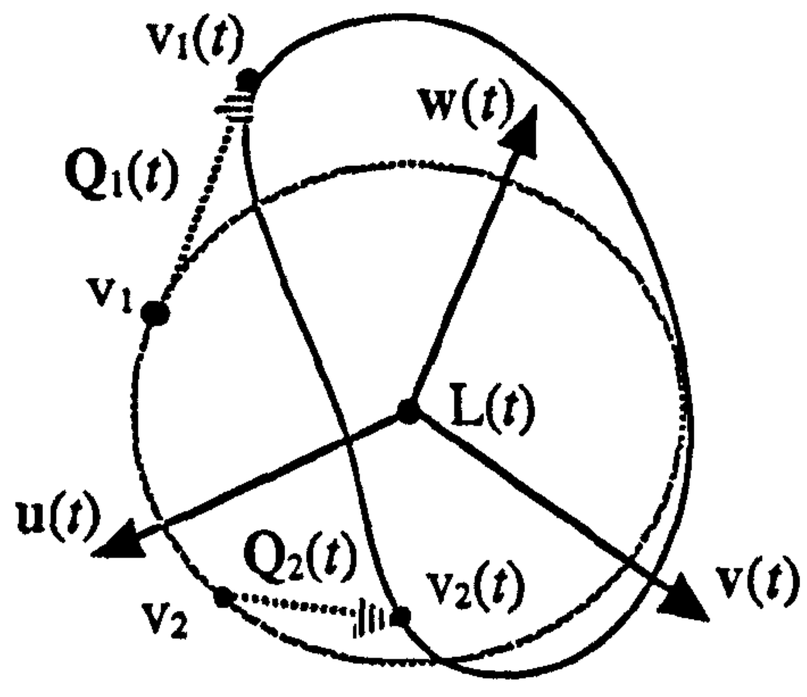


Figure 5.7 An object at rest state and deformed at time *t*.

The object in figure 5.7, does not receive any global translation. Notice though that at time *t*, vertices v_1 and v_2, receive vectors $\mathbf{Q}_1(t)$ and $\mathbf{Q}_2(t)$ respectively which differ in direction and magnitude. As a result the object is deformed. In general, vector $\mathbf{Q}_i(t)$ may be called the strain vector (or local deformation vector) of vertex v_i. Direct and dynamic manipulation of the strain vector $\mathbf{Q}_i(t)$ can lead to versatile and efficient local deformation models which are capable of

achieving fine control over the shape of the deformation.

In the past, several researchers have suggested a variety of deformation models based on the concepts described in this paragraph. Barr (1984) suggested a global and local deformation model based on the application of affine transformations to surface normals (see section 2.2.1). Magnenat-Thalmann and Thalmann (1987) introduced the concept of Joint dependent Local Deformation (JLD) operators which specify local deformation depending upon the nature of joints of articulated structures (see section 2.4.6). Witkin and Welch (1990) extended Barr's model by incorporating the property of mass and idealised elasticity and by applying attachment constraints to 3D objects (see section 2.4.1). Beier and Neely (1992) used local deformation for 2D feature based image metamorphosis (see section 2.3.1). Finally, Lerios *et al.* (1995) used local deformation for featured based volume metamorphosis (see section 2.3.1). In the following paragraphs a new method for controlling local space deformation through manipulation of the strain vector is presented.

5.3 Vector Offset Operators

Effective and efficient control of the shape of deformation is critical for a successful local deformation model. One such model should be capable of achieving fine manipulation in 3D space, as well as constrainable manipulation over time. The suggested approach introduces a new tool suitable for specifying a local deformation in 3D space. This tool provides an efficient and user-friendly approach to spatio-temporal control of local deformation.

5.3.1 Definition of a Vector Offset Operator

A Vector Offset Operator (VOO) is a tool for interactive or time based 3D strain application. A rigid pin immersed inside a piece of mouldable material may be thought of as the physical counterpart of a VOO. Any movement of the VOO pin will cause strain on the surrounding material. The functionality of a VOO is clearly illustrated in the colour plates 5.45 through to

5.56, in section 5.11 at the end of this chapter. The VOO pin may be capable of causing strain through its head and tip only or through its entire body length (see figure 5.8).

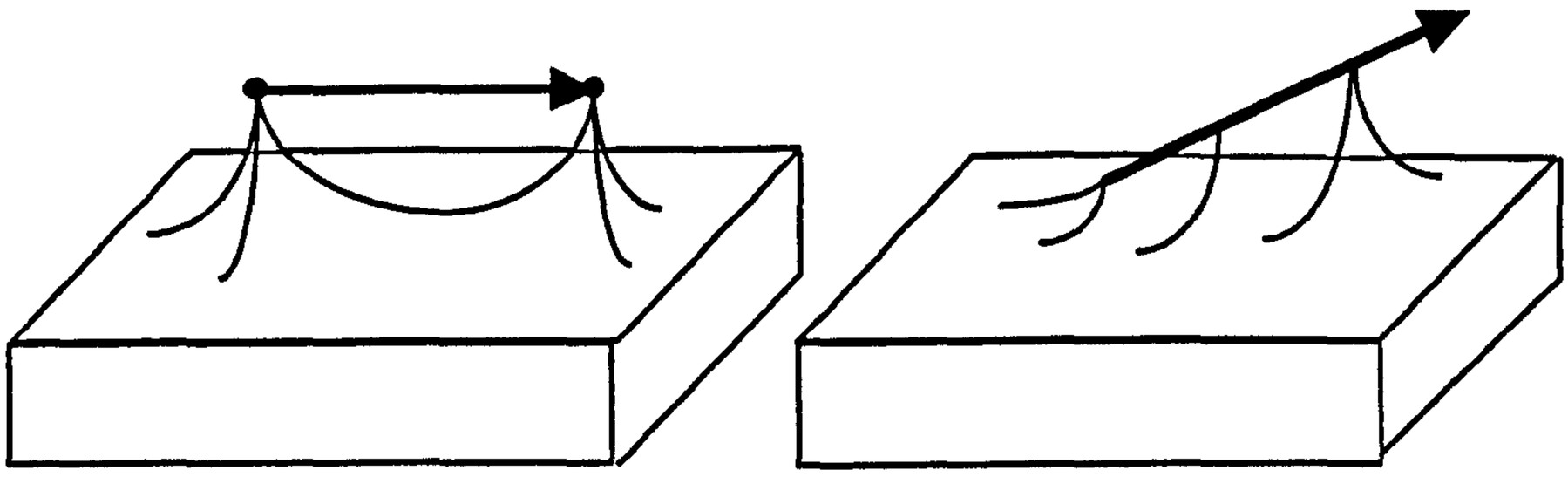


Figure 5.8 A VOO pin immersed in a piece of mouldable material.

A VOO consists of a fixed vector with magnitude, direction and position in 3D space. Two points define the Head and the Tip of a VOO (see figure 5.9). A VOO is an independent entity which is used for the construction of a local frame of reference.

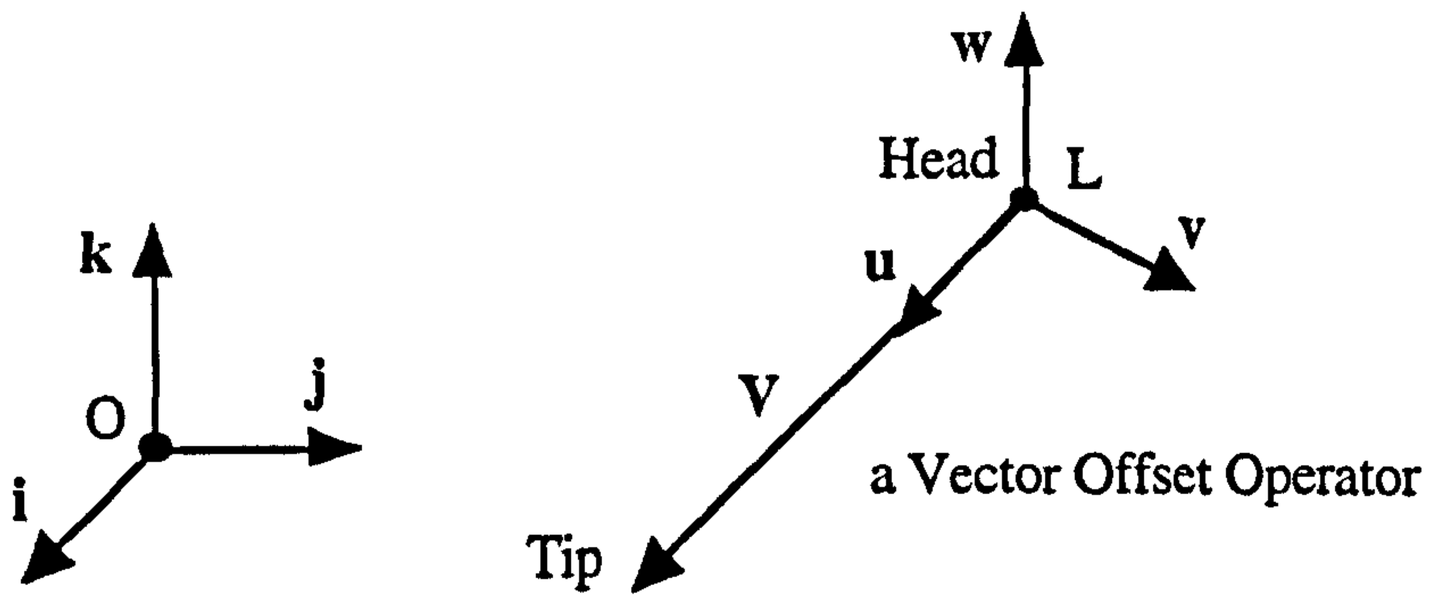


Figure 5.9 A VOO and a local frame of reference.

The origin of the local frame is located at the Head of the VOO. The unit vectors **u**, **v** and **w** of the local frame correspond to vectors **i**, **j** and **k** respectively and may be calculated using the following pseudo-code:

$$\mathbf{u} = \frac{\mathbf{V}}{|\mathbf{V}|} \qquad [5.9]$$

$$\text{if } \mathbf{u} = \mathbf{k} \text{ then } \mathbf{v} = \mathbf{j};\ \mathbf{w} = -\mathbf{i} \qquad [5.10]$$

$$\text{else if } \mathbf{u} = -\mathbf{k} \text{ then } \mathbf{v} = \mathbf{j};\ \mathbf{w} = \mathbf{i} \qquad [5.11]$$

$$\text{else } \mathbf{v} = \mathbf{k} \otimes \mathbf{u};\ \mathbf{v} = \frac{\mathbf{v}}{|\mathbf{v}|};\ \mathbf{w} = \mathbf{u} \otimes \mathbf{v} \qquad [5.12]$$

In equation [5.9], the VOO is normalised to produce vector **u**. Lines [5.10] and [5.11] deal with the special case where vector **u** is parallel or opposite to vector **k** of the global frame of reference. In such case vectors **v** and **w** are given default directions. Line [5.12] deals with the general case where **u** has an arbitrary direction, neither parallel nor opposite to **k.** In this case vectors **v** and **w** can be calculated as cross-products between **k** and **u** and between **u** and **v** respectively.

5.3.2 Manipulation of a VOO

Manipulating a VOO in 3D space facilitates the application of affine transformations to local space. A change of magnitude and direction of a VOO leads to a redefinition of its local frame of reference.

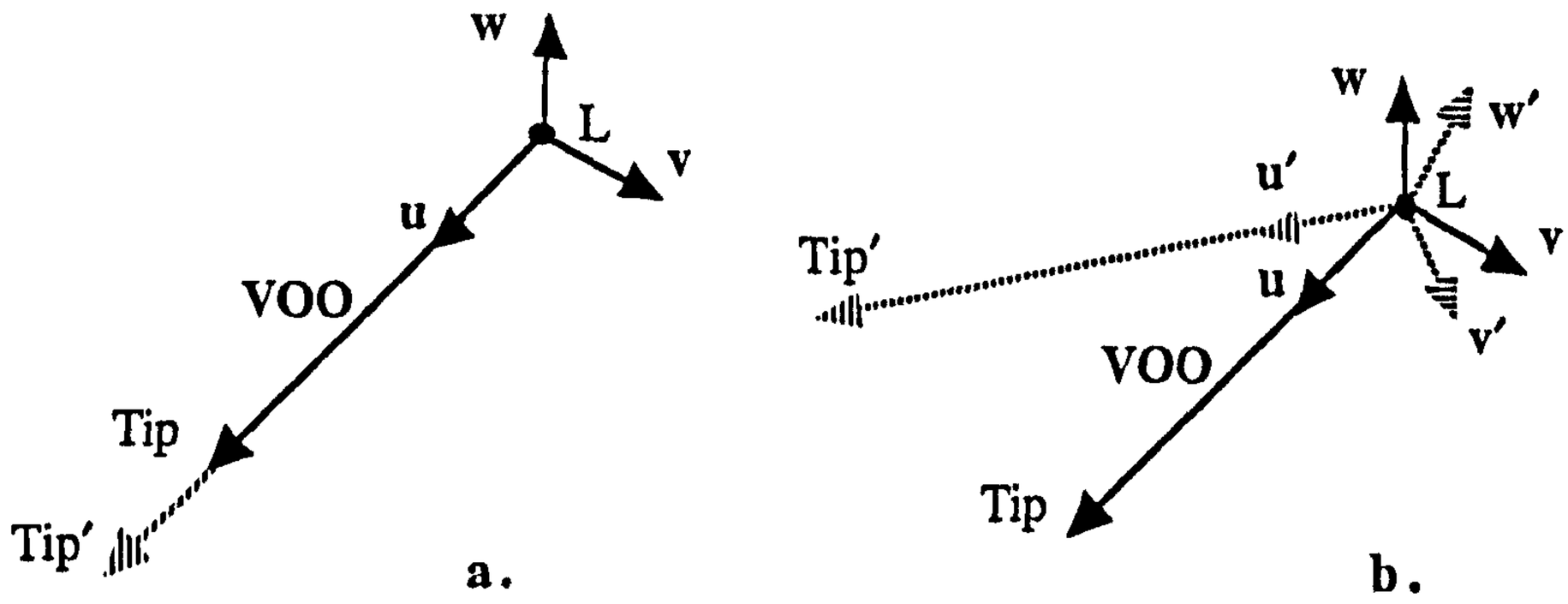


Figure 5.10 Offsetting the tip of a VOO.

Direct offsetting (hence the name "offset operators") of the the tip of a VOO may cause a change in magnitude of the VOO itself. Although, a change in magnitude alone does not affect the local frame of reference it does cause lengthening or shortening of a VOO (see figure 5.10a). Furthermore, offsetting the tip may cause a change in the direction of a VOO which is equivalent to an arbitrary rotation of the local frame about axes **v** and **w** (see figure 5.10b).

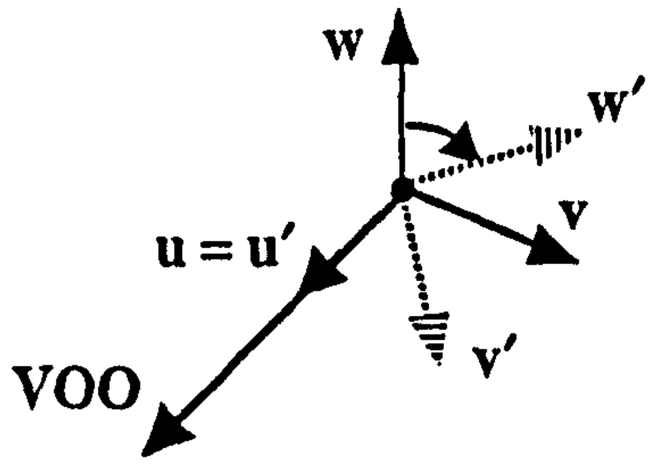


Figure 5.11 Rotation about axis **u**.

An extra degree of freedom is provided by allowing rotation about the VOO itself which causes the local frame to rotate about axis **u** (see figure 5.11).

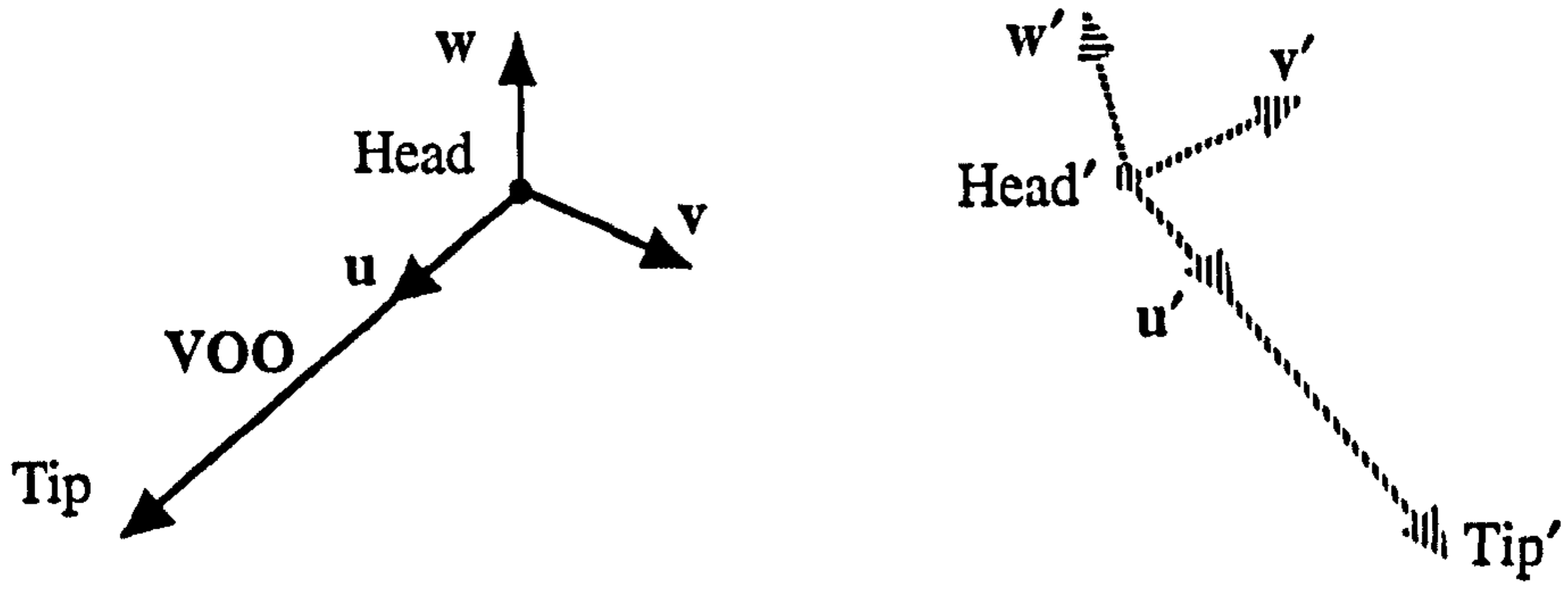


Figure 5.12 Offsetting both the tip and head of a VOO.

Offsetting the head of a VOO causes the local origin to move and the frame to receive an amount of global translation (see figure 5.12). A VOO may be manipulated interactively in space or animated in time by offsetting the head or the tip independently from one another or by rotating about the its main axis, **u**. Any of these operations must be followed by a redefinition of the local frame of reference.

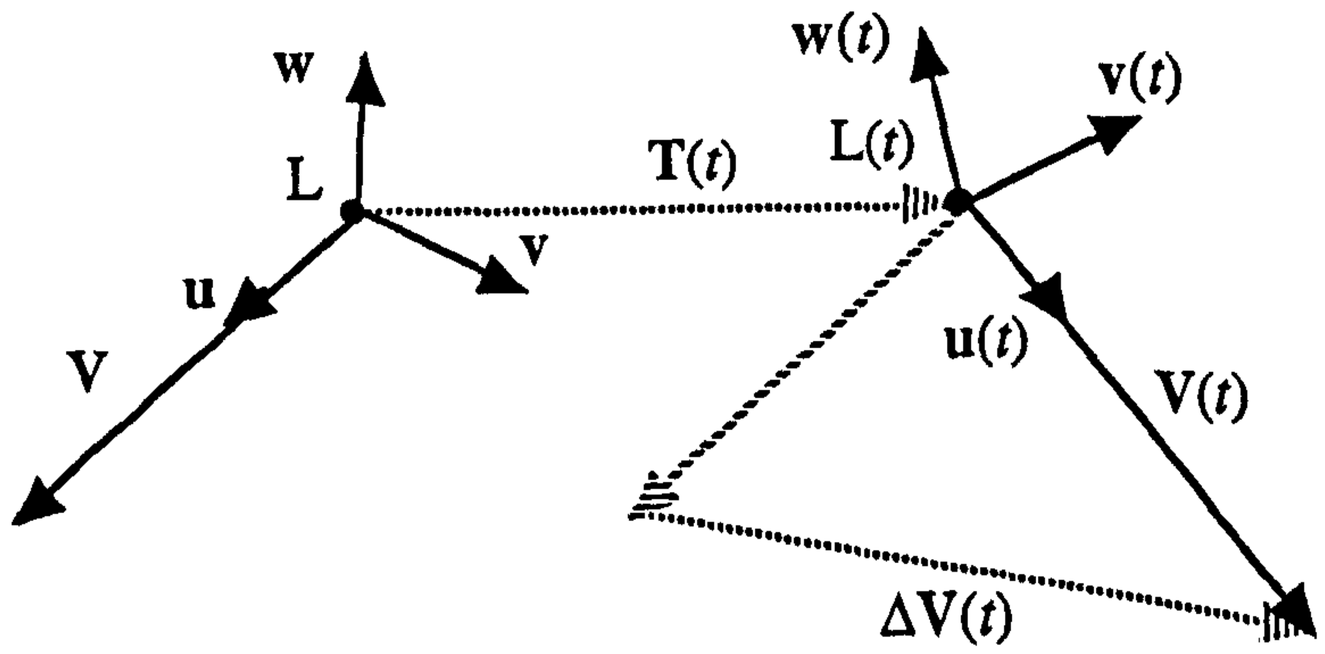


Figure 5.13 A VOO at rest state and at time *t*.

In figure 5.13, **V** represents a VOO at its rest state and **V**(t) at a time t, with both its head and tip at a new position in space. Vector **T**(t) represents the global translation of the frame and vector $\Delta\mathbf{V}(t)$ the local deformation of VOO:

$$\mathbf{T}(t) = \mathbf{OL}(t) - \mathbf{OL} \qquad [5.13]$$

$$\Delta\mathbf{V}(t) = \mathbf{V}(t) - \mathbf{V} \qquad [5.14]$$

Equation [5.13] is identical to equation [5.7] and equation [5.14] is equivalent to equation [5.8] which represents the strain vector of a vertex v_i (see section 5.2.3).

5.4 Application of Strain

In chapter 3 (see section 3.2.2), we saw that there are several basic types of strain: stretching, squashing, shearing, bending and twisting. A local deformation may consist of a combination of any of these types of strain. A VOO may be used to induce rigid body motion in the global space and deformation in the local space of a 3D object. To this end, a 3D object is initially being mapped from global space to the local frame of a VOO at rest state. This mapping produces local coordinates, $v_i = (v_{lxi}, v_{lyi}, v_{lzi})$, for every vertex of the object (see equation [5.3] in section 5.2.2). Subsequently, these local coordinates are subjected to a certain local transformation **M** and then they are used as local coordinates of vertex $v_i(t) = (v_{lxi}(t), v_{lyi}(t),$

$v_{lzi}(t)$) (see figure 5.14a and b).

$$\mathbf{v}_i(t) = \mathbf{M} \cdot \mathbf{v}_i \qquad [5.15]$$

The transformation matrix **M**, in equation [5.15], depends on the state of the VOO at time *t* (see equation [5.31] in section 5.4.6.4).

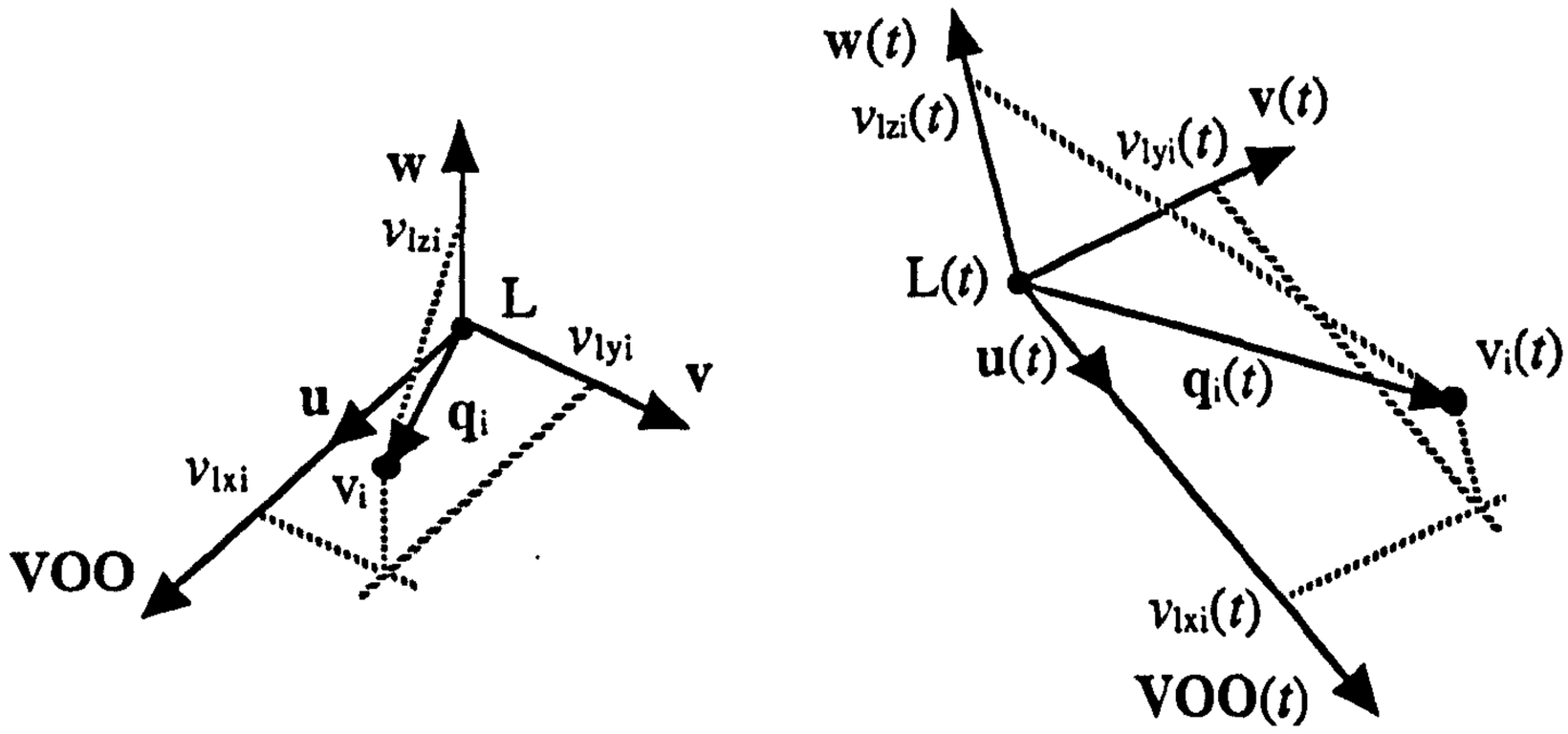


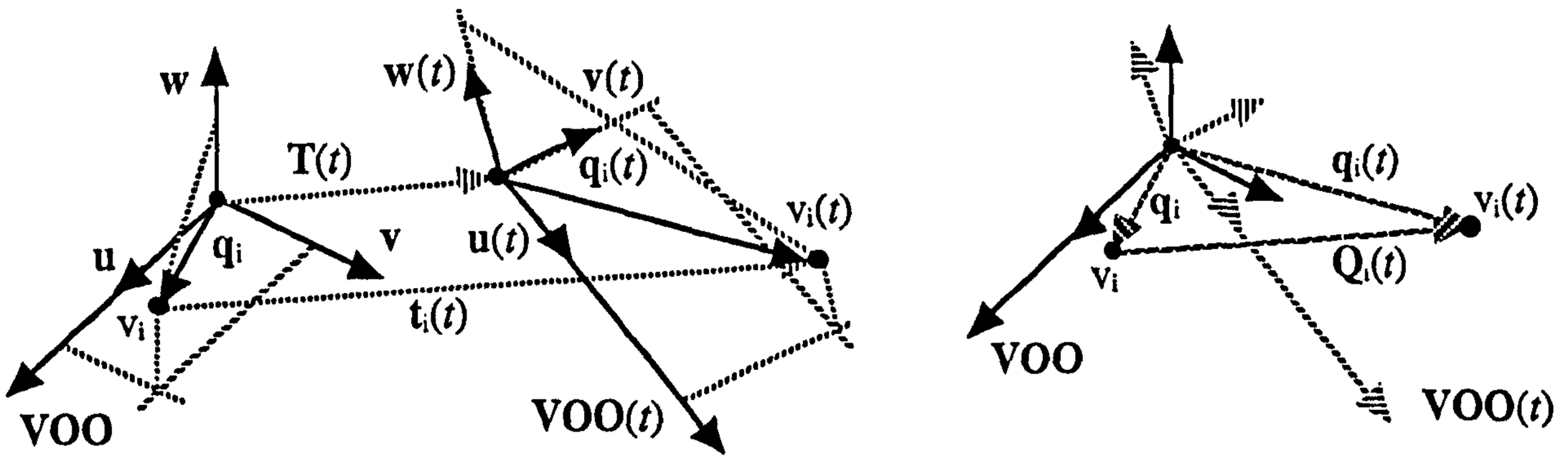
Figure 5.14 a.Vertex v_i defined using the local frame of a VOO at rest state.
b. The same vertex $v_i(t)$ at a time *t*.

In figure 5.14b, vertex $v_i(t) = (v_{lxi}(t), v_{lyi}(t), v_{lzi}(t))$ has been subjected to global translation and local scaling and rotation about the axes **u**(*t*), **v**(*t*) and **w**(*t*). The local position vector $\mathbf{q}_i(t)$ may be calculated as follows:

$$\mathbf{q}_i(t) = v_{lxi}(t) \cdot \mathbf{u}(t) + v_{lyi}(t) \cdot \mathbf{v}(t) + v_{lzi}(t) \cdot \mathbf{w}(t) \qquad [5.16]$$

5.4.1 Strain Vector Revisited

In figure 5.15a (which is similar to figure 5.6 in section 5.2.3), vector $\mathbf{t}_i(t)$ represents the total transformation of vertex v_i at time *t*. Vector **T**(*t*) represents the global translation. The difference between vectors $\mathbf{q}_i(t)$ and $\mathbf{q}_i$ represents the local transformation of vertex v_i (see figure 5.15b).



a. **b.**

Figure 5.15 a. Vector $\mathbf{t}_i(t)$ represents the total transformation of vertex v_i and $\mathbf{T}(t)$ is the global translation vector. b. $\mathbf{Q}_i(t)$ is the strain vector of vertex v_i.

According to equation [5.6] (see section 5.2.3):

$$\mathbf{t}_i(t) = \mathbf{T}(t) + \mathbf{Q}_i(t) \qquad [5.17]$$

where

$$\mathbf{Q}_i(t) = \mathbf{q}_i(t) - \mathbf{q}_i \qquad [5.18]$$

In equation [5.18], vector $\mathbf{Q}_i(t)$ contains both local scaling and rotation transformations and it is called the strain vector of vertex v_i (see figure 5.15b).

5.4.2 Strain Damping

When an external stress is being applied to a soft object the result is usually an amount of strain. This strain is centred at the point of stress application and it tends to spread out to neighbouring parts of the object. However, in soft materials, molecular connections being weak, they allow for local deformations which absorb deformation energy. The outcome of this is a spatial damping of the originally incurred strain.

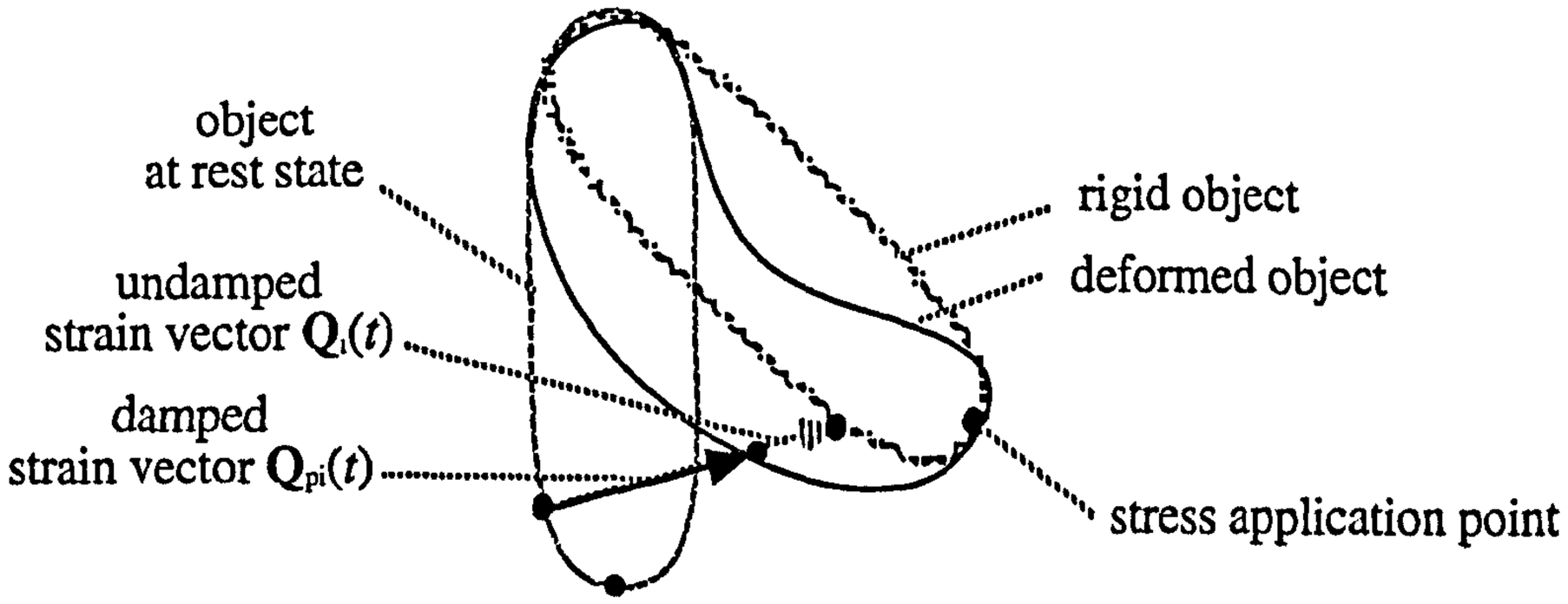


Figure 5.16 Spatial strain damping in a deformed object.

The amount of damping is related to the distance from the point of stress application (i.e. point of maximum strain) and it affects the magnitude of the strain vector (see figure 5.16). The greater the distance the greater the damping, so the strain eventually diminishes at some maximum distance away from the centre of strain application.

A spatial gradient may be applied to vector $\mathbf{Q}_i(t)$ (of equation [5.19]) in order to diminish its effect as distance increases away from a centre of maximum strain. This gradient may be represented by a function of distance (see figure 5.17).

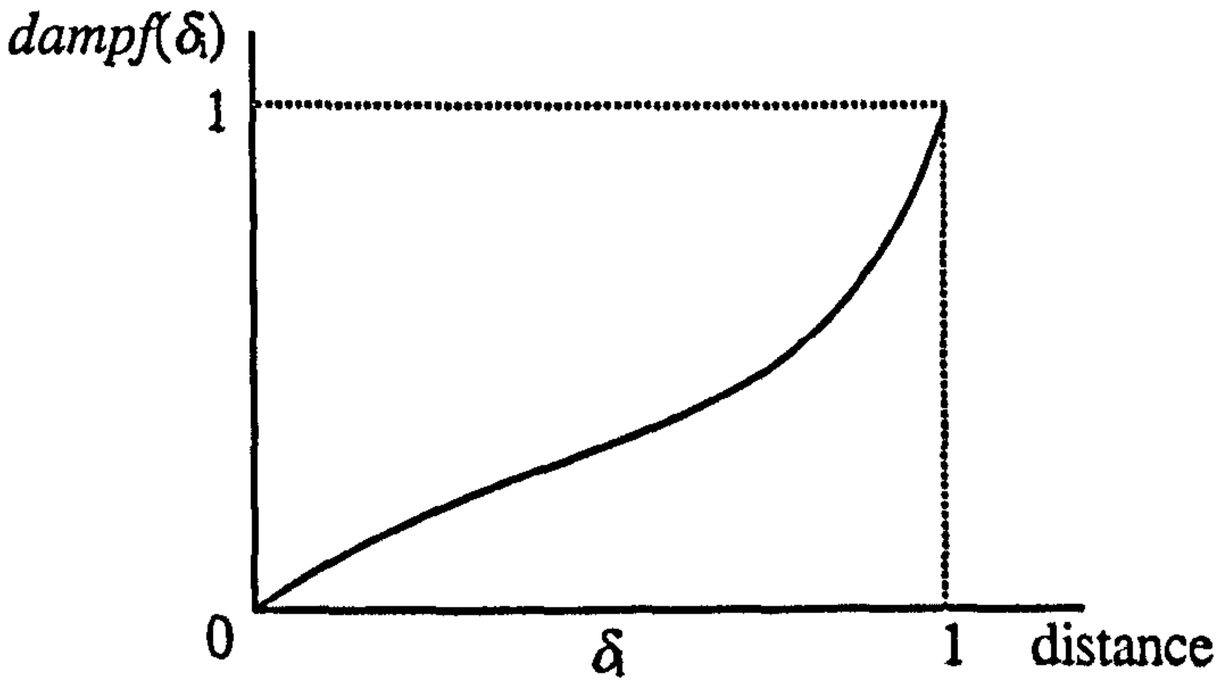


Figure 5.17 Spatial strain damping function $dampf(\delta_i)$.

The graph in figure 5.17 represents an arbitrary strain damping function. The value of 1 on the normalised distance axis represents the maximum distance at which the strain appears completely diminished. The value of 1 on the damping axis represents maximum damping of strain. Assuming, for the sake of simplicity, that the centre of maximum strain application is

located at the tip of a VOO, the absolute distance, d_i, between vertex v_i and the tip will determine the amount of damping that will be imposed on vector $\mathbf{Q}_i(t)$ (see figure 5.18).

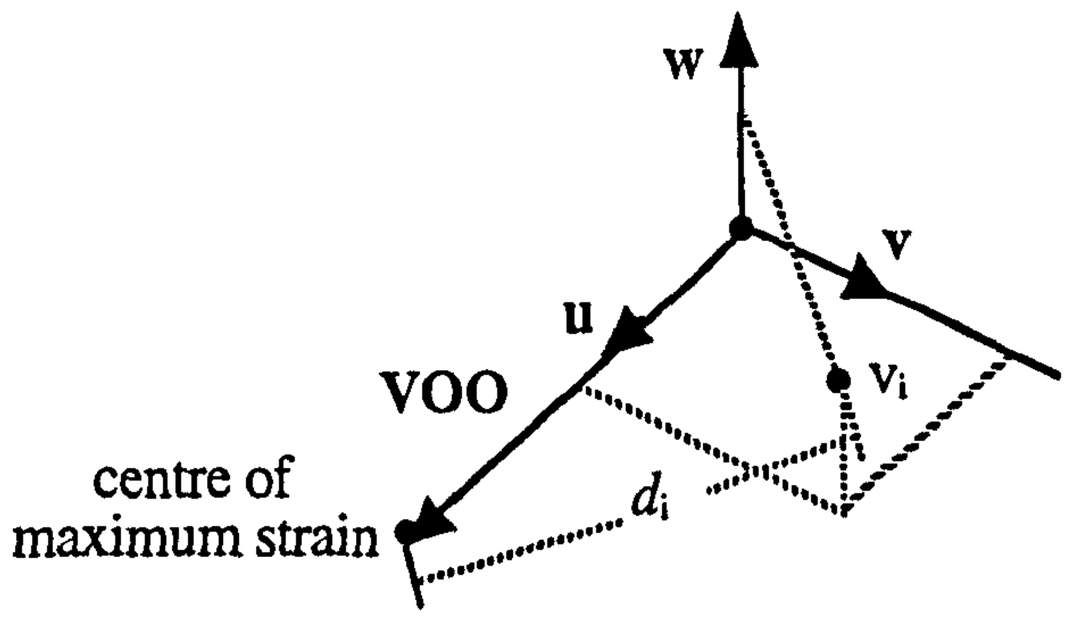


Figure 5.18 d_i is the distance between vertex v_i and the centre of maximum strain.

$$\delta_i = \frac{d_i}{d_{maxi}} \qquad [5.19]$$

In equation [5.19], d_i is the absolute distance of vertex v_i, d_{maxi} is the maximum distance at which the strain is completely diminished and δ_i is the normalised distance of the vertex. The maximum distance is determined by the shape and the size of the field of influence of a VOO (see section 5.5).

$$\mathbf{Q}_{pi}(t) = \mathbf{Q}_i(t)(1 - dampf(\delta_i)) \qquad [5.20]$$

In equation [5.20], $\mathbf{Q}_{pi}(t)$ is the damped strain vector of vertex v_i (see also figure 5.16). The subscript $_p$ stands for plastic strain because $\mathbf{Q}_{pi}(t)$ will be used for the implementation of plastic deformation. The following special cases may be derived from equation [5.20] and the graph in figure 5.17:

$\delta_i = 0$	Vertex v_i lies on the strain centre and it receives maximum strain.
$0 < \delta_i < 1$	Vertex v_i receives a damped strain $\mathbf{Q}_{pi}(t)$.
$\delta_i \geq 1$	Vertex v_i lies beyond the maximum distance and it receives no strain.

Figure 5.19 features several damping graphs which describe the deforming behaviour of a VOO. In figure 5.19a, the graph is a line parallel to the distance axis and at distance equal to 1 from it. This produces maximum damping of strain for the entire object. The object does not respond to any movement of the VOO and it behaves as if "frozen". In figure 5.19b, the graph coincides with the distance axis which causes no damping of strain, hence, no deformation. The object follows any movement of the VOO as if it were rigid. In figure 5.19c, a linear damping graph causes an even distribution of strain and the object appears to bend. Finally, in figure 5.19d an exponential graph produces a more complex spatial distribution of strain.

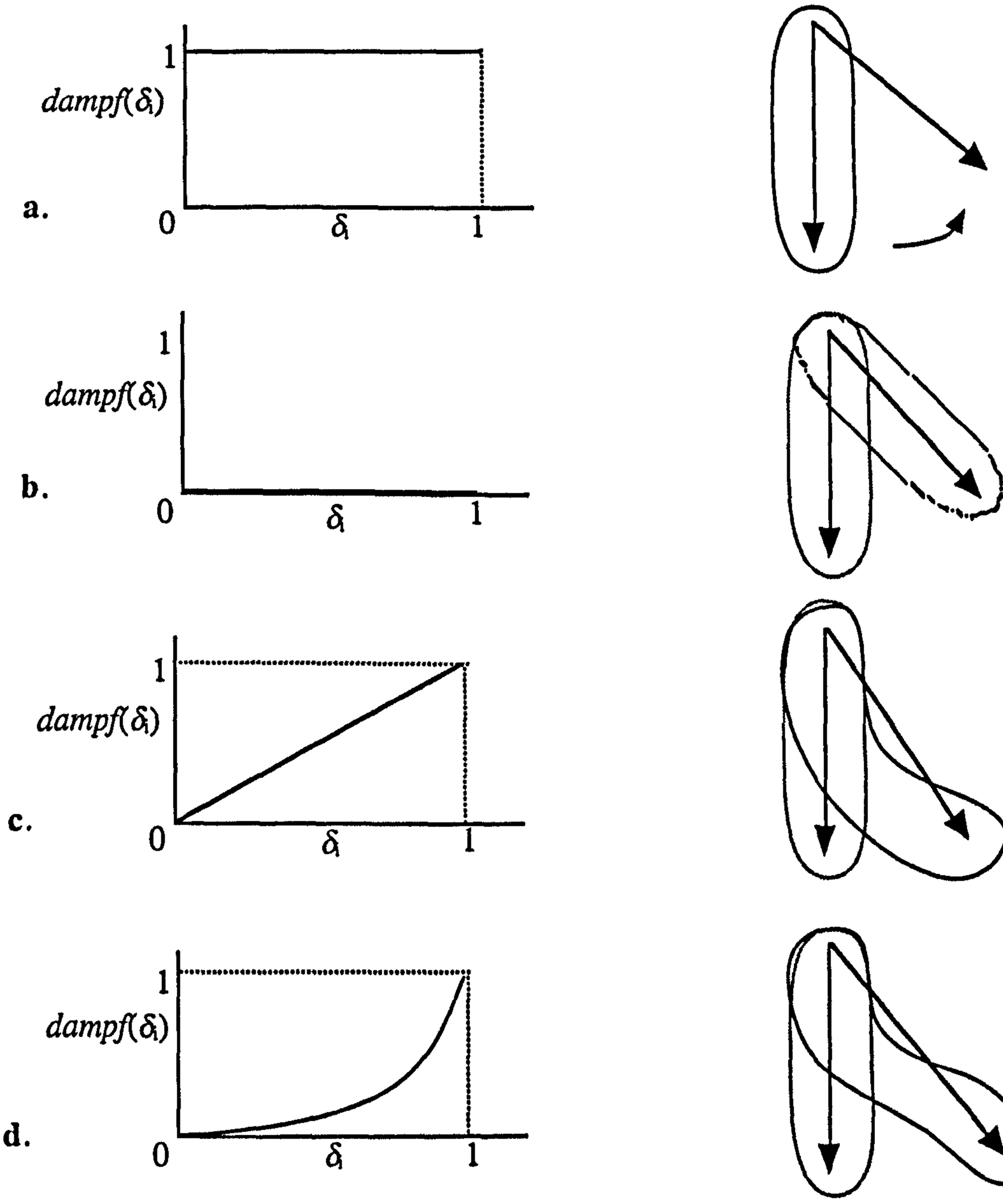


Figure 5.19 Various damping graphs and their influence in the shape of deformation.

Manipulation of a damping graph enables visual experimentation and fine-tuning of the deforming behaviour of a VOO in its local space. The user may design complicated damping graphs suitable for special purpose deformations or use a simple general purpose predefined graph, such as the ones in figure 5.19 (see also colour plates 5.49 and 5.50 in section 5.11).

5.4.3 Strain Control Coefficients

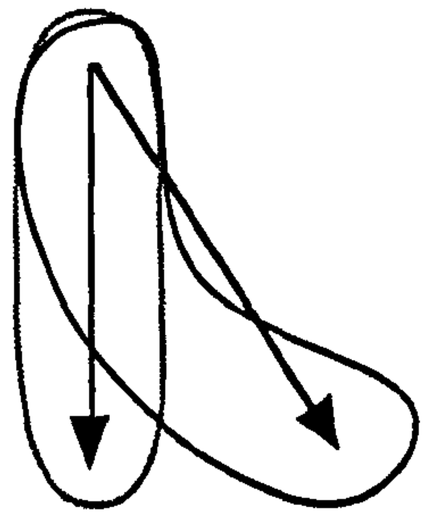
In the following paragraphs, several strain control coefficients are introduced which complement the effect of a damping function. The flexibility, locality and weighting coefficients are used to enhance the capability for fine-tuning a deformable behaviour.

5.4.3.1 The Flexibility Coefficient

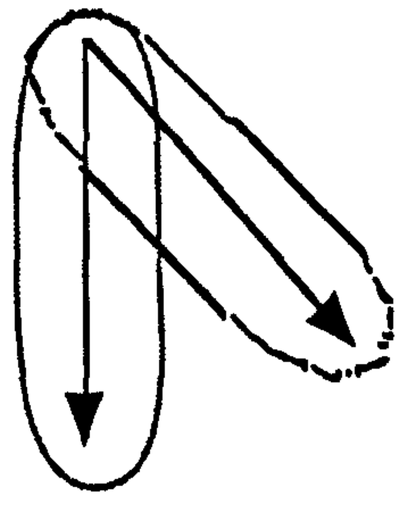
An object may be made to respond rigidly or flexibly using an appropriate damping function (see figure 5.19b, c and d). However, it may sometimes be awkward or inefficient to swap or interpolate between various damping functions. For this purpose, the flexibility coefficient, γ, is introduced. Equation [5.20] may be extended by incorporating the flexibility coefficient whose role is to scale the effect of the damping function:

$$\mathbf{Q}_{pi}(t) = \mathbf{Q}_i(t)(1 - \gamma \cdot dampf(\delta_i)) \qquad [5.21]$$

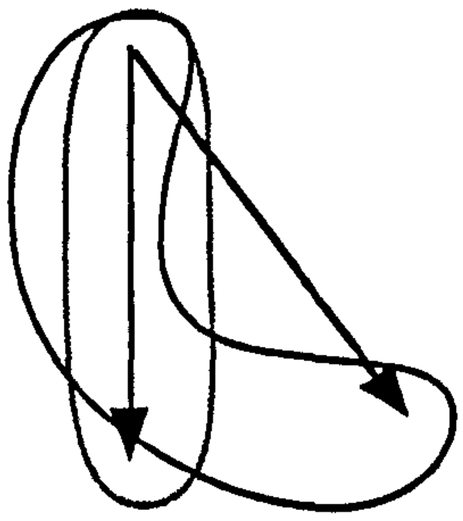
In figure 5.20, the same linear damping function (from figure 5.19c) is combined with different values of flexibility, γ, to achieve a variety of deformation profiles.



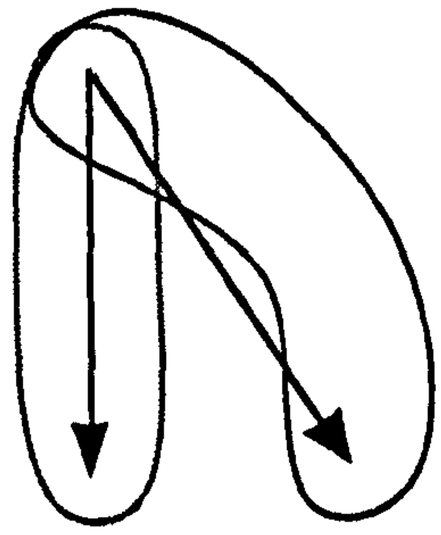
a. $\gamma = 1$ Regular damping.



b. $\gamma = 0$ No damping.



c. $\gamma > 1$ Exagerated damping.



d. $\gamma < 0$ Reversed damping.

Figure 5.20 The effect of flexibility coefficient, γ, on the shape of deformation.

The flexibility coefficient, γ, characterises the flexibility or the rigidity of local space. Under the influence of the same damping function a gradual increase of the value of γ, from 0 to 1, changes the behaviour of an object from perfectly rigid to perfectly plastic (or more precisely as described by the dumping function) (see also colour plates 5.45, 5.46, 5.49 and 5.50 in section 5.11).

5.4.3.2 The Locality Coefficient

The role of the locality coefficient is to localise the effect of a damping function. The locality coefficient is added to the normalised distance, δ_i, of each vertex which is used as a parameter of the damping function. Equation [5.21] may be extended to incorporate the locality coefficient, λ, as follows:

$$\mathbf{Q}_{pi}(t) = \mathbf{Q}_i(t)(1 - \gamma \cdot dampf(\delta_i + \lambda)) \qquad [5.22]$$

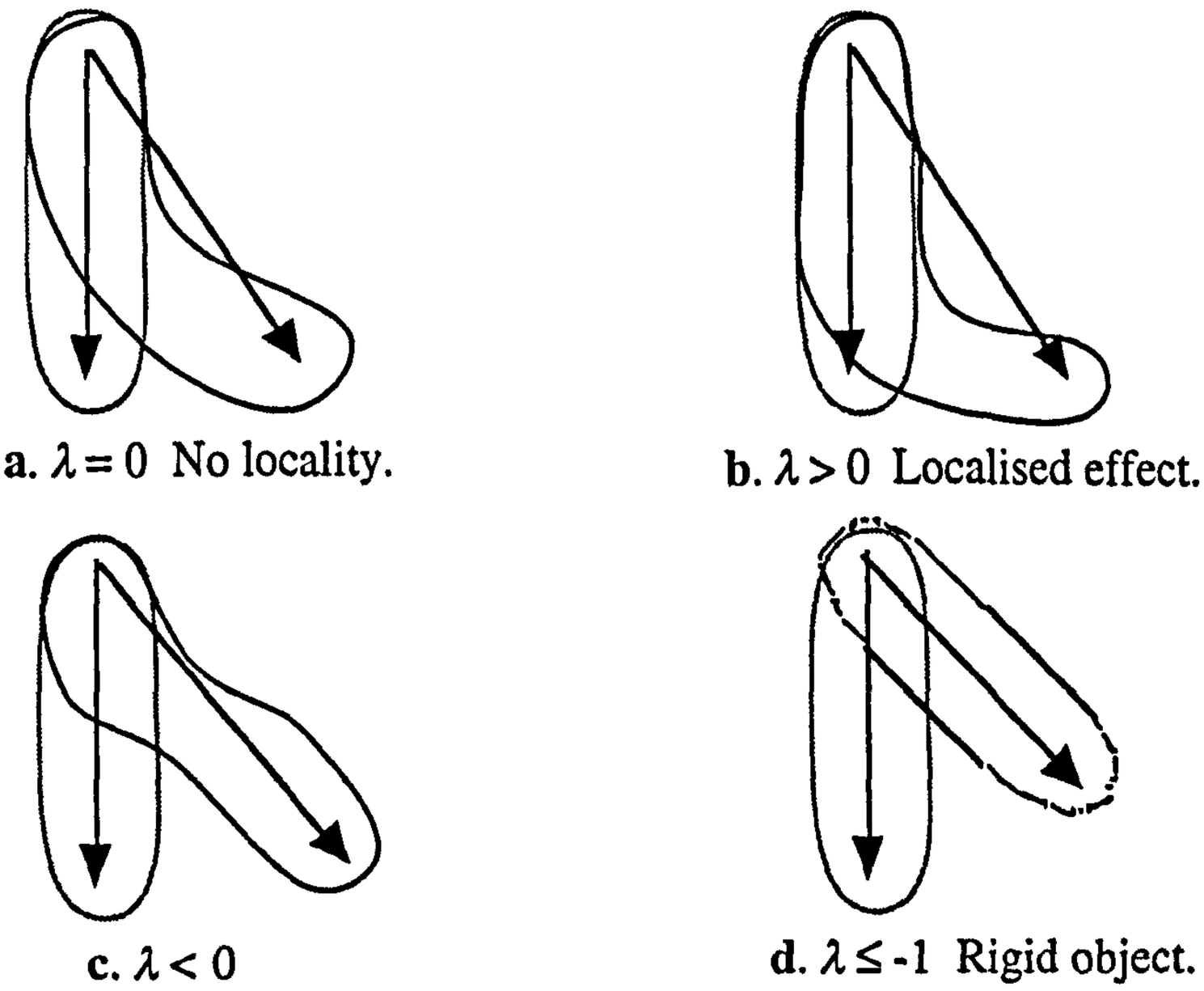


Figure 5.21 The effect of locality coefficient, λ, on the shape of deformation.

In figure 5.21b, a positive locality coefficient causes a more discontinuous damping of the strain and therefore a localisation of the deformation around the strain application centre (tip of VOO). In contrast, a negative value of λ causes the part of the object which is closer to the tip of the VOO to behave rigidly (no strain damping) as the rest of the object remains deformed (see figure 5.21c) (see also colour plate 5.51 in section 5.11).

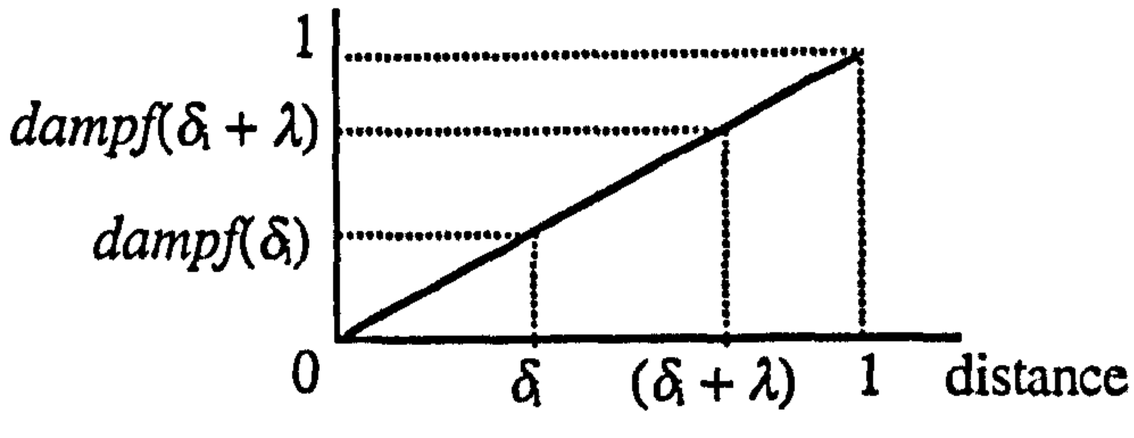


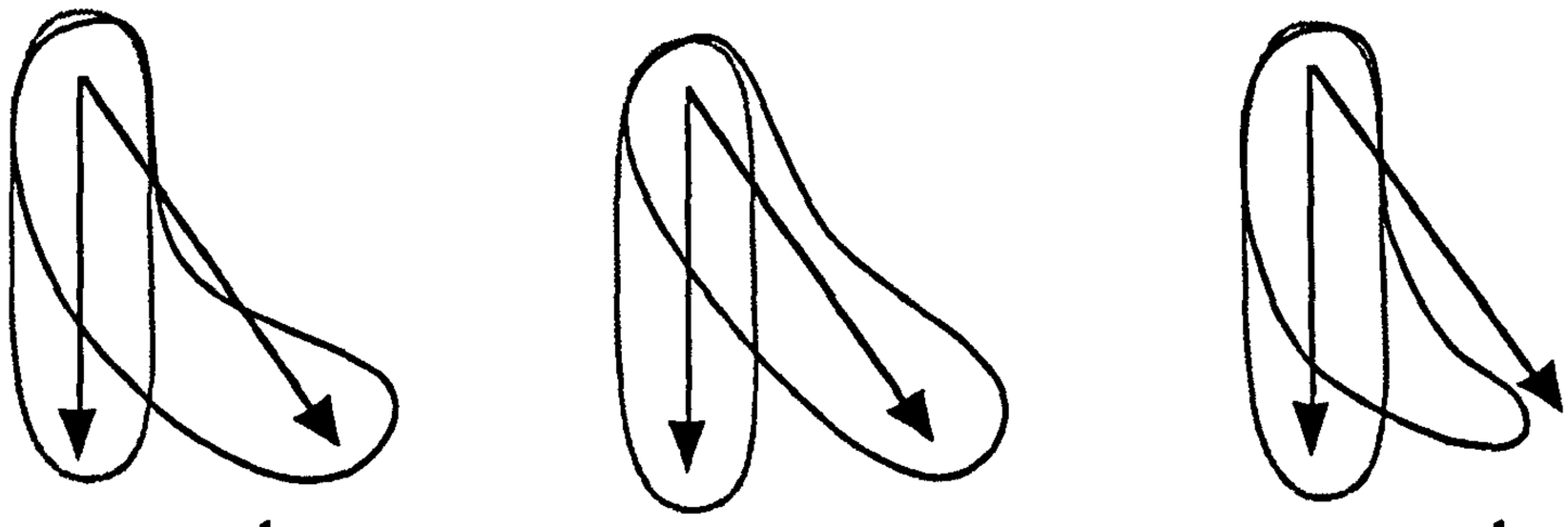
Figure 5.22 The effect of locality, λ, on the value of a damping function.

Figure 5.22 shows how a linear damping function is influenced by the existence of a locality coefficient.

5.4.3.3 The Weight Coefficient

The role of the weight coefficient is to emphasise or de-emphasise the overall deformation effect. For this purpose, the weight coefficient is used as an exponent for the value of the damping function (see equation [5.22] in previous paragraph). This equation may further be extended to incorporate the weight coefficient, ω, as follows:

$$\mathbf{Q}_{pi}(t) = \mathbf{Q}_i(t)\left(1 - \gamma \cdot dampf\left(\delta_i + \lambda\right)^{\omega}\right) \quad \text{where } \omega > 0 \qquad \textbf{[5.23]}$$



a. $\omega = 1$ **b.** $\omega > 1$ **c.** $\omega < 1$

Figure 5.23 The effect of weight coefficient, ω, on the shape of deformation.

In figure 5.23b, the deformation effect is raised to a power greater than 1. The opposite effect is achieved by raising to a power less than 1 (see figure 5.23c).

The three strain control coefficients of flexibility, γ, locality, λ and weight, ω, have no physical counterpart. However, they are particularly useful in the animation of deformable objects. They can be easily related to time and interactive control of their values provides a powerful and versatile deformation technique (see also colour plate 5.52 in section 5.11).

5.4.4 Bending and Twisting

Bending and twisting deformations may be achieved by rotation about a centre combined with a

gradient of increasing rotation angles.

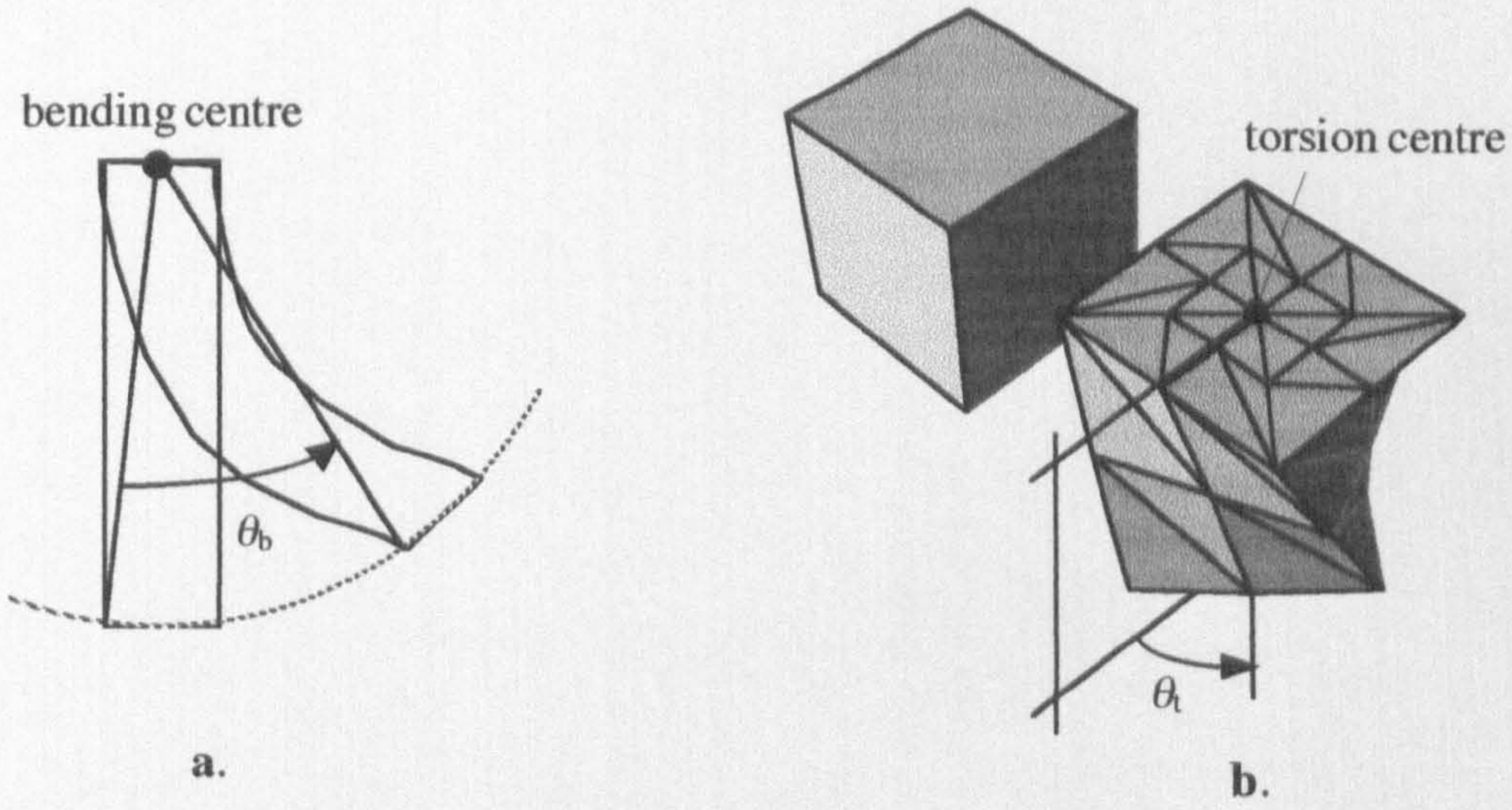


Figure 5.24 a. Bending, b. twisting.

The rotation angle is increased from 0 up to a specified maximum angle, θ, as distance increases away from this centre. Angle θ represents the bending strain or the twisting strain (also known as torsion) (see figure 5.24a and b).

5.4.4.1 Bending Using a VOO

A VOO may cause bending in local space, if rotated about the **v** or **w** axis or both.

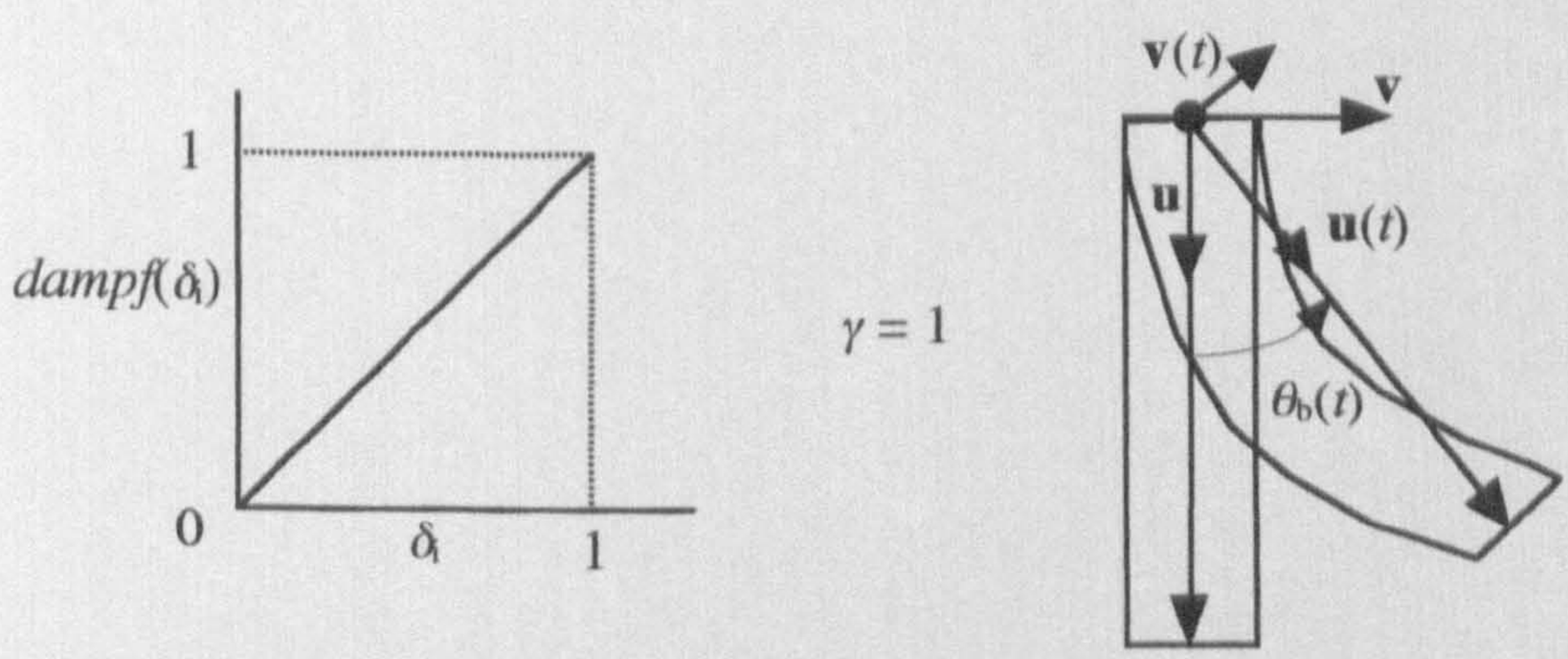


Figure 5.25: Bending by offsetting the tip of a VOO.

In fact, any offsetting of the tip in relation to the head of a VOO causes rotation of the VOO itself and, hence, bending strain. Therefore, there is no need for direct specification of the bending angle θ_b. In figure 5.25, the tip of a VOO is offset and as a result the local frame is rotated by angle θ_b, about the w axis (coming out of the page). A linear damping function and a damping coefficient of 1 enforce an even distribution of bending strain along the object. The bending centre is located at the head of the VOO and the angle of bending, $\theta_b(t)$, is equal to the angle between the direction of the VOO at rest state and the direction of the deformed VOO at time *t*. (see figure 5.25) (see also colour plates 5.46, 5.49 and 5.50 in section 5.11).

5.4.4.2 Twisting Using a VOO

A VOO may cause torsion strain in local space if rotated about the **u** axis. As mentioned above (see section 5.3.2) a VOO may be directly rotated about its major axis. One such rotation is depicted in figure 5.26. The torsion strain is increased from 0 up to θ_t as the distance increases away from the torsion centre.

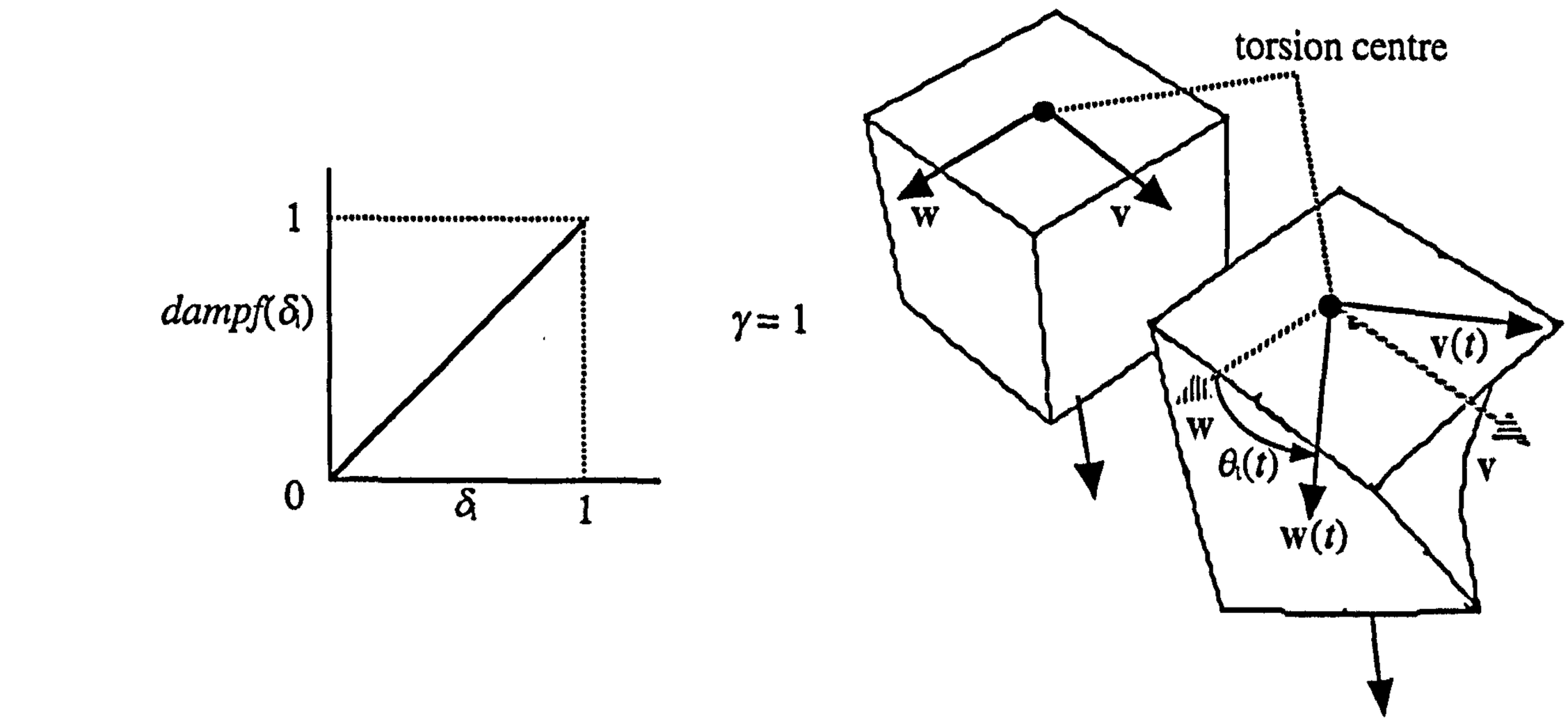


Figure 5.26 Twisting by rotating a VOO about the **u** axis.

As in bending, a linear damping function and a damping coefficient of 1 enforce an even

distribution of torsion strain along the object. The torsion centre is also located at the head of the VOO and the angle of torsion, $\theta_t(t)$, is equal to the angle between the direction of w axis at rest state and the direction of the deformed w(t) axis at time t (see figure 5.26). Angle $\theta_t(t)$ may be directly specified by the user (see also colour plate 5.47 in section 5.11).

Both bending and torsion strains are applied to an object by directly manipulating a VOO and its local frame (i.e. offsetting the tip and rotating the local frame about the **u** axis). Therefore, there is no need to apply rotation matrix transformations to the local coordinates of a vertex $v_i(t)$.

5.4.5 Stretching and Squashing

Stretching and squashing of objects are perhaps the most common types of deformation used in animation. They are usually associated with the phenomenon of collision between soft and rigid objects or wherever there is need to visualise action and reaction forces. Stretching of an object results in elongation strain and squashing to contraction strain.

5.4.5.1 Stretching and Squashing Using a VOO

In the proposed model, elongation and contraction are related to the change of length of a VOO. Lengthening of a VOO will normally result in elongation and shortening in contraction of an object. Elongation and contraction are defined as scaling operations along the major axis, **u**, of the local frame. The magnitude of a VOO at time t, may differ from the one at rest state. In such case, the nominal elongation strain of a VOO may be defined as follows (see equation [3.3] in section 3.2.1):

$$e(t) = \frac{(|\mathbf{V}(t)| - |\mathbf{V}|)}{|\mathbf{V}|} \qquad [5.24]$$

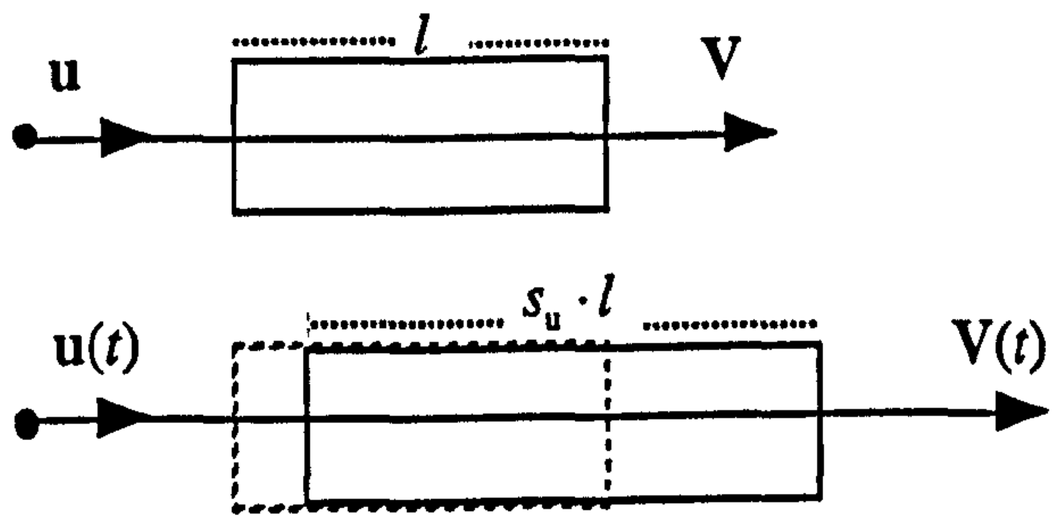


Figure 5.27 Stretching a VOO causes scaling along the **u** axis.

Elongation generates a positive nominal strain: $e(t) > 0$ and contraction generates a negative nominal strain: $e(t) < 0$ (see equation [5.24]). So, both elongation and contraction may be treated as one strain. Scaling along the **u** axis of a local frame may be achieved using the following scaling factor (see figure 5.27):

$$s_u(t) = \frac{|\mathbf{V}(t)|}{|\mathbf{V}|} = \frac{|\mathbf{V}|}{|\mathbf{V}|} + \frac{(|\mathbf{V}(t)| - |\mathbf{V}|)}{|\mathbf{V}|} = 1 + e(t) \qquad [5.25]$$

The scaling factor of equation [5.25] will scale the local space that surrounds a VOO proportionately to the change of length of the VOO (see figure 5.27) (see also colour plate 5.48 in section 5.11).

5.4.5.2 The Elongation Coefficient

In an interactive deformation model, the user may wish to have some direct control over the stretchability or contractility of objects. Although, a VOO may be freely manipulated to any length, the surrounding objects may be instructed to respond in a different way. For this purpose, equation [5.25] is amended as follows:

$$s_u(t) = 1 + \varepsilon e(t) \qquad [5.26]$$

where

$$\varepsilon \neq -\frac{1}{e(t)} \quad \text{(in order to avoid scaling by 0).}$$

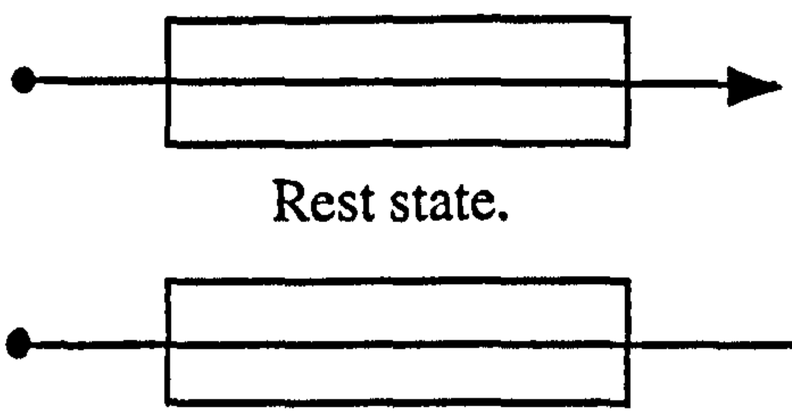


Figure 5.28 Various effects of the elongation coefficient.

The multiplying factor, ε, in equation [5.26], dictates the percentage of the nominal elongation strain of the VOO that will be transferred to local space. ε is the elongation coefficient and it characterises the behaviour of local space that surrounds a VOO. Equation [5.26] yields the special cases illustrated in figure 5.28.

5.4.6 Volume Preservation

Stretching a real object in a given direction is often accompanied by a thinning of the cross-sectional area which lies on the plane perpendicular to the direction of stretch (see figure 5.29).

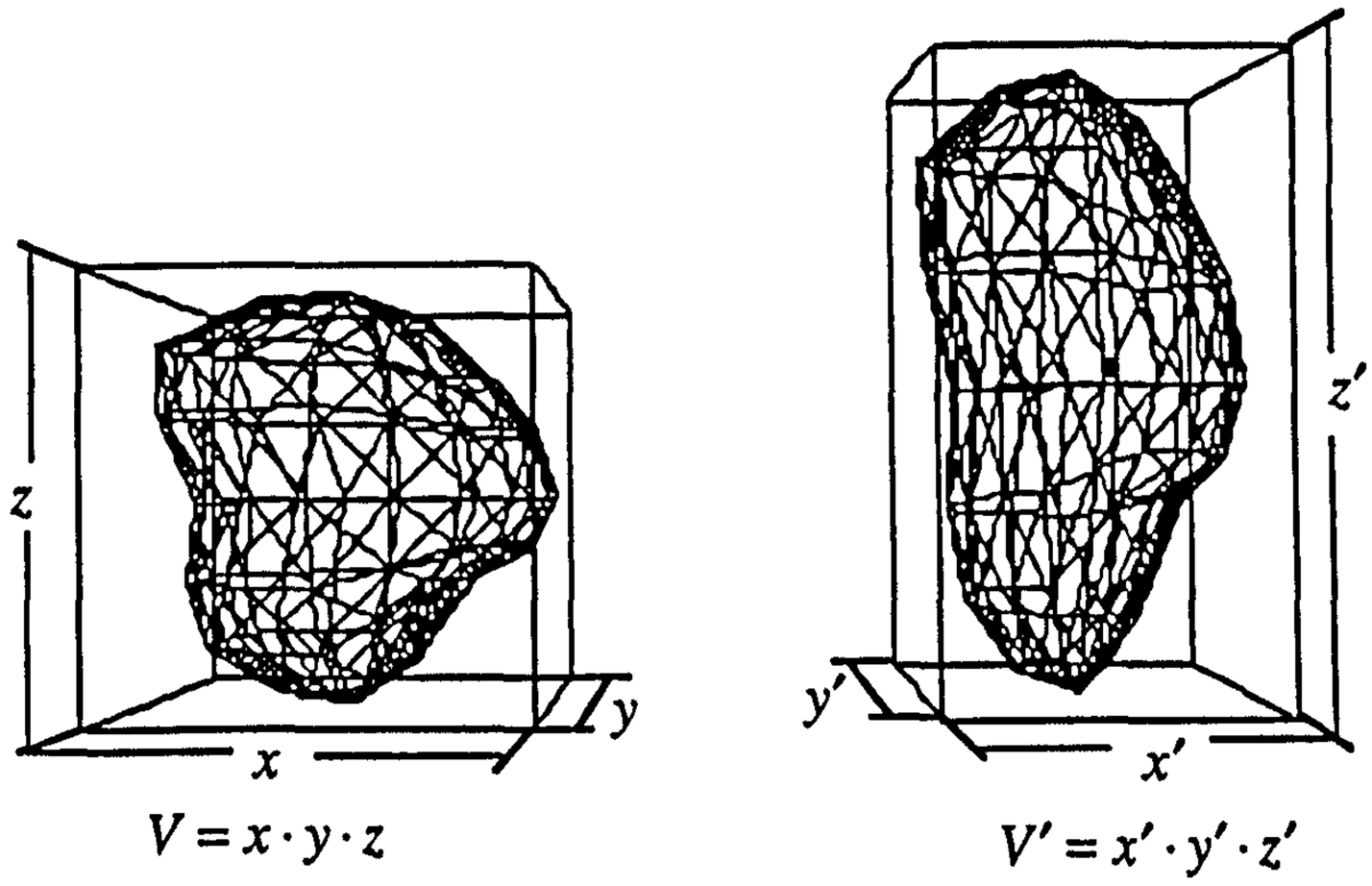


Figure 5.29 Volume preservation of bounding boxes.

In contrast, contraction is accompanied by expansion of the cross-sectional area. This is known as the property of volume preservation. An approximation of volume preservation of a complex 3D object may be achieved by keeping the volume of its bounding box constant. The bounding box of an object is the smallest orthogonal parallelepiped which completely encloses the object. In figure 5.29, the object on the right is a scaled version of the object on the left with the bounding box volume preserved. So, the two volumes are equal and therefore:

$$
\begin{aligned}
& V = V' \\
& \Rightarrow x \cdot y \cdot z = x' \cdot y' \cdot z' \\
& \Rightarrow x \cdot y \cdot z = (s_x \cdot x)(s_y \cdot y)(s_z \cdot z) \\
& \Rightarrow s_x \cdot s_y \cdot s_z = 1
\end{aligned}
\qquad \textbf{[5.27]}
$$

where s_x, s_y and s_z are the scaling factors along x, y and z respectively.

5.4.6.1 Volume Preservation Using a VOO

In the proposed model, elongation or contraction is achieved by scaling along the **u** axis of the local frame by a factor s_u (see equation [5.26]). Therefore, any change of the cross-sectional area of an elongated or contracted object may be implemented as a uniform scaling along the axes **v** and **w** by the factors s_v and s_w respectively (see figure 5.30).

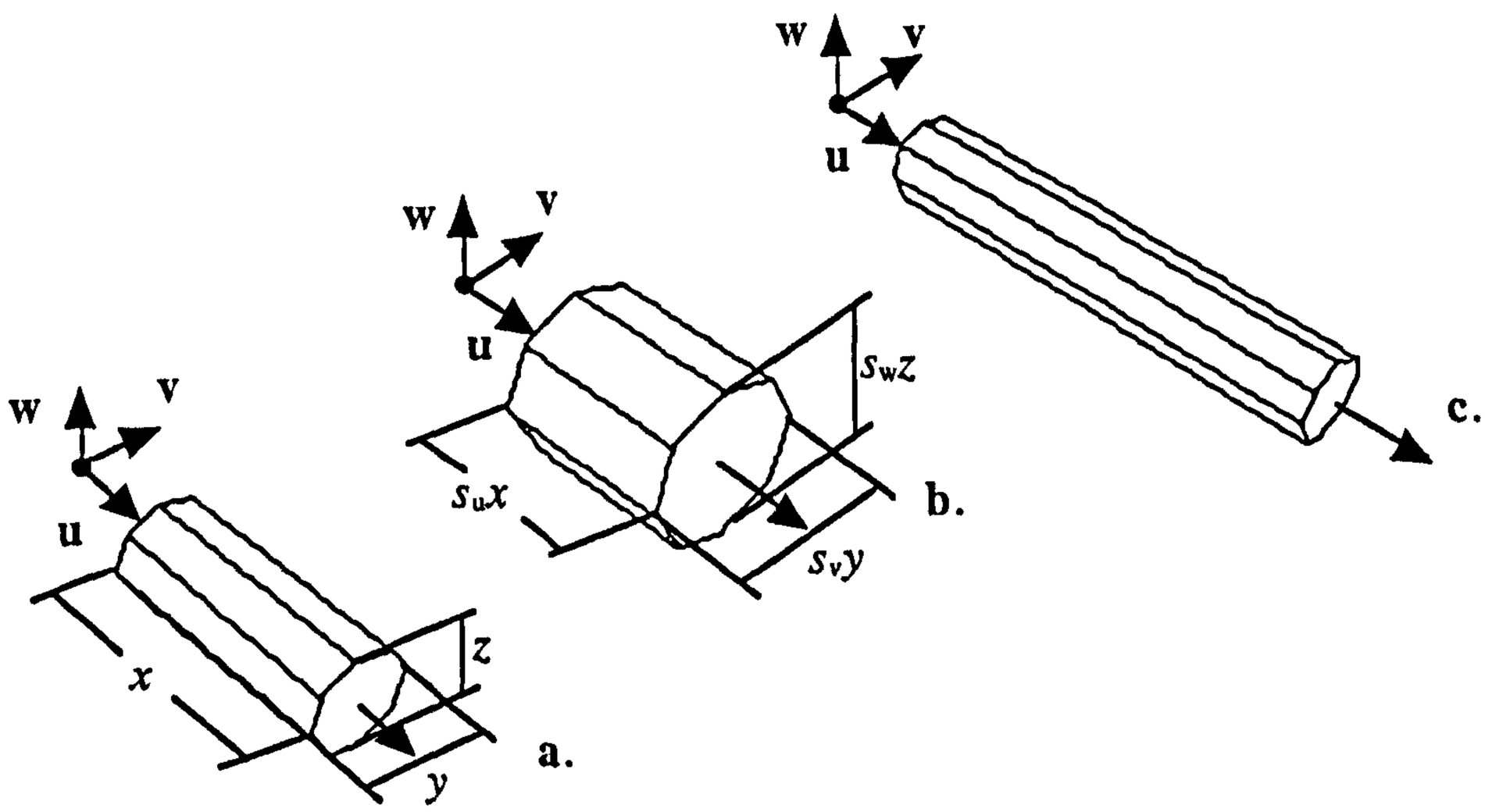


Figure 5.30 a. Rest state. b. Contraction in **u** with expansion in **v** and **w**.
c. Elongation in **u** with thinning in **v** and **w**.

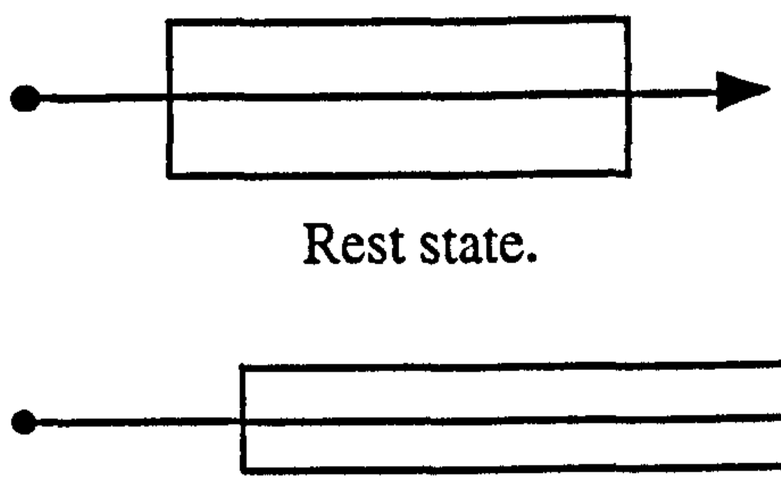
A uniform scaling along **v** and **w** means that $s_v = s_w$. According to equation [5.27] the scaling factors s_u, s_v and s_w must satisfy the following condition:

$$s_u \cdot s_v \cdot s_w = 1$$

$$\Rightarrow s_v = s_w = \frac{1}{\sqrt{s_u}} \qquad [5.28]$$

5.4.6.2 The Volume Preservation Coefficient

The incorporation of a volume preservation coefficient allows the user of the interactive model, to directly control the volume of the deformed object.



Rest state.

a. $a = 1$ Volume is preserved.

b. $0 < a < 1$ Volume is reduced.

c. $a > 1$ Volume is increased.

Figure 5.31 Various effects of the volume preservation coefficient.

Equation [5.28] may be enriched with a volume preservation coefficient, a, as follows:

$$s_v = s_w = \frac{a}{\sqrt{s_u}} \quad \text{where } a \neq 0 \qquad \textbf{[5.29]}$$

Equation [5.29] yields the special cases illustrated in figure 5.31.

5.4.6.3 The Cross-Sectional Coefficient

In equations [5.28] and [5.29], it is assumed that the cross-sectional area of an object is

subjected to uniform scaling $s_v = s_w$. This may not always be desirable. In order to facilitate non-uniform scaling along v and w a cross-sectional scaling factor, b, is introduced:

$$s_v = b\frac{a}{\sqrt{s_u}}, \quad s_w = \frac{1}{b}\frac{a}{\sqrt{s_u}} \quad \text{where } b \neq 0 \qquad \textbf{[5.30]}$$

The cross-sectional coefficient, b, preserves the volume by squashing the v dimension and stretching w or vice versa. The user may directly control the volume of the deformed object with greater accuracy using combined modification of the volume preservation coefficient, a, and the cross-sectional coefficient, b, (see also colour plate 5.48 in section 5.11).

5.4.6.4 Constructing the Scaling Matrix

The scaling factors $s_u(t)$, $s_v(t)$ and $s_w(t)$ are time dependent because they are derived from the time dependent elongation strain $e(t)$ (see equation [5.26]). Once all three of them have been determined, they are applied to the local coordinates of vertex v_i to produce the local coordinates of the vertex $v_i(t)$ at time t.

$$\mathbf{v}_i(t) = \begin{bmatrix} s_u(t) & 0 & 0 \\ 0 & s_v(t) & 0 \\ 0 & 0 & s_w(t) \end{bmatrix} \cdot \mathbf{v}_i \qquad \textbf{[5.31]}$$

The scaling matrix in equation [5.31] is the same as matrix **M** in equation [5.15] (see section 5.4). The damping function (see section 5.4.2) is used to damp the total strain vector, $\mathbf{Q}_i(t)$, of a vertex v_i. Consequently, bending, torsion and elongation strains are all damped in the same manner with increasing distance from the centre of strain application. Shearing strain is rarely encountered in 3D deformation and for the sake of brevity it is not implemented.

5.5 Fields of Influence

5.5.1 Introduction

In the previous sections, the concept of a VOO was introduced as a strain application tool in local space. The deformation character of a VOO is described by a set of parameters such as a damping graph and strain and volume control coefficients.

The influence of a VOO on its surrounding space is limited by a 3D field. This field of influence of a VOO contains a region of local space which inherits the deformation characteristics of the VOO. As a result, any object that enters the field of influence of a VOO is subjected to an amount of deformation. The space inside a field of influence experiences a gradient of deformation behaviour based on the shape of a damping graph (see section 5.4.2). The purpose of a field of influence is to localise the deformation effect of a VOO and so, to enable fine control upon complex 3D objects. The use of several VOOs on the same object or the use of one VOO over several objects can be facilitated by assigning an appropriate field of influence to each VOO. The basic shape of a field of influence is an ellipsoid aligned to a VOO. The size of this ellipsoid may be directly controlled by the user. Moreover, any modification of the magnitude or direction of a VOO is also passed on to its field of influence (see figure 5.32).

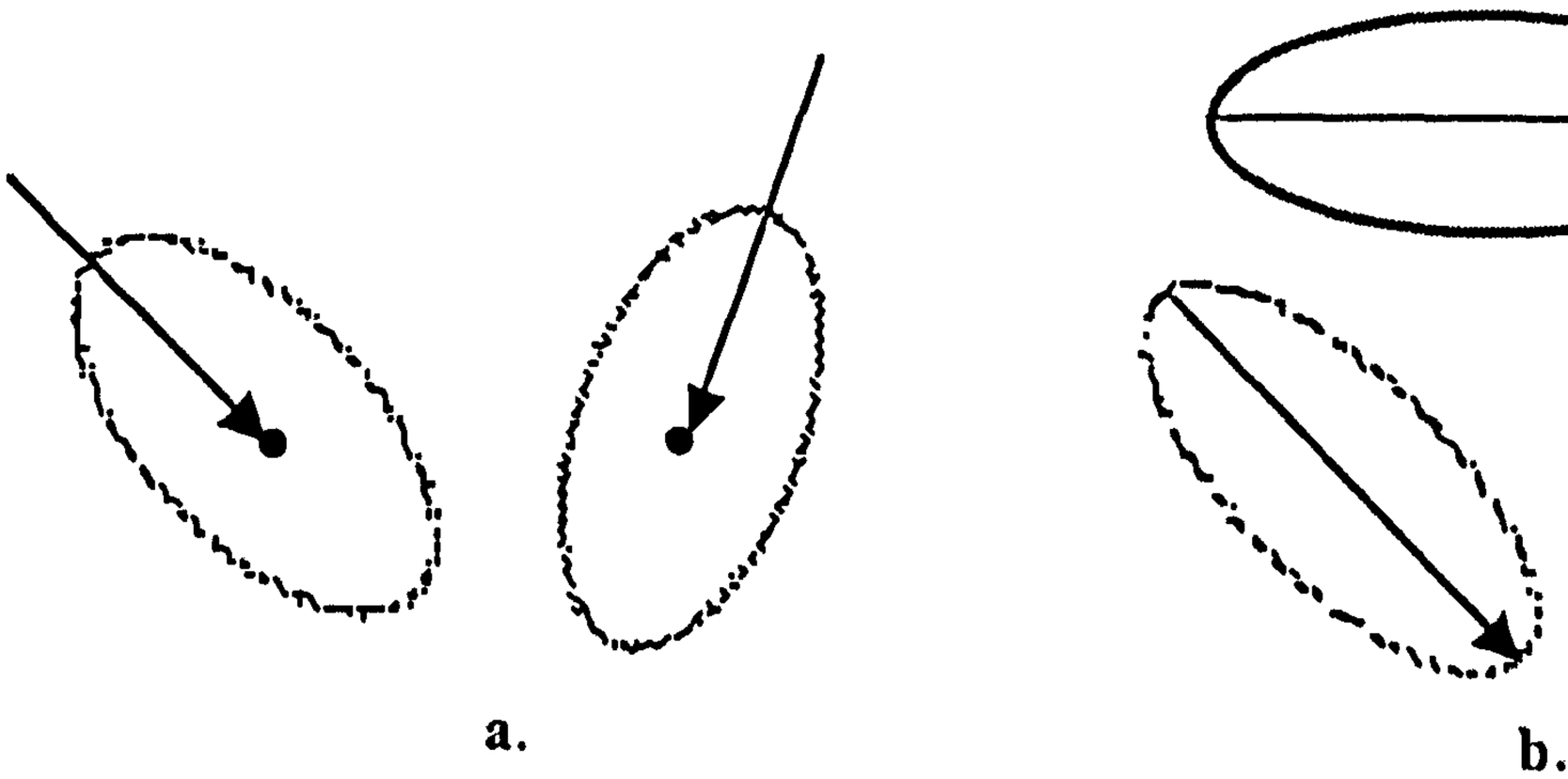


Figure 5.32 Two types of influence fields.

In the proposed model there are two distinct types of fields of influence and inherently two types of VOOs: the polar VOO (see figure 5.32a) and the length VOO (see figure 5.32b).

5.5.2 The Polar VOO and Field of Influence

In figure 5.32a, the centre of the ellipsoid is located at the tip of a VOO. This is a polar VOO and its tip acts as the centre of strain application. So far in this document, for simplicity, it has been assumed that this is always the case. In fact, a polar VOO may have up to two centres of strain application, one located at the head and one at the tip, each with its own independent field of influence (see figure 5.33).

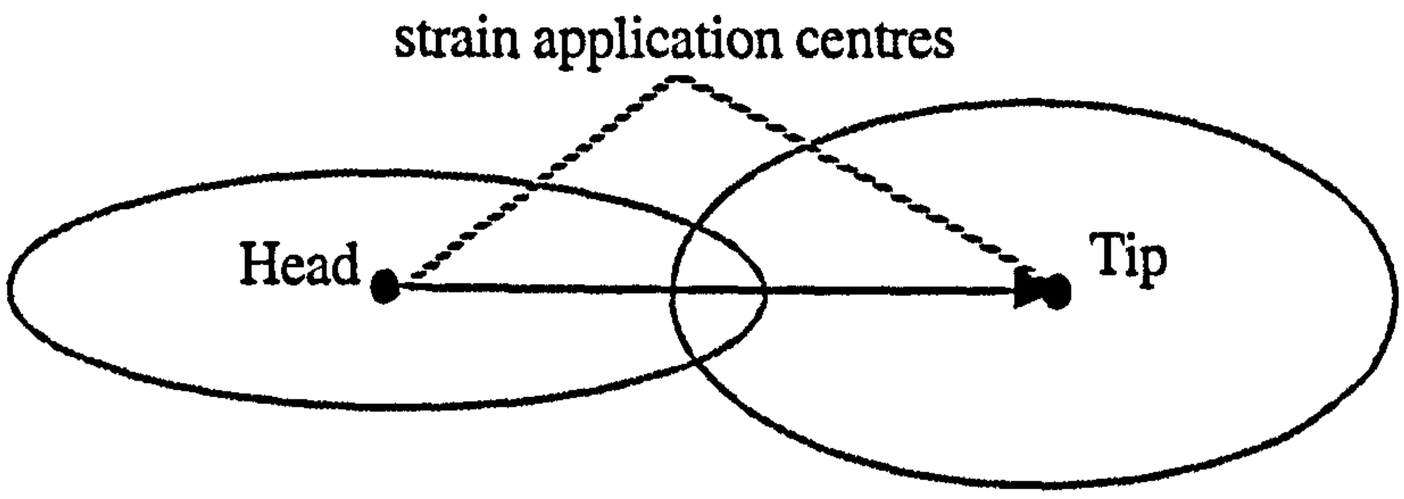


Figure 5.33 A polar VOO.

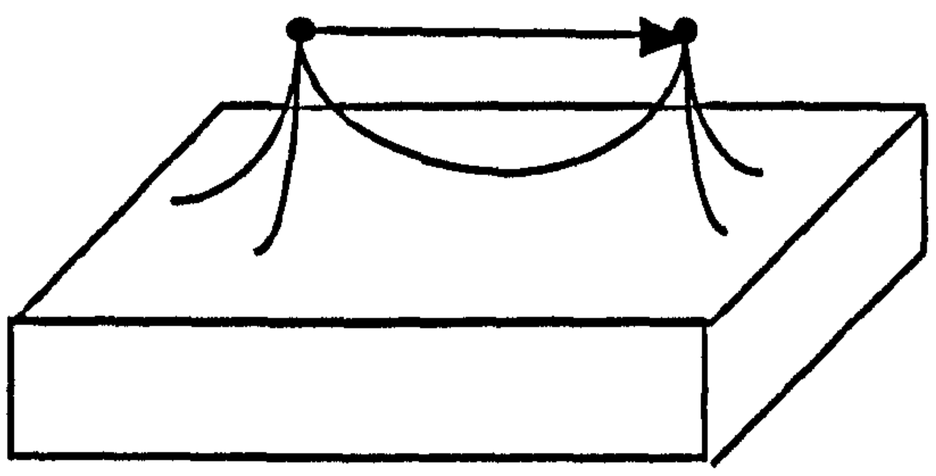


Figure 5.34 A polar VOO attached to a piece of mouldable material.

The deformation effect of a polar VOO resembles that of a rigid pin attached to a mouldable material at both its ends (see figure 5.34) (see also colour plate 5.53 in section 5.11).

5.5.2.1 The Ellipsoid of the Polar Field of Influence

The role of the field of influence of a VOO is to assist the calculation of the amount of deformation that will be received by each part of an object that enters the field. The damping graph of a VOO (see section 5.4.2) does exactly that using the normalised distance for each vertex, δ_i.

$$\delta_i = \frac{d_i}{d_{maxi}} \qquad [5.32]$$

In equation [5.32], d_i is the absolute distance of the vertex from the strain application centre and d_{maxi} is the distance from the centre to the surface of zero influence of the field. Figure 5.35a illustrates the simple case of a polar VOO with only one end active (the tip) and one vertex v_i inside the field. Initially the VOO, the field ellipsoid and the vertex are translated so the strain centre coincides with the origin O and are rotated so they become aligned to the global Z axis. This is done to simplify the calculation of d_i and d_{maxi} (see figure 5.35b).

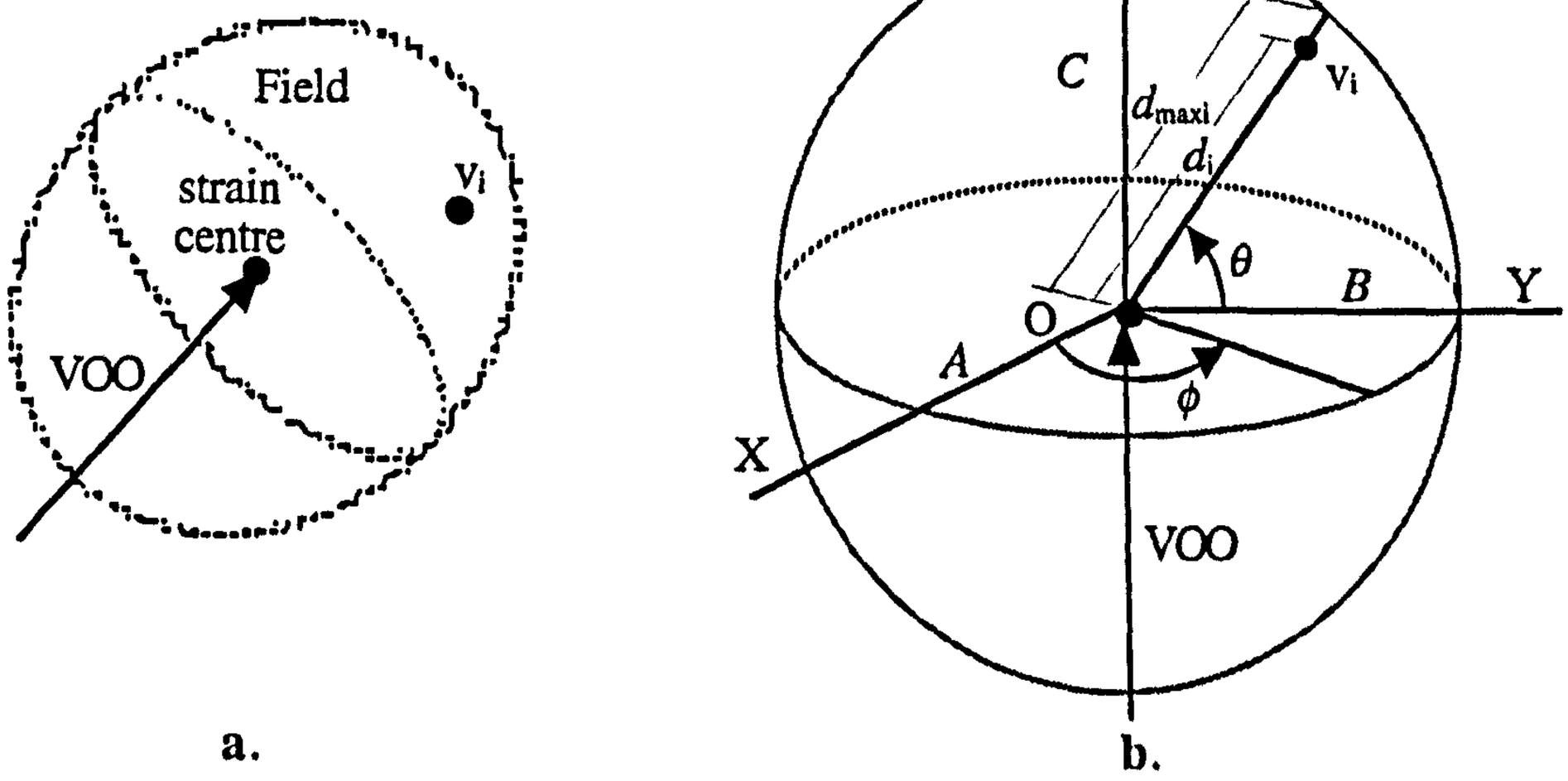


Figure 5.35 A vertex v_i inside the field of a VOO.

The distance d_i may be calculated as the distance of vertex v_i from the origin O:

$$d_i = \sqrt{v_{xi}^2 + v_{yi}^2 + v_{zi}^2} \qquad [5.33]$$

The distance d_{maxi} is equal to the length of the ellipsoid radius that passes through vertex v_i (see figure 5.35b). This radius may be calculated as follows:

$$d_{maxi} = \sqrt{(A\cos(\theta)\cos(\phi))^2 + (B\cos(\theta)\sin(\phi))^2 + (C\sin(\theta))^2} \qquad [5.34]$$

where C is the major radius of the ellipsoid and A and B the minor radii. A, B and C should not be equal to 0 at the same time. The cosines and the sines of angles θ and ϕ in equation [5.34] may be calculated as follows (see figure 5.35b):

$$\sin(\theta) = \frac{v_{zi}}{d_i} \qquad \cos(\theta) = \frac{\sqrt{v_{xi}^2 + v_{yi}^2}}{d_i}$$

$$\sin(\phi) = \frac{v_{yi}}{\sqrt{v_{xi}^2 + v_{yi}^2}} \qquad \cos(\phi) = \frac{v_{xi}}{\sqrt{v_{xi}^2 + v_{yi}^2}} \qquad [5.35]$$

In equations [5.35], if $d_i = 0$ the sine and cosine of angle θ cannot be defined. In this case, vertex v_i coincides with the centre of strain application. Therefore, the normalised distance δ_i is equal to 0 and there is no need to calculate d_{maxi}. If both v_{xi} and v_{yi} are equal to 0 then the sine and cosine of angle ϕ cannot be defined. In this case, vertex v_i lies on the Z axis and d_{maxi} is assumed to be equal to C. The coordinates of vertex v_i which are used for the calculation of d_i and d_{maxi} are the undeformed global coordinates of vertex v_i after they have been subjected to a translation and rotation (see ellipsoid alignment to Z axis above). The major and minor radii of the ellipsoid C, A and B, may be directly manipulated by the user. This is useful for localising the deformation effect of a VOO.

5.5.2.2 The Head and Tip Fields

As mentioned in section 5.5.2, a polar VOO may have two strain application centres, each with

a separate field of influence (see figure 5.33). In such case, the head and tip fields may partially overlap (see figure 5.36).

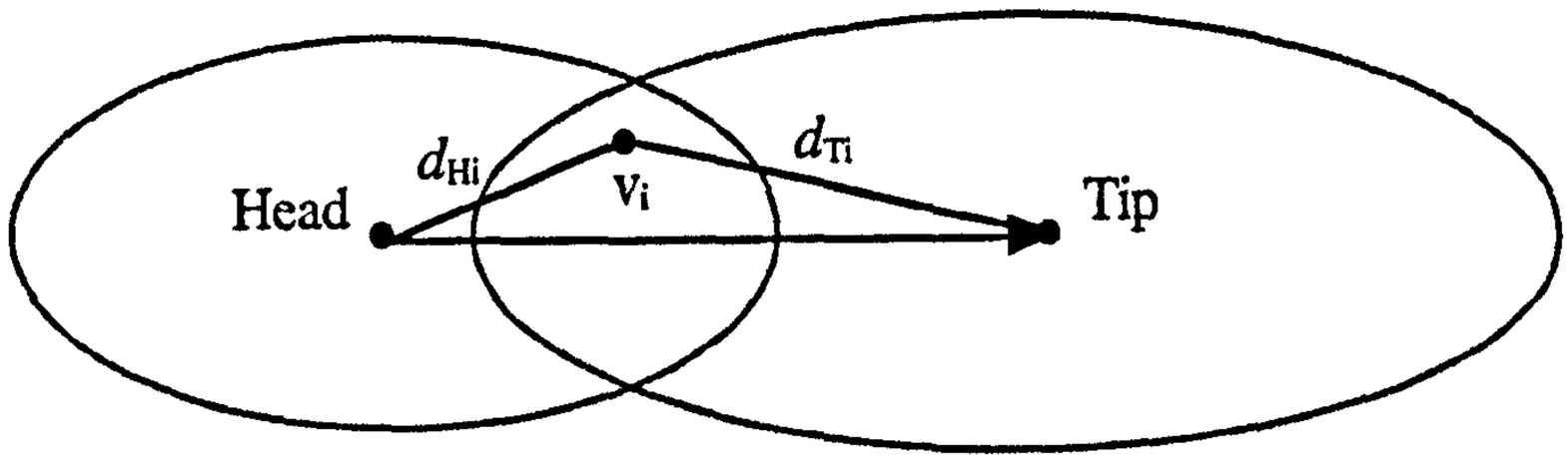


Figure 5.36 Polar vector with two partially overlapping fields.

Assuming that vertex v_i lies inside the common space between the two fields, the process of calculating the normalised distance δ_i for that vertex (see section 5.5.2.1) will be repeated twice. The head field will produce the normalised distance δ_{Hi}:

$$\delta_{Hi} = \frac{d_{Hi}}{d_{Hmaxi}} \qquad [5.36]$$

and the tip field will produce the normalised distance δ_{Ti}:

$$\delta_{Ti} = \frac{d_{Ti}}{d_{Tmaxi}} \qquad [5.37]$$

The two normalised distances are multiplied together to generate one combined normalised distance δ_{Ai}:

$$\delta_{Ai} = \delta_{Hi} \cdot \delta_{Ti} \qquad [5.38]$$

The combined normalised distance δ_{Ai} is then used as a parameter for the damping function $dampf(\delta_{Ai})$. It is assumed, though, that both head and tip fields share the same damping function. If that is not the case, then the two damping functions would have to be evaluated first and then averaging would have to be applied to the damping values.

5.5.2.3 The Ellipsoid Cap

In a polar field, it is sometimes desirable to reduce the volume of the part of the field that does not contain the VOO (stippled area in figure 5.37). This effect is achieved by splitting the ellipsoid in two halves, one that contains the VOO and another that does not. The latter may be scaled by reducing its major radius C' and by keeping the other two radii A' and B', the same as in the other half ellipsoid.

$$C' = \kappa C \quad \text{where } \kappa > 0 \qquad [5.39]$$
$$A' = A \quad B' = B$$

In equation [5.39], κ is the cap coefficient and its purpose is to scale the "cap" of the field of influence. The calculations required for d_i and d_{maxi} for vertices inside the scaled cap of the field are identical to the ones introduced in section 5.5.2.1.

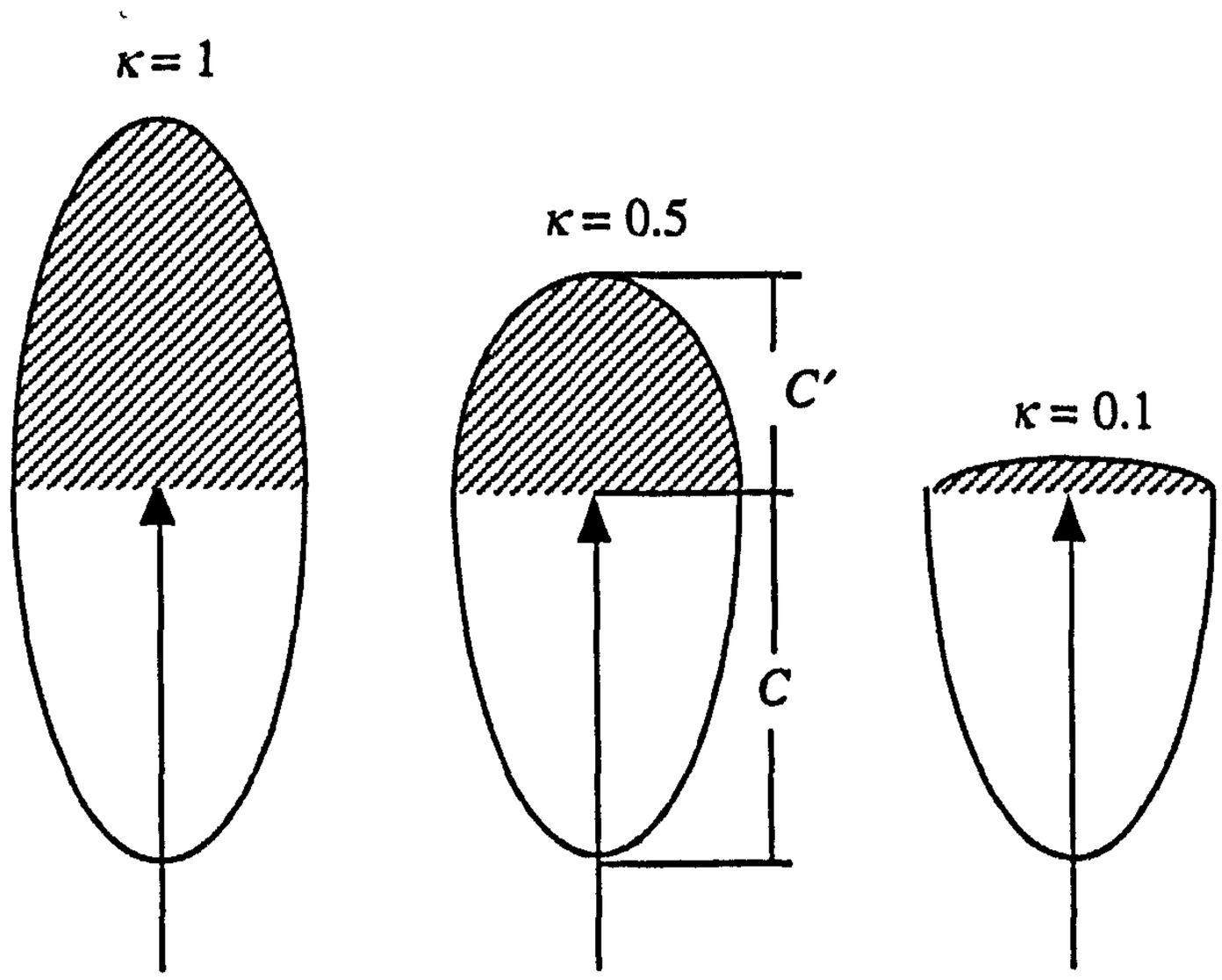


Figure 5.37 The cap coefficient, κ, scales half of the field ellipsoid.

The importance of the cap coefficient, κ, will become more apparent in chapter 8, where methods for connecting VOOs are discussed.

5.5.3 The Length VOO and the Field of Influence

In a length VOO the strain application centre is spread along the entire length of the vector.

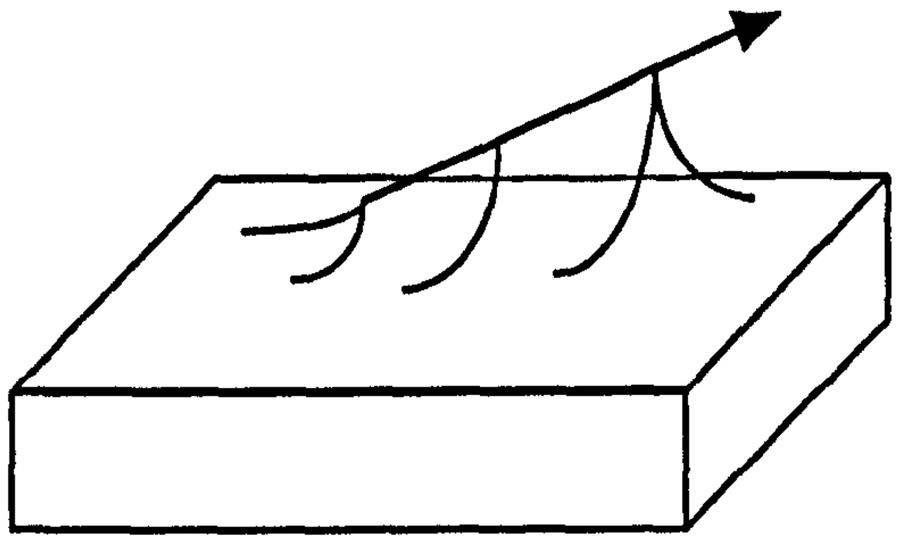


Figure 5.38 A length VOO pin immersed in a piece of mouldable material.

The deformation effect of a length VOO resembles that of a rigid pin immersed in a piece of mouldable material (see figure 5.38) (see also colour plate 5.54 in section 5.11). The field of influence of a length VOO is an ellipsoid centred at the middle of the VOO (see figure 5.39a). The VOO, its field of influence and vertex v_i are initially translated so the centre of the VOO coincides with the global origin O and then rotated to become parallel to the Z axis (see figure 5.39b).

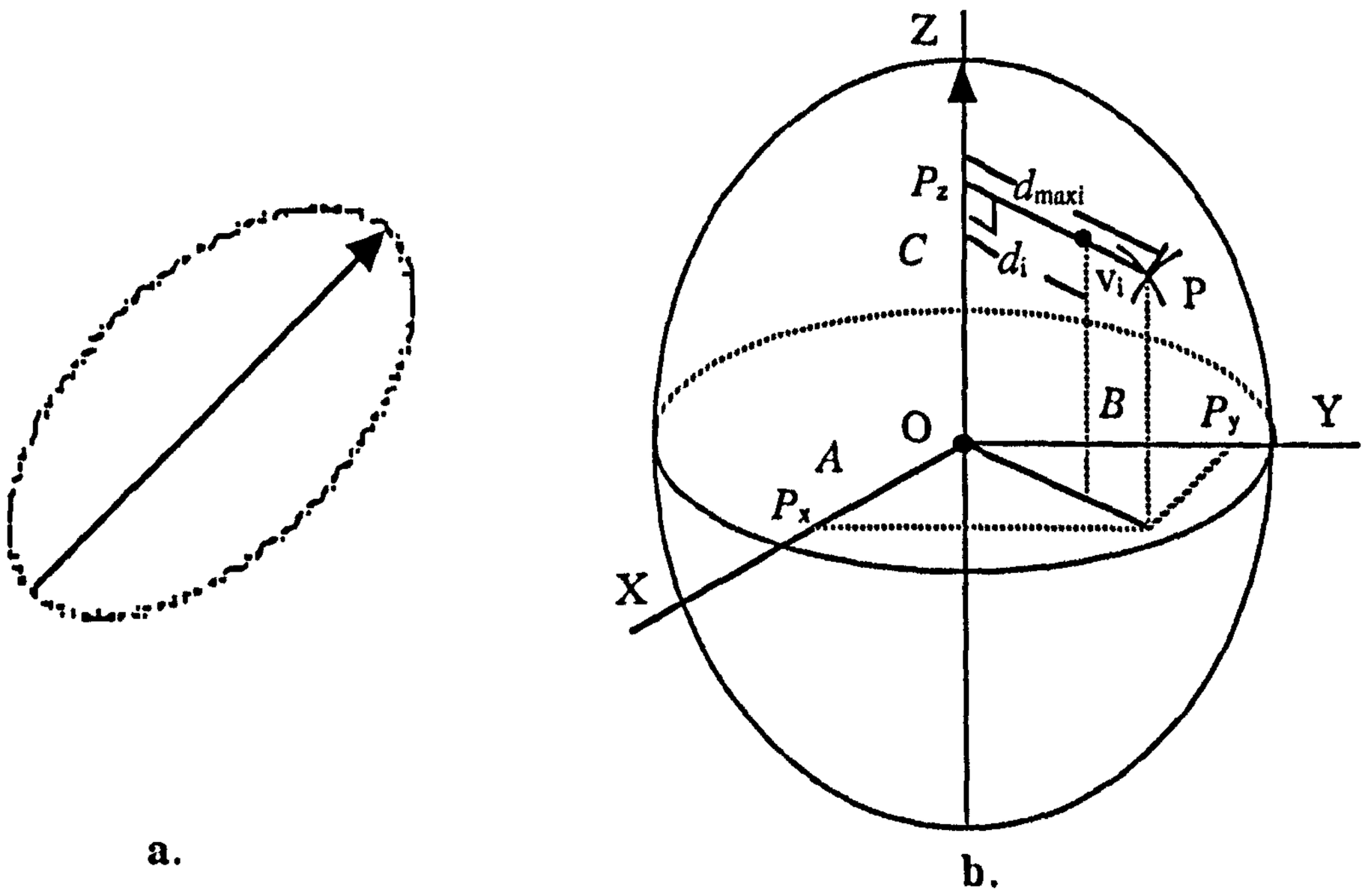


Figure 5.39 A vertex inside the field of a length VOO.

The objective, as in the case of polar fields, is to calculate the two absolute distances d_i and d_{maxi} which will then be used in equation [5.29]. The distance d_i in this case is the perpendicular distance of vertex v_i from the Z axis and may be calculated as follows:

$$d_i = \sqrt{v_{xi}^2 + v_{yi}^2} \qquad [5.40]$$

For the sake of efficiency, the two minor radii A and B of the ellipsoid of a length field are assumed to be equal. In figure 5.39b, P is a point on the surface of the ellipsoid and therefore its coordinates satisfy the analytic equation of the ellipsoid:

$$\left(\frac{P_x}{A}\right)^2 + \left(\frac{P_y}{B}\right)^2 + \left(\frac{P_z}{C}\right)^2 = 1 \qquad [5.41]$$

So, with $A = B$ equation [5.41] becomes:

$$\sqrt{P_x^2 + P_y^2} = A\sqrt{1 - \left(\frac{P_z}{C}\right)^2} \qquad [5.42]$$

The left part of equation [5.42] is equal to the distance d_{maxi}. The coordinate P_z in the right part of the equation, is equal to the z coordinate of vertex v_i. Therefore, equation [5.42] may rewritten as:

$$d_{maxi} = A\sqrt{1 - \left(\frac{v_{zi}}{C}\right)^2} \qquad \text{where } A \neq 0 \text{ and } C \neq 0 \qquad [5.43]$$

In the case where the two minor radii of the ellipsoid A and B were not equal, the calculation of distance d_{maxi} would involve the calculation of the intersection between a line and an ellipsoid. This would result to a non-trivial calculation overhead.

5.5.4 Inside an Ellipsoid

The field of influence of a polar or length VOO may contain several objects. Some vertices of an object may lie inside a field and some may lie outside (see figure 5.40).

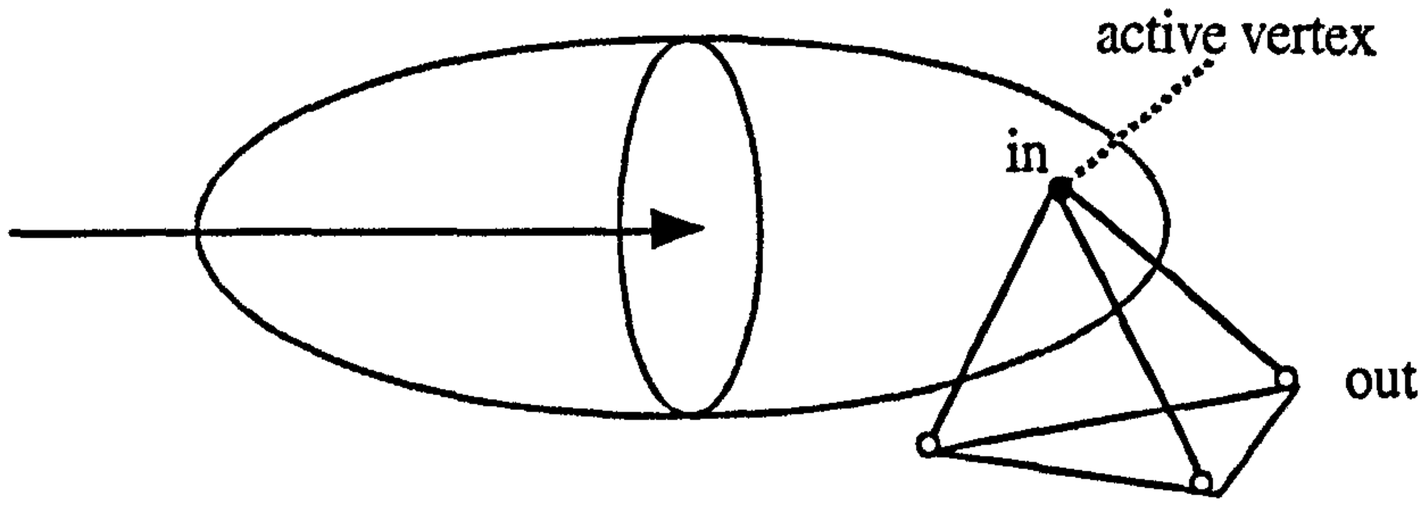


Figure 5.40 An object partially in and partially out of a field of influence.

If the coordinates of a vertex v_i satisfy the following inequality then the vertex lies inside the ellipsoid and it is characterised as active (see figure 5.40):

$$\left(\frac{v_{xi}}{A}\right)^2+\left(\frac{v_{yi}}{B}\right)^2+\left(\frac{v_{zi}}{C}\right)^2 \leq 1 \qquad [5.44]$$

Every vertex of an object must be tested using this inequality [5.44]. This creates a set of active vertices which will be processed further by the deformation model. This improves the efficiency of the model because a lot of unnecessary calculations are avoided by excluding large parts of an object from the deformation process.

5.6 Precalculation of the Damping Element

In the previous sections, two different types of field of influence were used to calculate the normalised distance δ_i for every vertex that comes under the influence of a VOO. This distance is used in the calculation of a damping value for the plastic strain vector of each vertex v_i (see equation [5.23] in section 5.4.3.3):

$$\mathbf{Q}_{pi}(t) = \mathbf{Q}_i(t)\left(1 - \gamma \cdot dampf\left(\delta_i + \lambda\right)^{\omega}\right) \qquad [5.45]$$

The damping element of equation [5.45] is not related to time. It depends on the damping function and the three strain control coefficients, γ, λ and ω, which are properties of a VOO. It also depends on the normalised distance, δ_i, which expresses the relative position of a vertex in relation to a VOO and its field of influence at rest state. Therefore, this damping element may be isolated from equation [5.45] and be precalculated:

$$damp_i = \left(1 - \gamma \cdot dampf\left(\delta_i + \lambda\right)^{\omega}\right) \qquad [5.46]$$

So, prior to any deformation, a precalculation of the damping elements, $damp_i$, must be performed for every active vertex (i.e. vertex that lies inside a field of influence). This process will result in a set of active vertices and their precalculated damping elements. This increases the memory requirement of the model and causes an initial computation overhead. However, the precalculation process greatly improves the efficiency of the time related deformation. The set of damping elements, $damp_i$, must only be recalculated every time the rest state of a VOO or its field of influence are modified. So, during the actual deformation process, the damping elements are being accessed directly.

5.7 Vertex Deformation

Any movement of a VOO will result in an undamped strain vector $\mathbf{Q}_i(t)$ for every vertex v_i that lies inside the field. This vector will have to be multiplied by the damping element of the same vertex in order to yield the plastic strain vector $\mathbf{Q}_{pi}(t)$:

$$\mathbf{Q}_{pi}(t) = \mathbf{Q}_i(t) \cdot damp_i \qquad [5.47]$$

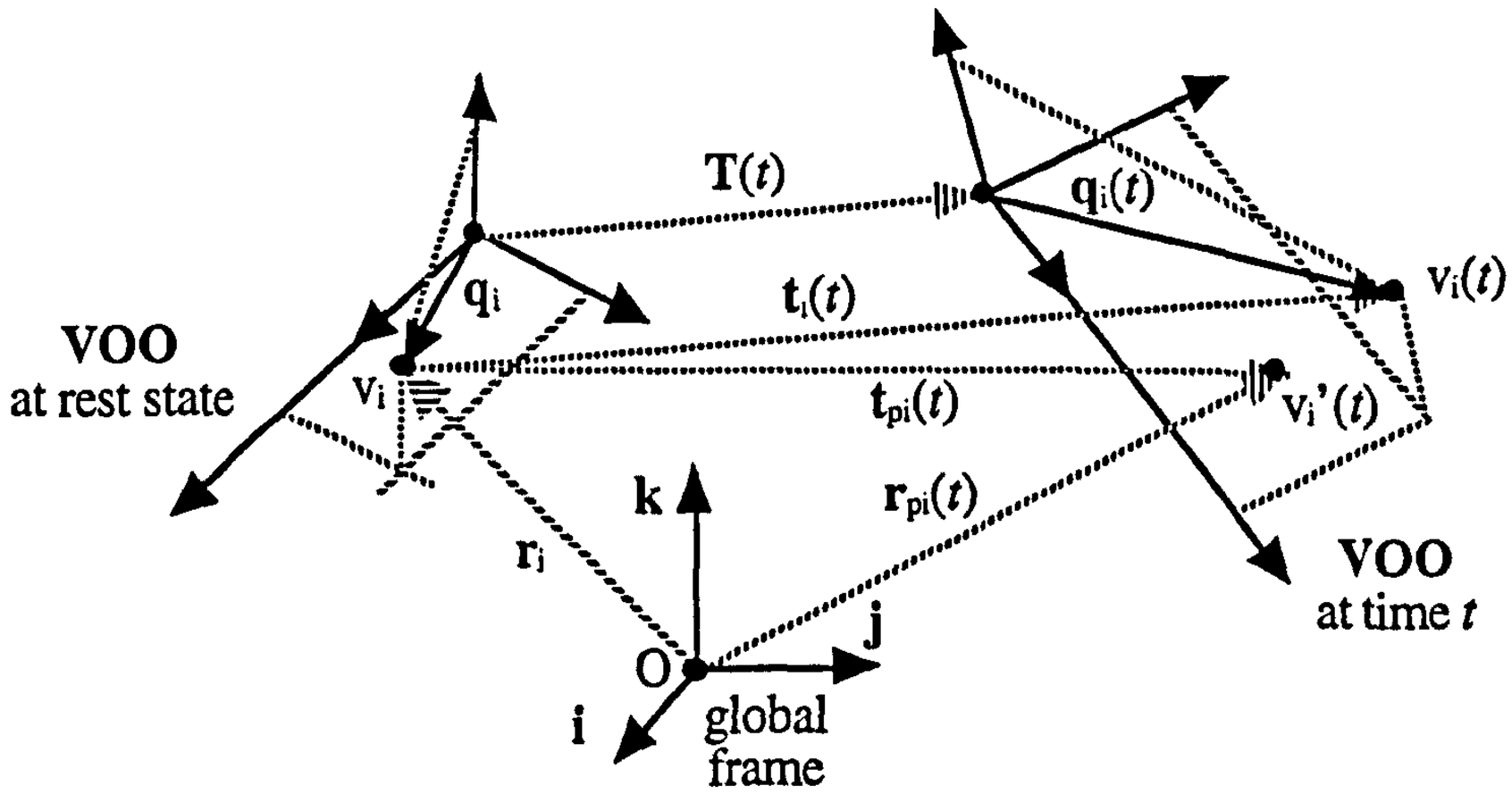


Figure 5.41 Vector $\mathbf{t}_{pi}(t)$ represents the total deformation of vertex v_i at time t.
Vertex v_i is at rest state, $v_i(t)$ is the rigid transformed vertex
and $v_i'(t)$ is the deformed vertex.

The plastic strain vector $\mathbf{Q}_{pi}(t)$ will, then, be used to calculate the total deformation vector $\mathbf{t}_{pi}(t)$ and the new position of the deformed vertex $v_i'(t)$ at time t (see figure 5.41):

$$\mathbf{t}_{pi}(t) = \mathbf{T}(t) + \mathbf{Q}_{pi}(t) \qquad [5.48]$$

$$\mathbf{r}_{pi}(t) = \mathbf{r}_i + \mathbf{t}_{pi}(t) \qquad [5.49]$$

where $\mathbf{T}(t)$ is the global translation vector, $\mathbf{r}_i$ is the global position vector of vertex v_i at rest state and $\mathbf{r}_{pi}(t)$ is the global position vector of the deformed vertex $v_i'(t)$ at time t. The plastic strain vector $\mathbf{Q}_{pi}(t)$ cannot be drawn in figure 5.41 because of the existence of global translation $\mathbf{T}(t)$. Vector $\mathbf{Q}_{pi}(t)$ appears in figures 5.15b and 5.16.

5.8 Set of VOOs

So far in this chapter, it was assumed that only one VOO affects a given object. However, a set of VOOs may be instructed to operate upon a set of objects. This may frequently result in a situation where a given vertex of an object is affected by two or more VOOs (see figure 5.42)

(see also colour plate 5.56 in section 5.11). In such a case, the precalculation process will generate a list of damping elements for each active vertex v_i, one for each contributing VOO:

$$damp_{ij} = \left(1 - \gamma_j \cdot dampf_j\left(\delta_{ij} + \lambda_j\right)^{\omega_j}\right) \quad \text{where } 1 \le j \le nv \qquad \textbf{[5.50]}$$

where subscript $_i$ denotes a vertex, subscript $_j$ denotes a VOO and nv is the number of contributing VOOs for vertex v_i.

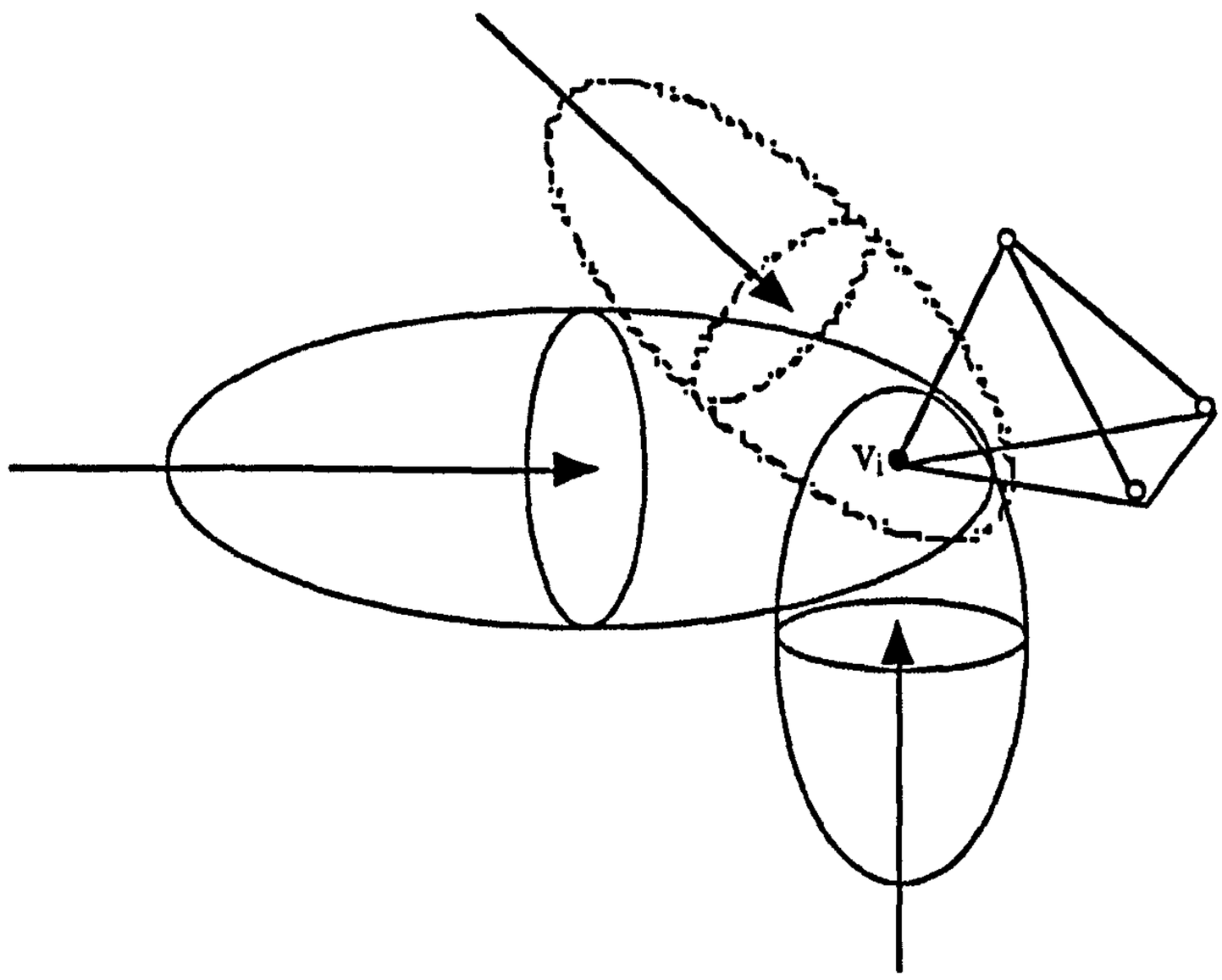


Figure 5.42 A vertex under the influence of a set of VOOs.

Each active vertex v_i receives a set of plastic strain vectors, one from each VOO:

$$\mathbf{Q}_{pij}(t) = \mathbf{Q}_{ij}(t) \cdot damp_{ij} \qquad \textbf{[5.51]}$$

The total deformation vector, $\mathbf{t}_{pi}(t)$, of a vertex at time t is then given by:

$$\mathbf{t}_{pi}(t) = \sum_{j=1}^{nv}\left(\mathbf{T}_j(t) + \mathbf{Q}_{pij}(t)\right) \qquad \textbf{[5.52]}$$

where *nv* is the number of contributing VOOs for vertex v_i.

5.9 Results

5.9.1 Effectiveness of the Plastic Model

A vector offset operator is an interactive strain application tool which offers a simple and effective interface for local deformation. The user may define VOOs in 3D space and manipulate their fields of influence in order to clearly specify the areas(s) of the object(s) that will be subjected to deformation. Thus, influence fields serve as spatial constraints which limit the deformation effect of a VOO. Further, offsetting the tip or the head of a VOO or rotating a VOO about its major axis away from a rest state causes deformation to local space inside the field of influence. Manipulation of a damping graph and of the set of strain and volume control coefficients enables a fine tuning of the spatial behaviour of the deformation.

When a VOO is used as an interactive strain application tool, it offers an intuitive and direct way of controlling the shape of a deformation. This is the main advantage over more conventional deformation techniques such as free form deformation, which is based on the manipulation of control points. A VOO may also be used as a tool for animated deformation. For this purpose, a VOO may be subjected to conventional key-framing animation techniques, which define key states at specified times of an animation. Interpolation between these key states of a VOO may result in a smooth deformation of the objects under its influence. Another approach is to apply procedural animation to the time related state of a VOO, using a scripting language.

A VOO creates an intermediate layer between the user / animator and the actual deformation process of CG objects. This provides a more efficient approach to deformation and simplifies the application of time and space constraints to the deformation process. Also, it enables further development of the deformation model by allowing to incorporate physical properties to the modelling process (see chapters 6, 7 and 8).

Colour plates relevant to this chapter can be found in section 5.11. Also, see animation tests 1, 2 and 3 in video tape, time code: from 10:04:05:00 to 10:04:46:00.

5.9.2 Performance of the Plastic Model

The efficiency of the proposed method is primarily based on the implementation of a simple local deformation model. The main objective of the model is to calculate a plastic strain vector $\mathbf{Q}_{pij}(t)$ for every active vertex for every active VOO at time t (see section 5.4). Moreover, the involvement of ellipsoids as fields of influence offers a very efficient mechanism for identifying the set of active vertices of a VOO (see section 5.5). Finally, the precalculation of the damping elements for every active vertex / for every active VOO also contributes to the efficiency of the model (see section 5.6).

To measure the speed of the deformation model, several tests have been carried out. A basic performance test was set as follows: Using the interactive framework, one polar VOO with only an active tip, was set to operate on a single object. The object consisted of 200 vertices. An offsetting of the tip was performed over 25 frame steps, while the duration of the process was being timed. If t was the timing in seconds then the process performed $(25/t)$ frames/sec, including the deformation calculations and the drawing process. The test was then repeated, either by adding more objects or by adding more VOOs to the scene. The tests were carried out on an HP 715/9000 series with 32 Mb RAM, running at 75 MHz and capable of 30 Mflops.

5.9.2.1 Performance Test 1

The basic performance test was repeated by adding extra copies of the initial object inside the influence field of one VOO and the process was timed again. The results are shown in the graph of figure 5.43 (see section 5.11). The same test was performed with precalculation of the damping elements (see blue line in graph) and without precalculation (see red line in graph). The graph in figure 5.43 shows that the suggested deformation model is capable of deforming

objects of considerable size at speeds suitable for interactive manipulation. It is also shown that the precalculation of the damping elements offers an increase in speed of deformation of approximately between 16% (for 200 vertices) and 21% (for 1200 vertices).

5.9.2.2 Performance Test 2

The basic performance test was repeated by adding extra VOOs and the process was timed again. The results are shown in the graph of figure 5.44. The test was performed with and without precalculation of the damping elements. The graph in figure 5.44 shows that the suggested deformation model is capable of deforming objects using several VOOs at speeds suitable for interactive manipulation. It is also shown that the precalculation of the damping elements offers an increase in speed of deformation of approximately between 16% (for one VOO) and 20% (for six VOOs).

5.10 Summary and Conclusion

This chapter presented the theoretic base of a local deformation. Then, a new tool for applying such local deformations was introduced.

A vector offset operator is a strain application tool which is suitable for interactive, time based manipulation of deformable objects. Interactive manipulation of VOOs offers an intuitive way for applying complex deformations and for fine tuning the spatial behaviour of the deformable objects. Spatial damping of strain was introduced by allowing the user to define damping graphs which are complemented with strain control coefficients of flexibility, locality and weight. The volume of a deformed object may be controlled by manipulating the elongation, volume preservation and cross-sectional coefficients. Fields of influence were introduced in order to localise the deforming effect of VOOs and to increase the efficiency of the model by predetermining the set of active vertices of an object. Precalculation of the damping elements of an object contributes to the efficiency of the deformation model.

The isolation of the strain vector $\mathbf{Q}_i(t)$ for every vertex of an object enables modular development of the model. Extension of the model towards the physical properties of plasticity, elasticity and viscoelasticity is achieved by applying further processing to the strain vector. This further ensures independence of representation by allowing the same strain vector to be applied to any 3D object descriptions, such as control points of surface data or volumetric data.

The original concepts introduced in this chapter are:

- the concept of a strain application tool used to produce a local deformation,
- the concept of precalculation of damping elements.

The following chapter will extend the deformation model to incorporate elastoplastic material behaviour.

5.11 Colour Plates

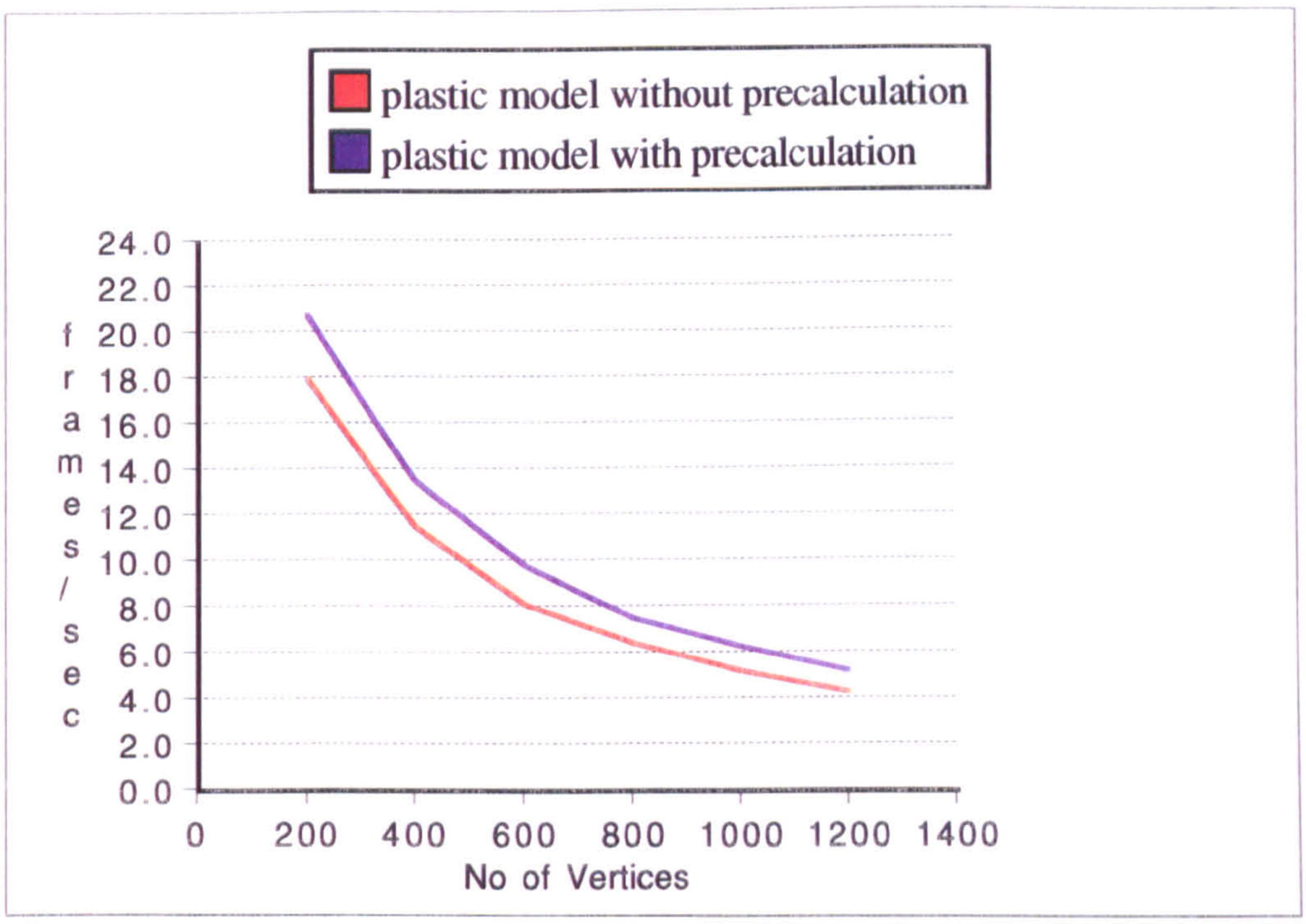


Figure 5.43 Graph of speed of deformation over the number of vertices of an object.

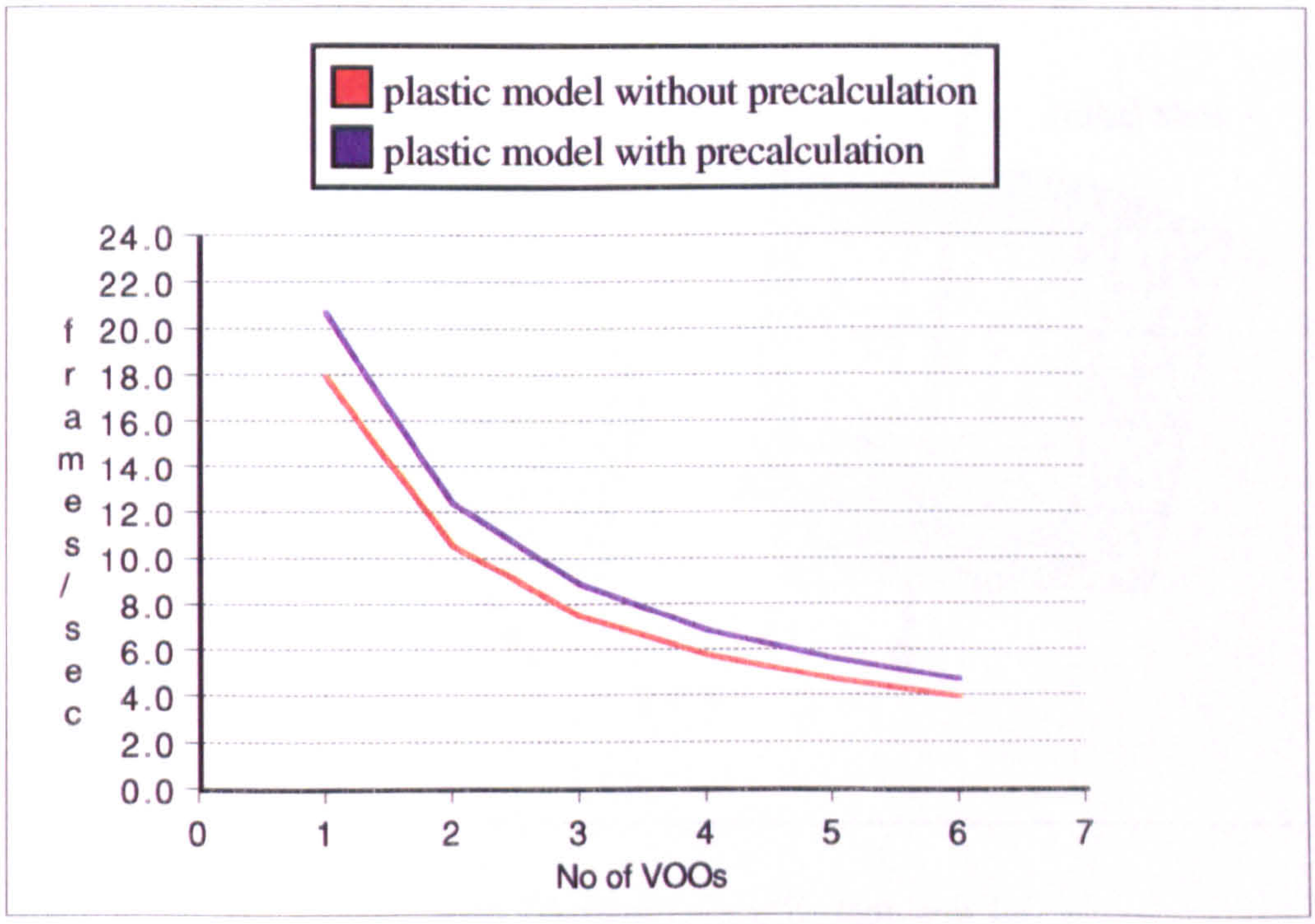


Figure 5.44 Graph of speed of deformation over the number of VOOs.

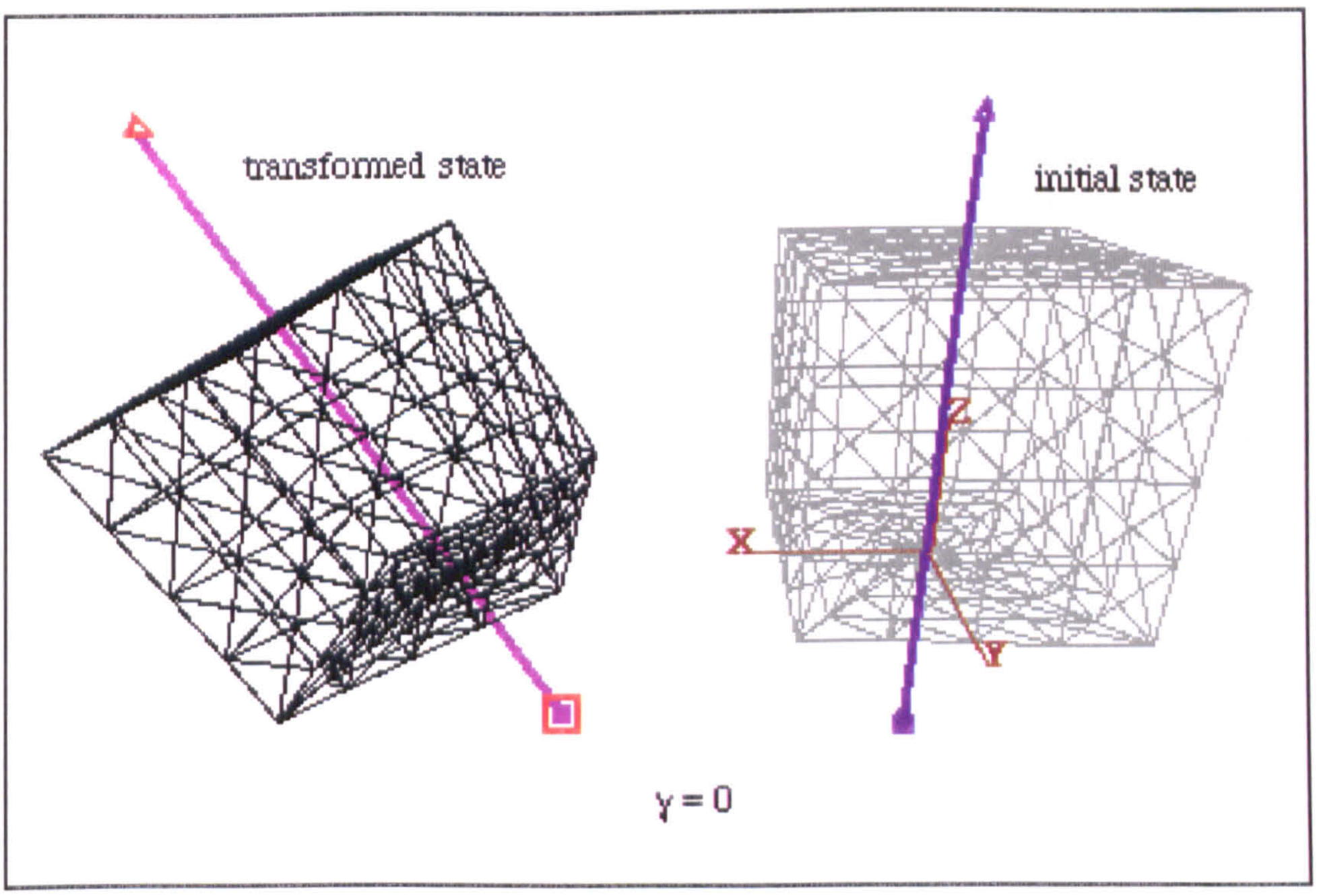


Figure 5.45 Flexibility coefficient, $\gamma = 0$.
An object is subjected to global translation and local rotation.

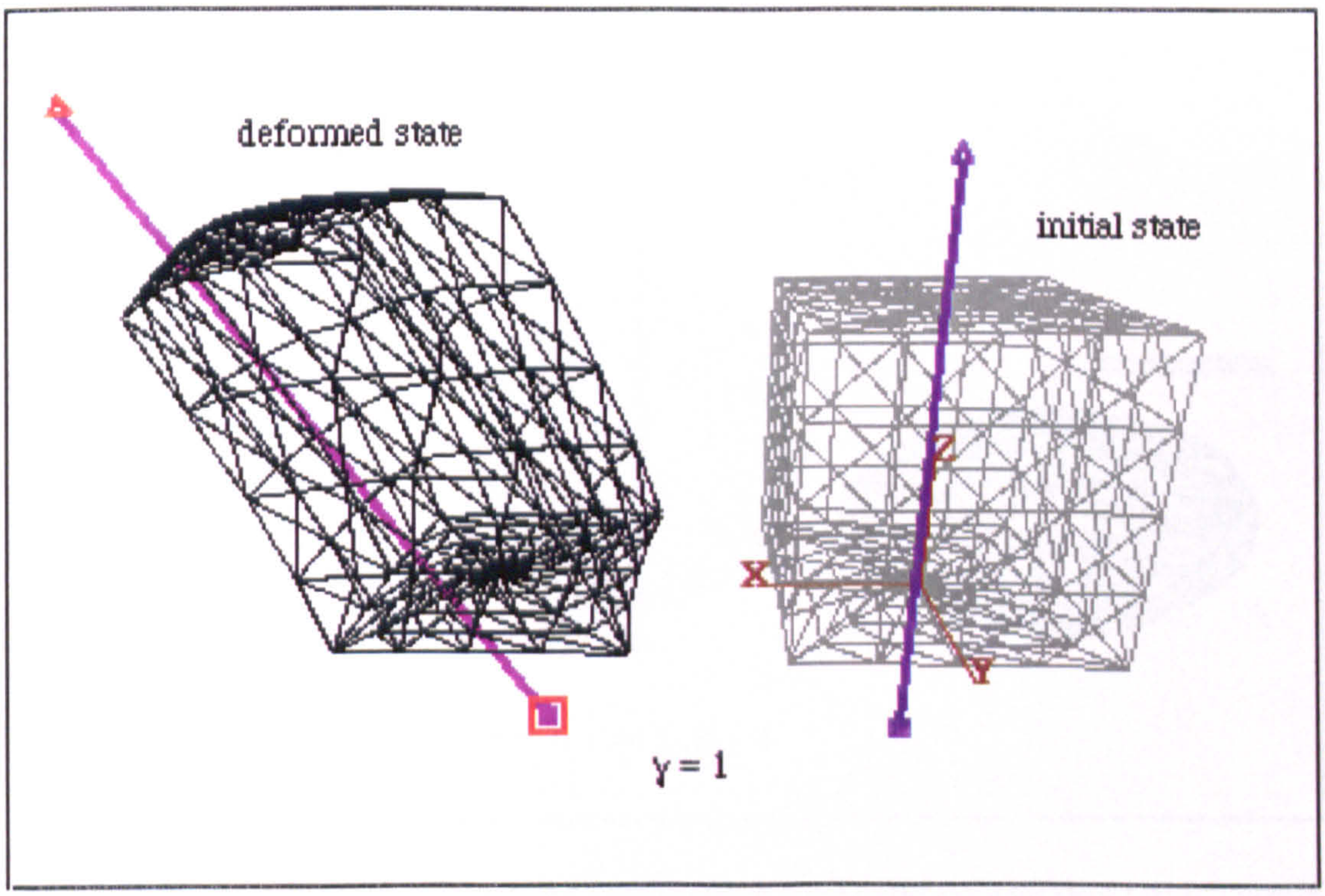


Figure 5.46 Flexibility coefficient, $\gamma = 1$.
An object is subjected to a combined elongation and bending strain.

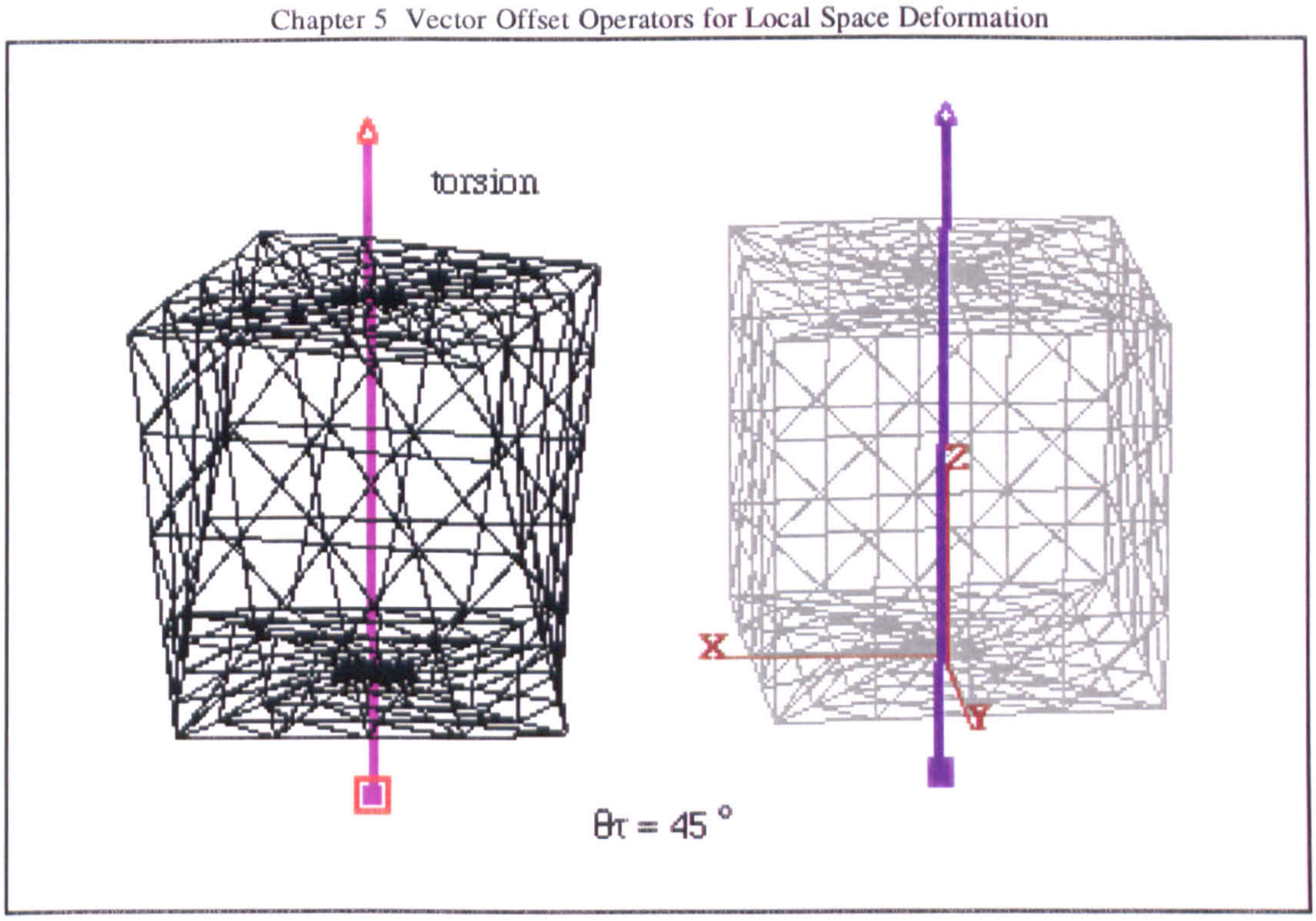


Figure 5.47 An object is subjected to torsion strain, $\theta_t = 45°$.

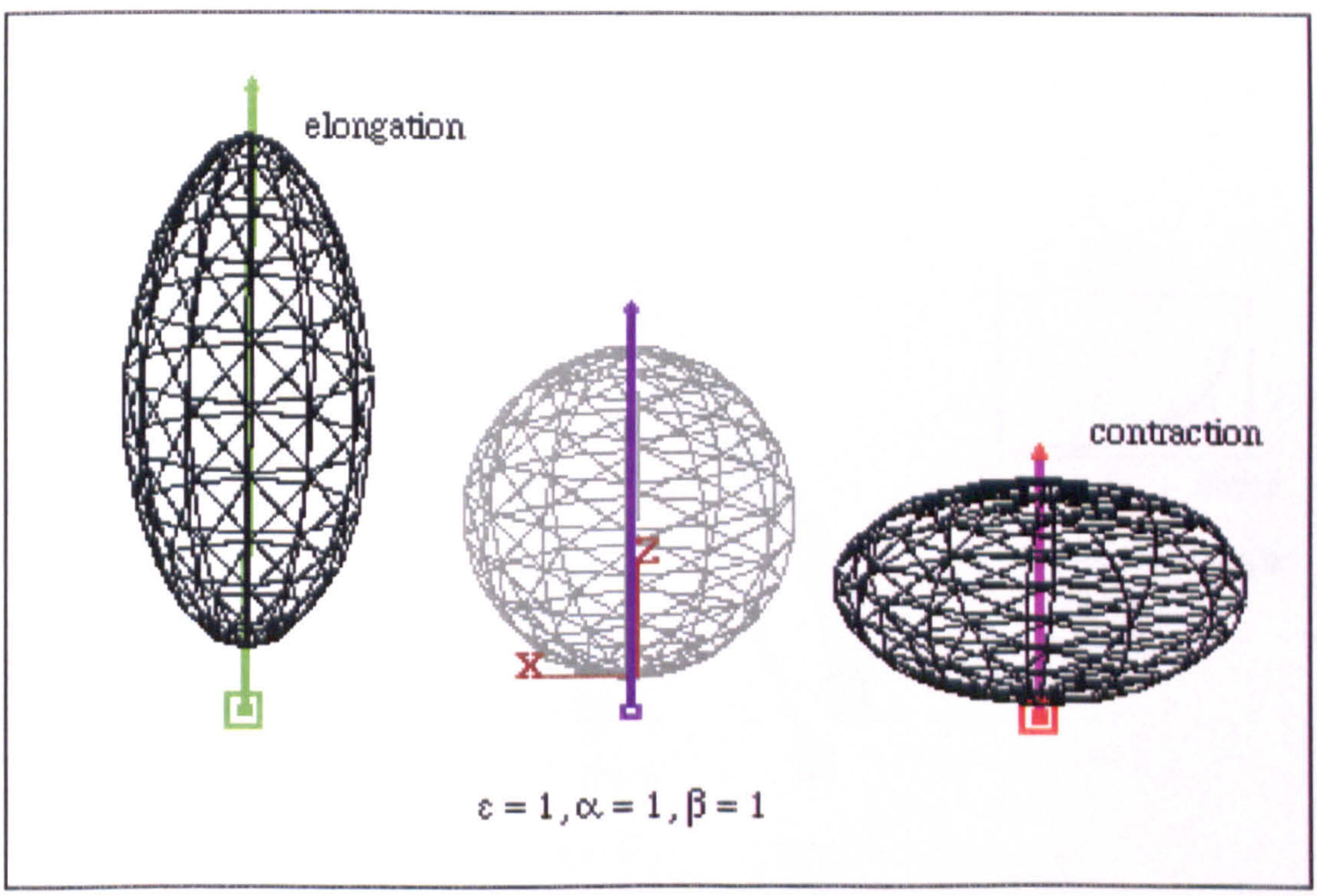


Figure 5.48 An object is subjected to elongation (left) and contraction (right) strain. Elongation coef. $\varepsilon = 1$, volume preservation coef. $a = 1$, cross section coef. $b = 1$.

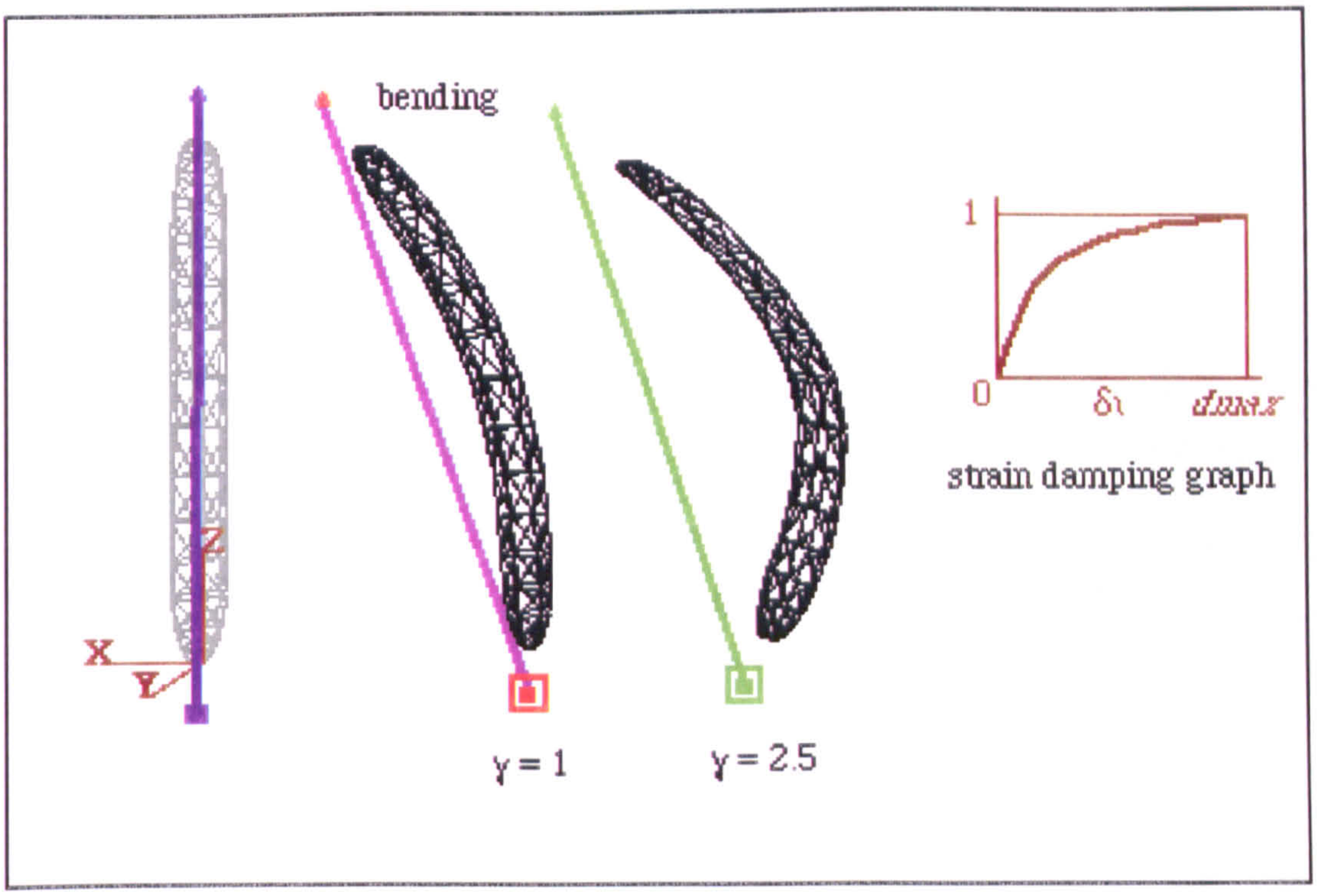


Figure 5.49 Combining the effects of spatial strain damping graph and flexibility coefficient, γ, to achieve different degrees of bending strain.

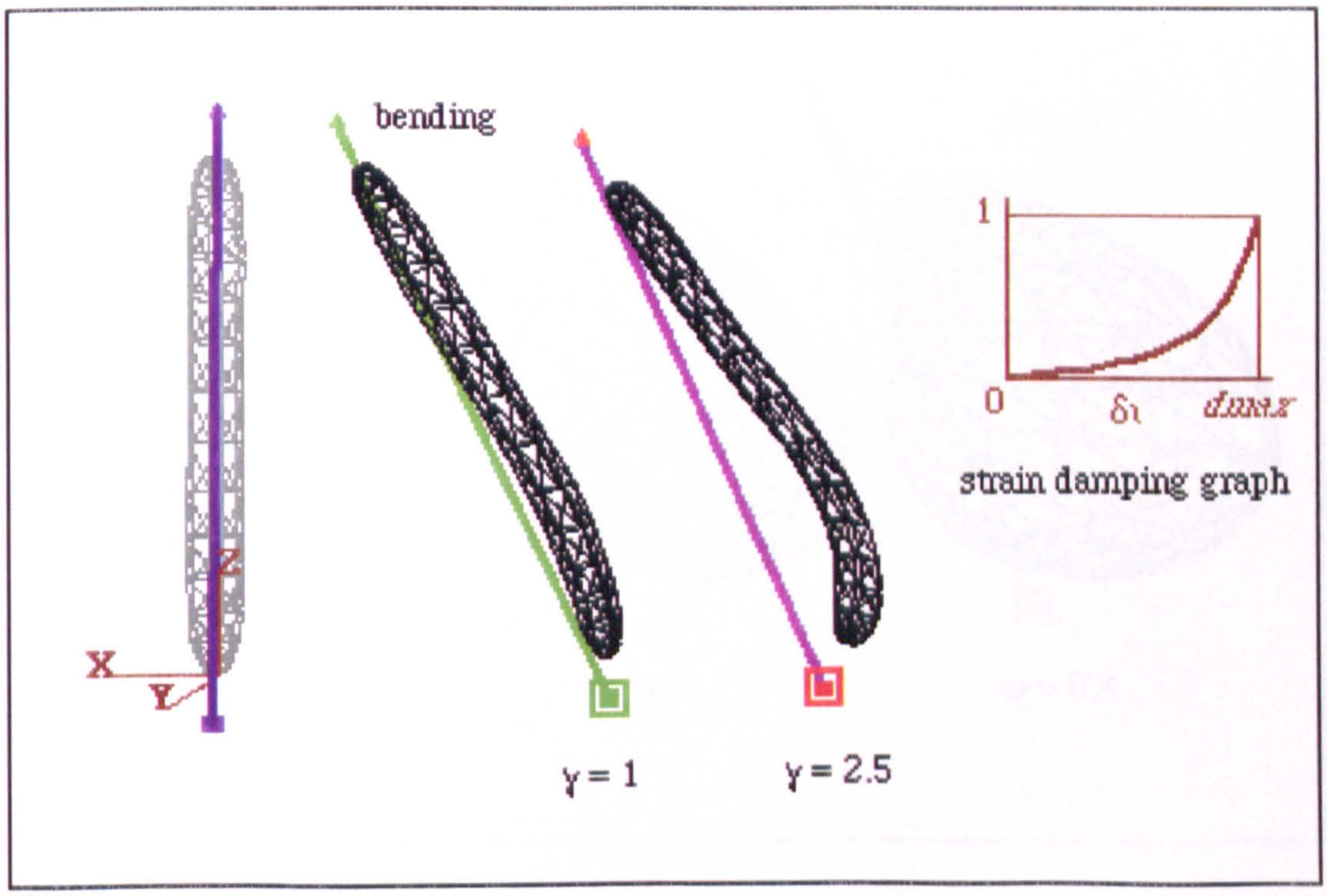


Figure 5.50 The same effect (see figure 5.49) with a different damping graph.

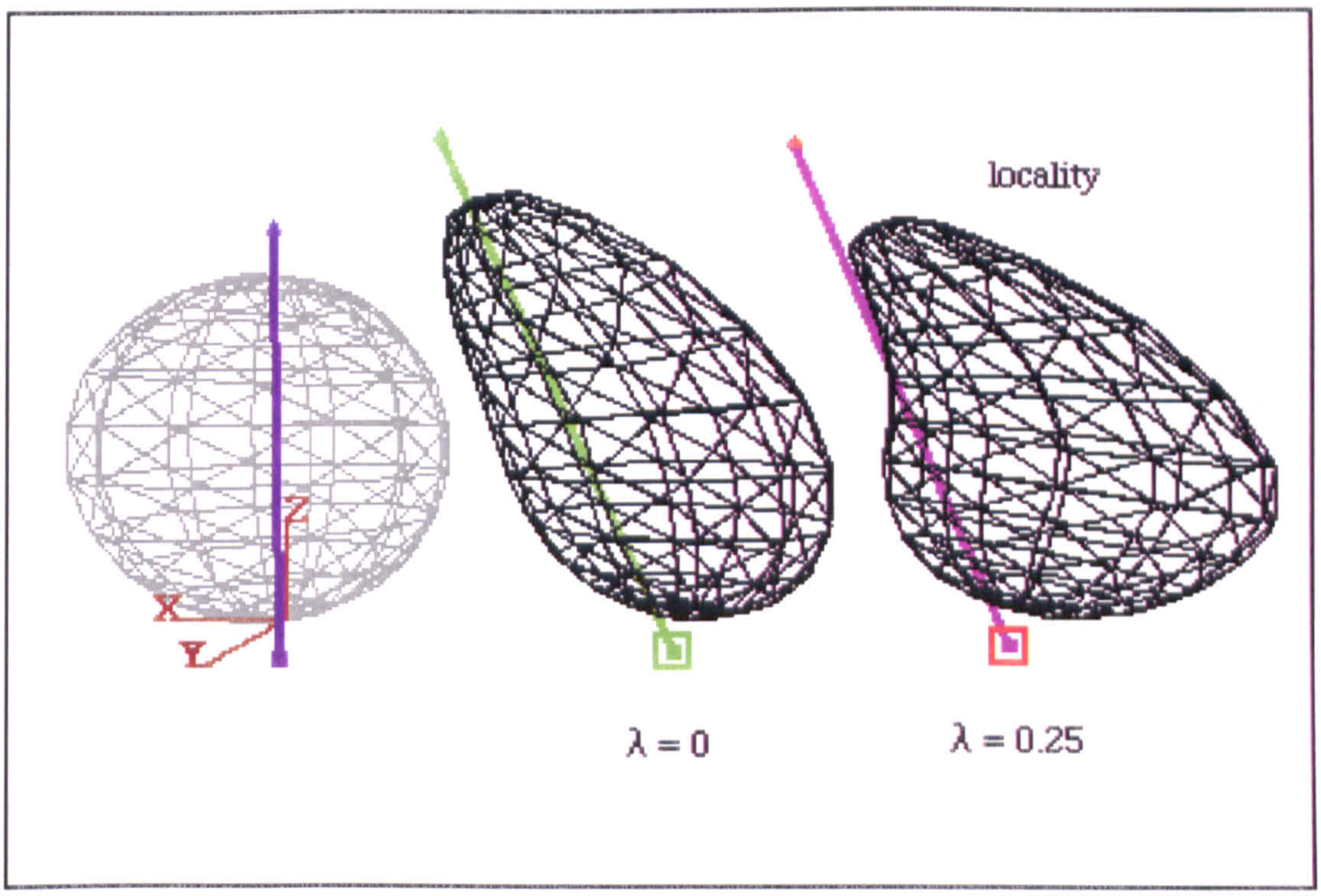


Figure 5.51 The effect of the locality coefficient, λ, on bending strain.

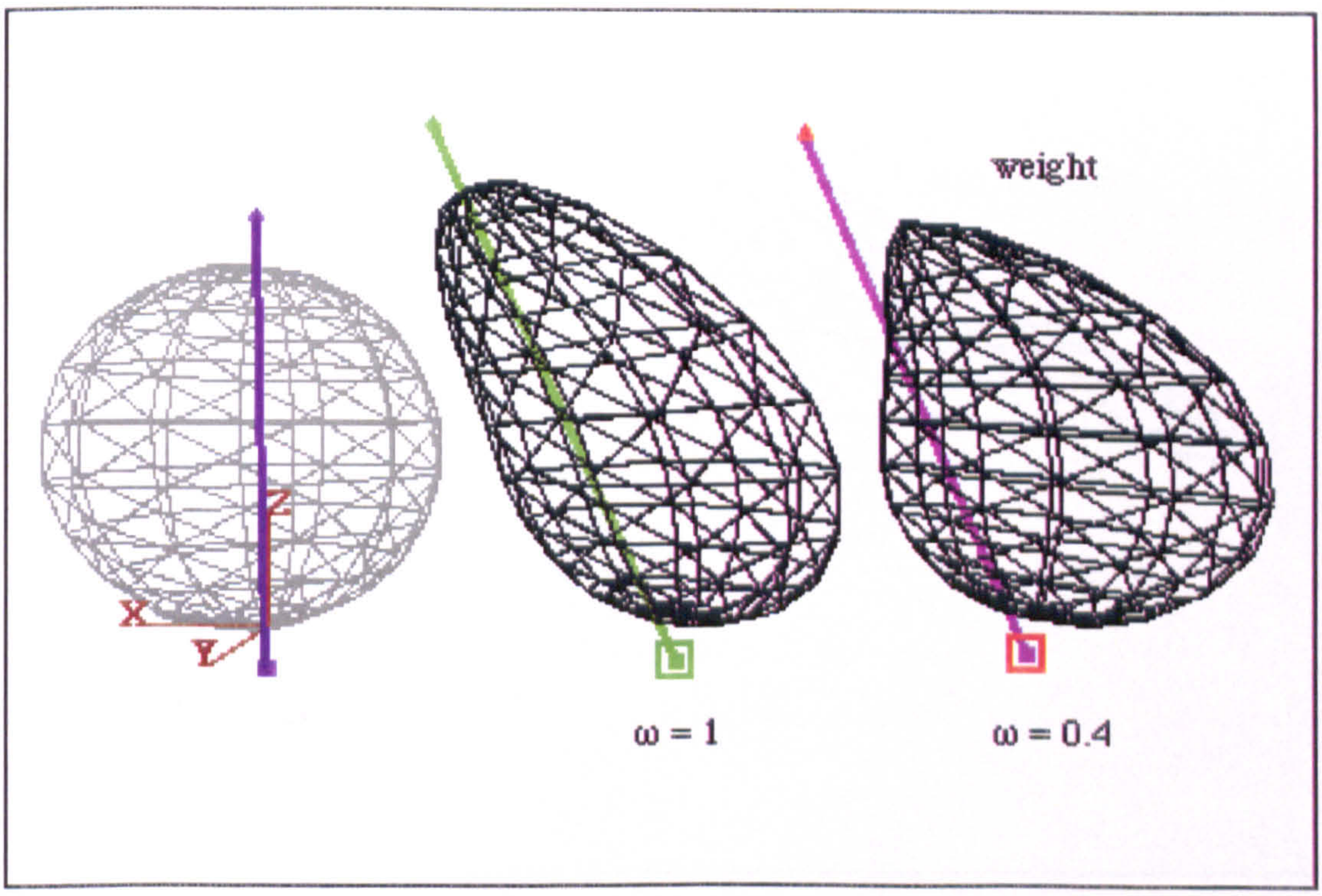


Figure 5.52 The effect of the weight coefficient, ω, on bending strain.

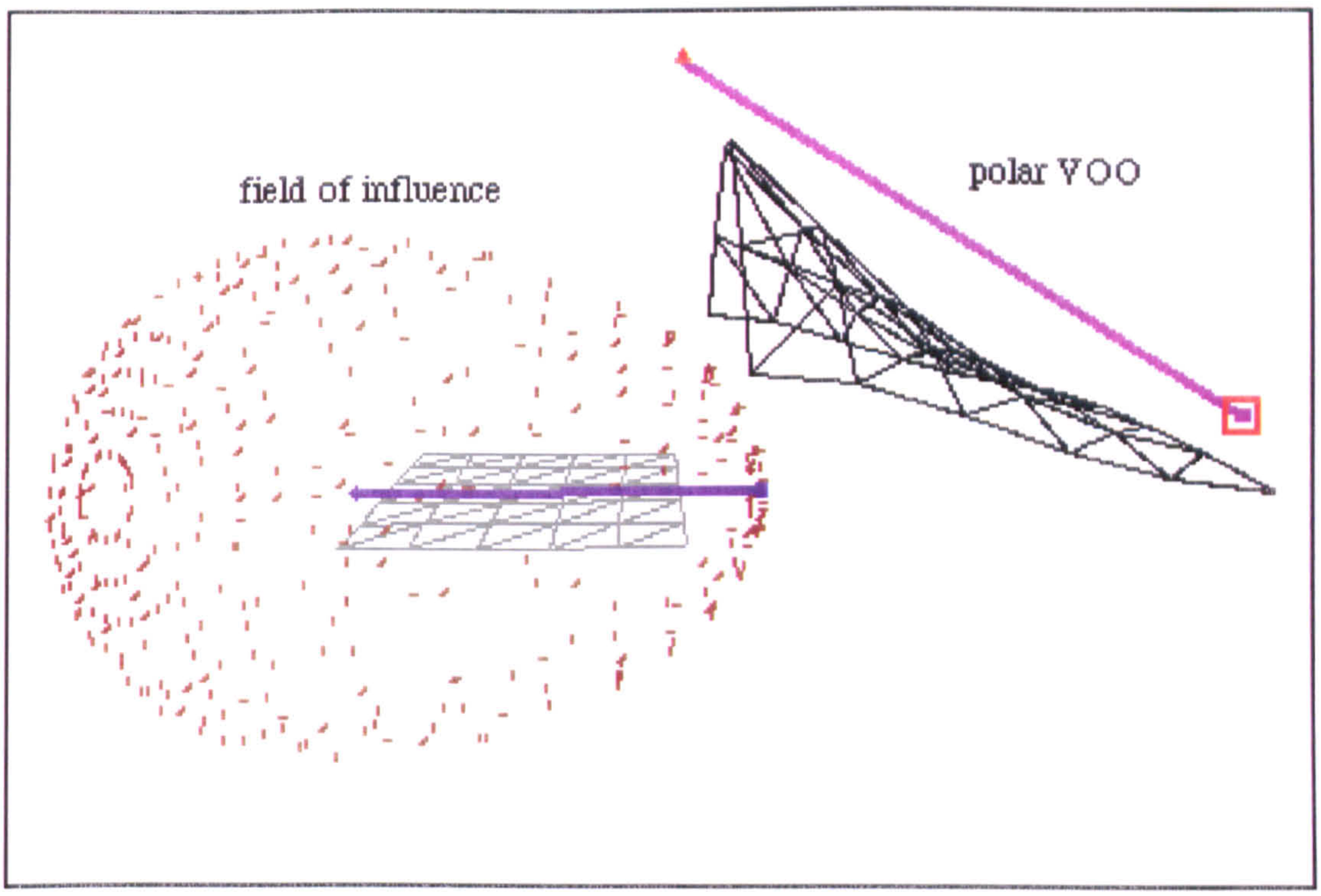


Figure 5.53 A polar VOO with influence field around its active tip.

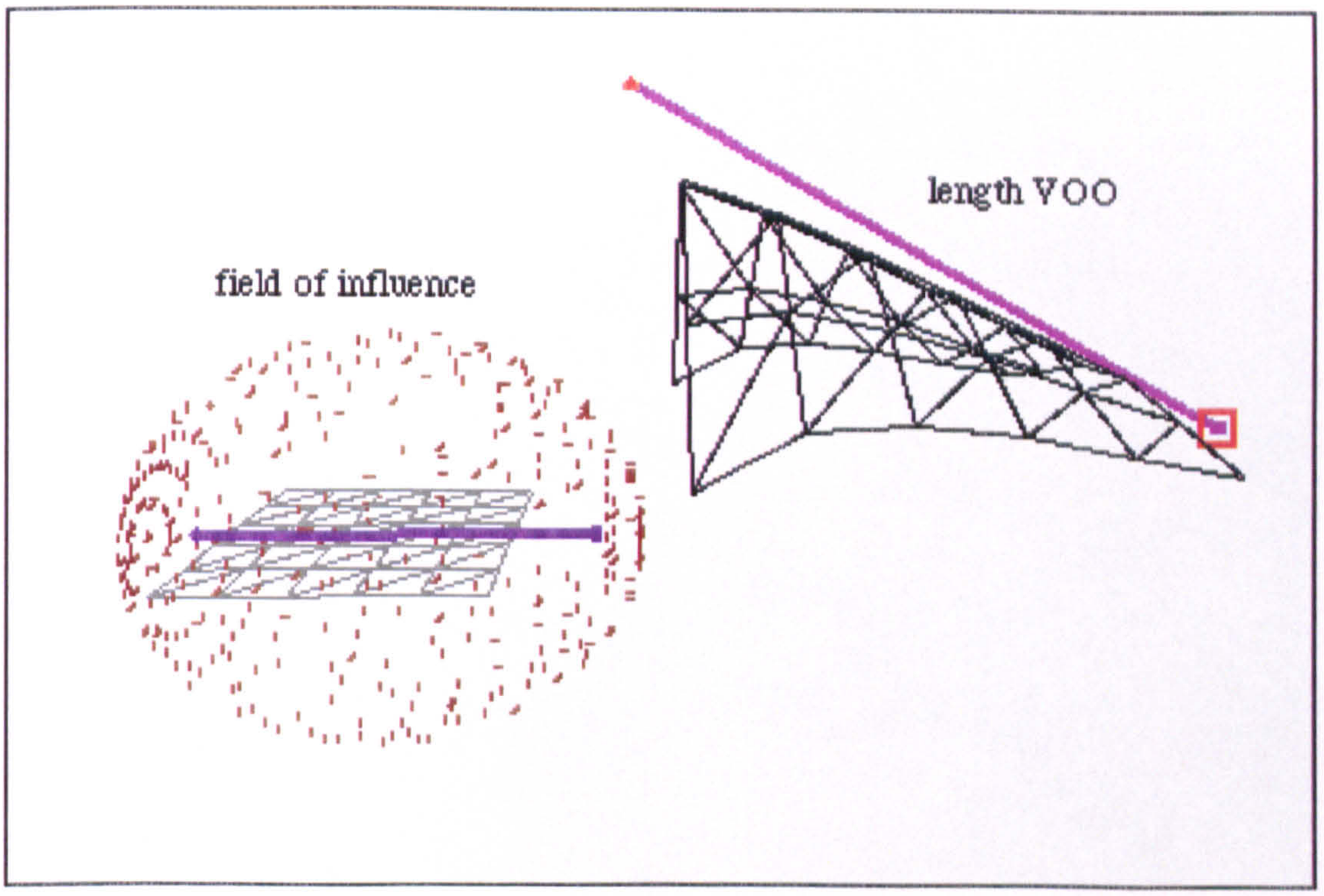


Figure 5.54 A length VOO with its influence field.

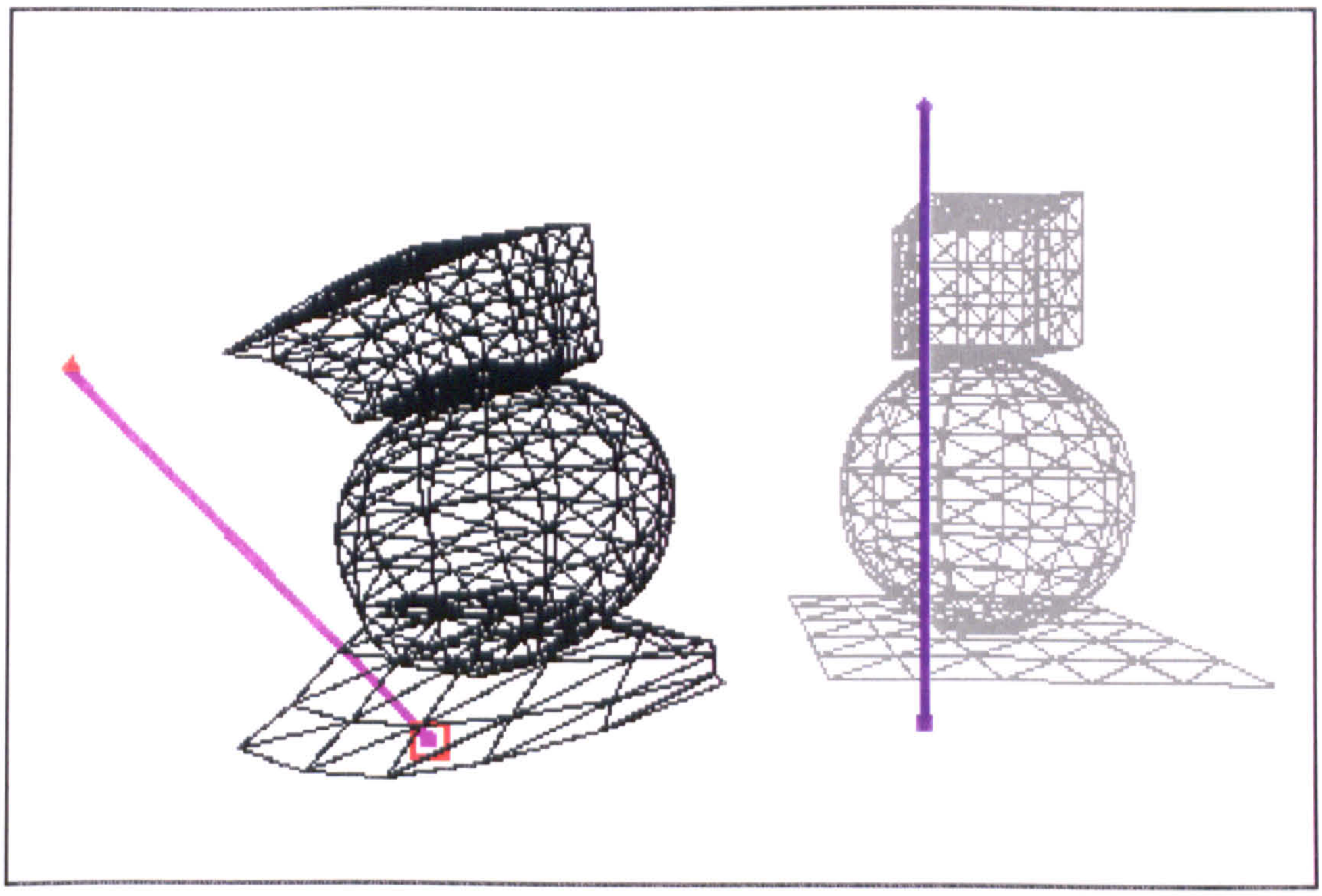


Figure 5.55 One VOO operating on three objects.

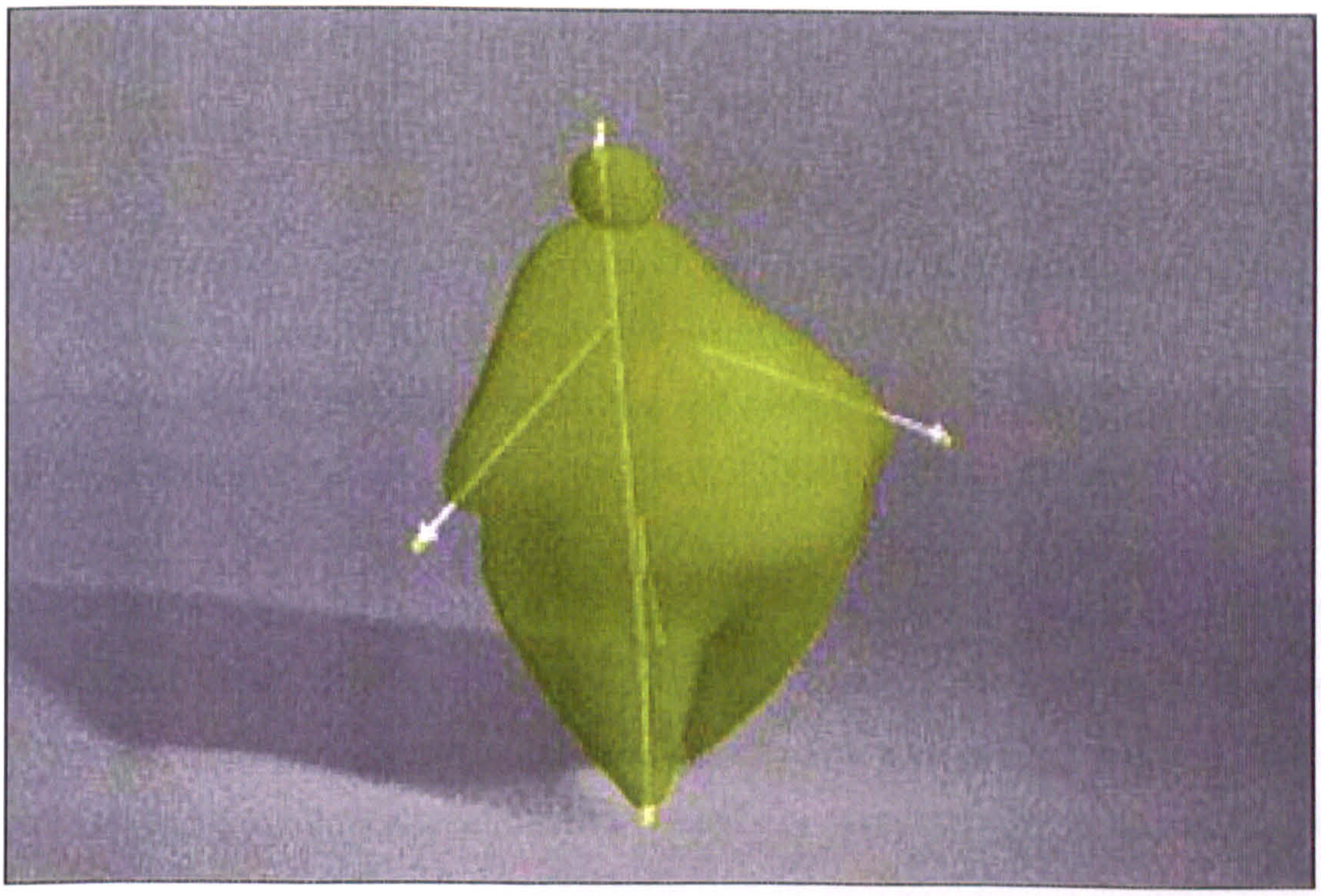


Figure 5.56 Five VOOs operating on two objects
(see animation test 16 in video tape, time code: from 10:07:37:00).

6

The Elastoplastic Model

6.1 Introduction

Perfectly elastic objects tend to spontaneously resume their normal shape after distortion. However, in most natural materials, if the deformation exceeds a certain limit the object only partially reverts to its normal shape leaving a permanent plastic deformation. At the other extreme, a perfectly plastic object does not exhibit any resilience at all and accepts any amount of deformation as permanent.

The objectives of this chapter are:

- to extend the deformation model with the properties of elasticity and plasticity,
- to enable interaction with elastically deformable objects,
- to enable efficient animation testing and rendering,
- to enable application of temporal constraints to elastoplastic objects.

The deformation model introduced in the previous chapter, is extended to cover elastic and plastic material behaviours. This is done to improve the realistic look of animated deformable objects by incorporating organic material properties, such as elasticity and plasticity. The model must be fast so as to allow interaction with elastically restoring objects, as well as efficient animation testing and rendering. A mechanism will be provided for the customisation of the material behaviour of an object, ranging from perfectly elastic to perfectly plastic. Temporal constraints which restrict the duration of elastic restoration will also be introduced.

6.2 The Elastic Model

When some objects are deformed, the relative positions of the molecules of the material are modified. This creates elastic restoring forces which tend to return the object to its original shape after the distorting forces have been removed (see section 3.2).

6.2.1 The Elastic VOO

In the suggested deformation model, a change in the magnitude and direction of a VOO causes deformation of the local space and of any objects inside it. So, instead of attempting to elastically restore each deformed vertex of an object to its initial position, the model attempts to restore a VOO to its initial state.

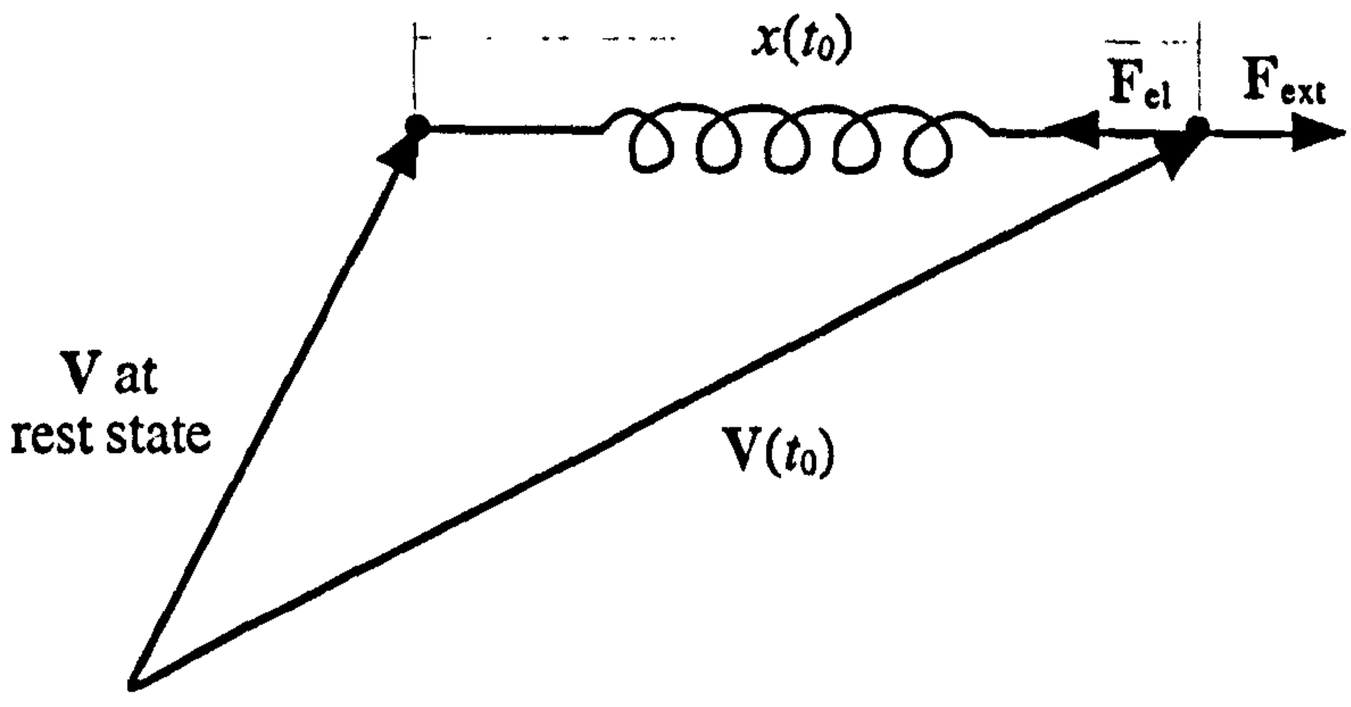


Figure 6.1 A spring connects the tips of **V** and $\mathbf{V}(t_0)$.

In the simplest case where there is only one VOO and it is deformed by offsetting its tip, the task of the restoring process is to bring the deformed vector $\mathbf{V}(t_0)$ back to **V** at the rest state. To achieve this it is assumed that an elastic spring is attached at the tip of the VOO at rest state and at time t_0 (see figure 6.1). It is also assumed that the initial length of the spring is equal to 0. An external force, $\mathbf{F}_{ext}$, is used to stretch the spring to a new length, $x(t_0)$, and to bring the tip of the VOO to its new position at time t_0. The elastic spring force, $\mathbf{F}_{el}$, is opposite to $\mathbf{F}_{ext}$ and it

attempts to minimise the difference between vectors $\mathbf{V}(t_0)$ and $\mathbf{V}$ as soon as $\mathbf{F}_{ext}$ ceases to exit. According to Hooke's law of elasticity the elastic spring force is given by:

$$\mathbf{F}_{el} = -s \cdot \Delta\mathbf{V}(t_0) \qquad [6.1]$$

where s is the elastic spring constant and $\Delta\mathbf{V}(t_0)$ is the difference between vectors $\mathbf{V}(t_0)$ and $\mathbf{V}$ (see figure 6.2). The elastic spring constant, s, is directly related to the tensile stiffness mentioned in chapter 3 (see equation [3.1] in section 3.2.1).

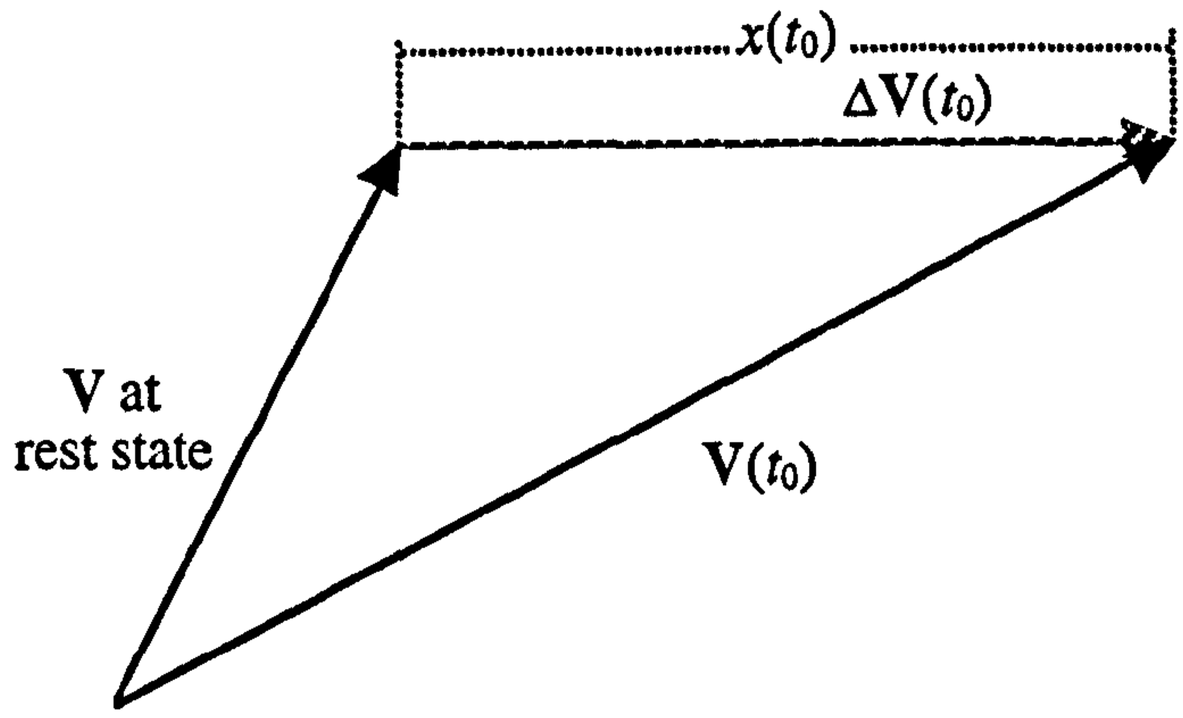


Figure 6.2 A deformed VOO at time t_0.

$$\Delta\mathbf{V}(t_0) = \mathbf{V}(t_0) - \mathbf{V} \qquad [6.2]$$

In figure 6.2, $x(t_0)$ represents the magnitude of vector $\Delta\mathbf{V}(t_0)$. In equations [6.1] and [6.2], t_0 represents the time at which the external force is removed and the elastic restoration process begins. At a later time t, after t_0, elastic restoration is in progress and the VOO is at a new state $\mathbf{V}(t)$. The tip of the VOO travels with acceleration and velocity whose direction is initially opposite to the direction of vector $\Delta\mathbf{V}(t_0)$ (see figure 6.3).

$$\Delta\mathbf{V}(t) = \mathbf{V}(t) - \mathbf{V} \qquad [6.3]$$

$\dot{\mathbf{x}}(t)$ velocity of tip [6.4]

$\ddot{\mathbf{x}}(t)$ acceleration of tip [6.5]

In equations [6.4] and [6.5], the single dot symbolises the first derivative of $\mathbf{x}(t)$ with respect to time and the double dot symbolises the second derivative.

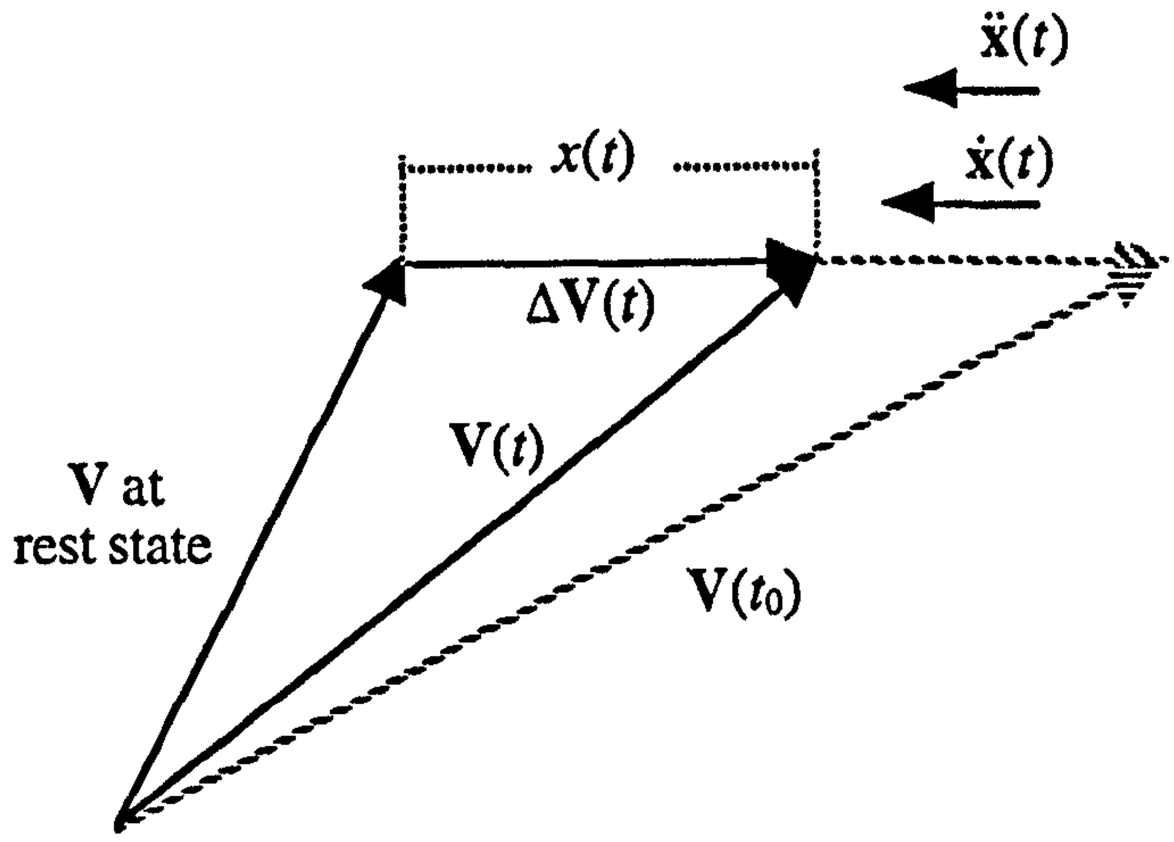


Figure 6.3 Elastic VOO at time *t*.

The problem of elastic behaviour of a deformed object may be seen as the problem of calculating vector $\Delta\mathbf{V}(t)$ from an initial time t_0 until a current time t. Hence, any calculations involved are performed only once per VOO and not per object vertex.

6.2.2 Harmonic Oscillation

Assuming that an amount of mass, m, is concentrated at the tip point of a VOO, then the elastic spring force described in the previous section, will cause a simple harmonic oscillation of the mass point about the rest position of the tip. The spring exerts an elastic force, $\mathbf{F}_{el}$, on the mass point and this results in an acceleration of the mass according to the equation of motion. A combination of the equation of motion and the equation derived from Hooke's law (see equation [6.1]) and using magnitudes only yields:

$$F_{el} = m\ddot{x}(t) = -sx(t) \qquad [6.6]$$

$$\Rightarrow \quad \ddot{x}(t) = -\left(\frac{s}{m}\right)x(t) \qquad \text{(where } m > 0\text{)} \qquad [6.7]$$

In equations [6.6] and [6.7], the scalar values represent the magnitudes of the force, acceleration and displacement vectors.

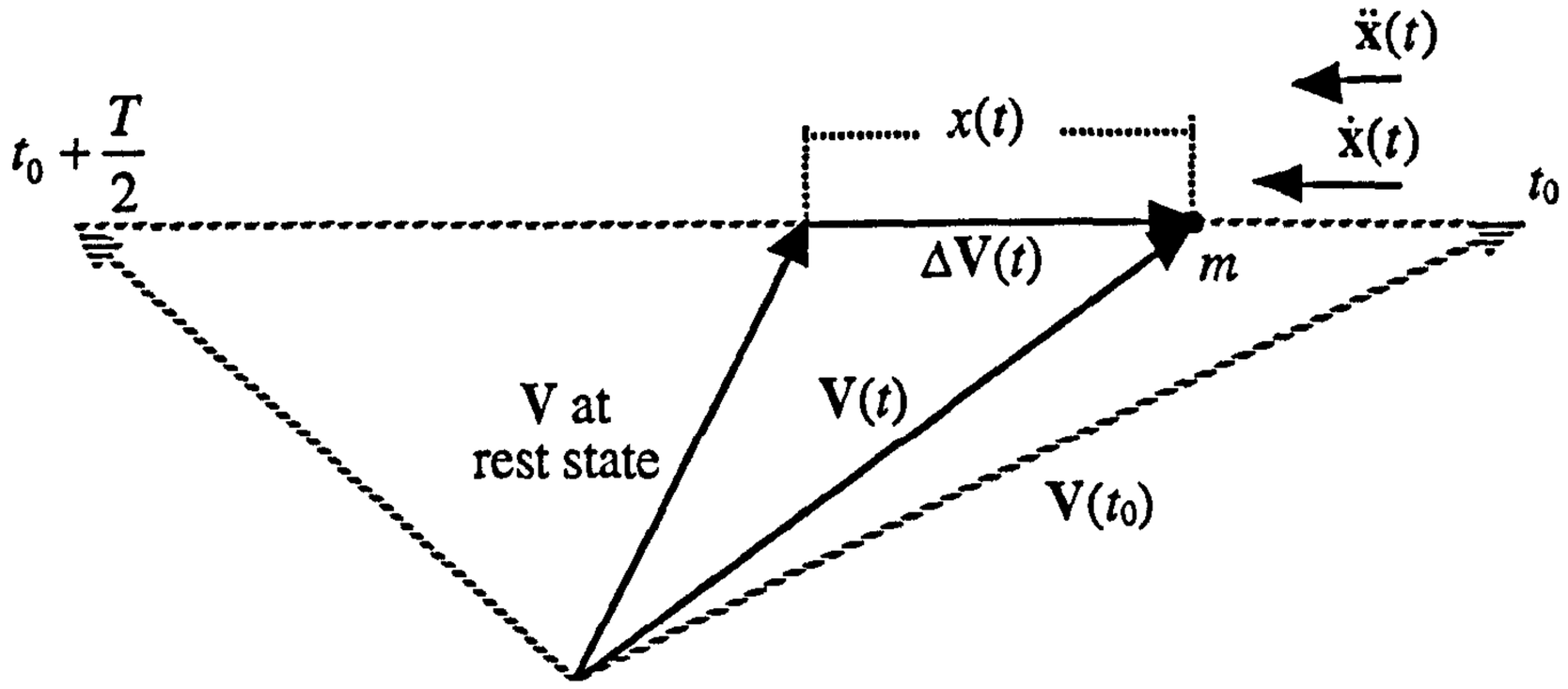


Figure 6.4 Simple harmonic oscillation of a VOO about its rest state.

In an ideal environment (i.e. no friction, no air resistance, etc.) the VOO will be subjected to an undamped harmonic oscillation about its rest state (see figure 6.4). The period of one complete cycle of oscillation is given by:

$$T = 2\pi\sqrt{\frac{m}{s}} \qquad [6.8]$$

6.2.3 Energy of Oscillation

At a time t, the spring-mass system will have stored an amount of elastic (potential) energy which is given by (see section 3.2.3):

$$U_e = \frac{1}{2}sx(t)^2 \qquad [6.9]$$

As it was mentioned in section 6.2.1, the initial length of the spring is equal to 0. The spring-mass system will also have an amount of kinetic energy due to the velocity of its motion which is given by:

$$U_k = \frac{1}{2} m\dot{x}(t)^2 \qquad [6.10]$$

When an oscillation is undamped, the system wastes no work against resistive forces and its total energy remains constant in time. As velocity increases, elastic energy in being converted to kinetic energy and vice versa (see figure 6.5).

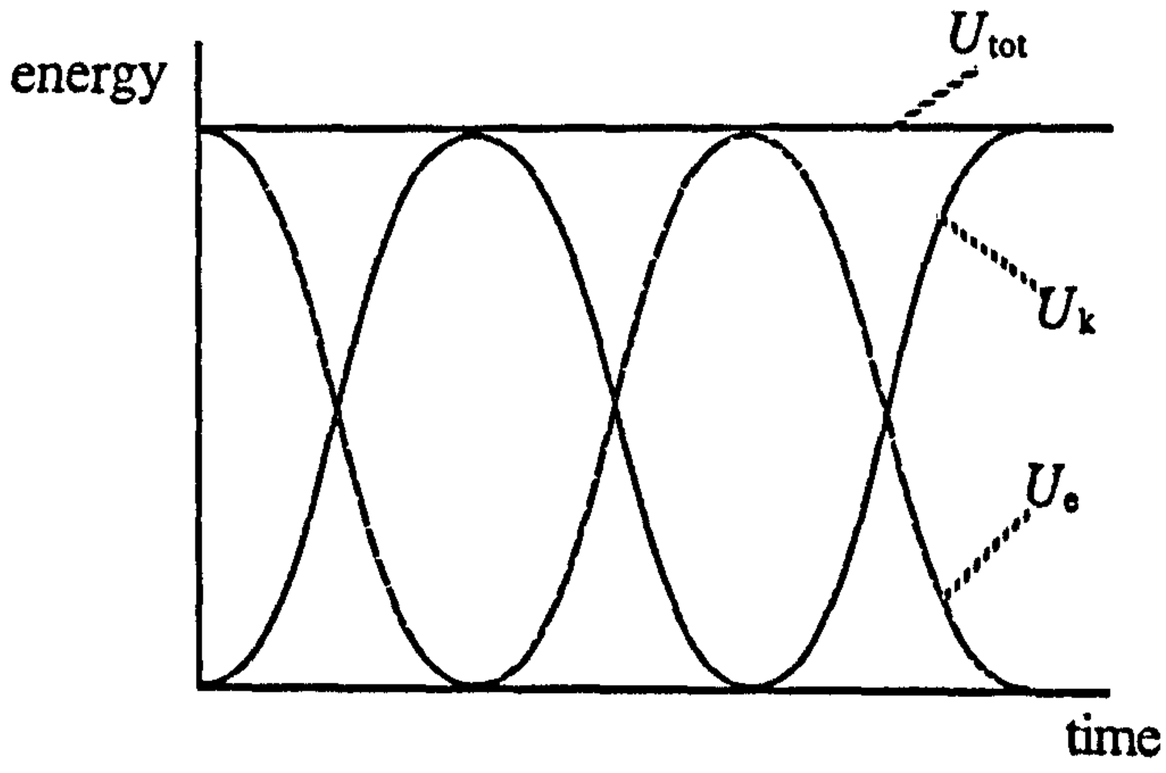


Figure 6.5 Elastic, kinetic and total energy of a spring-mass system.

The total energy of the spring-mass system is given by:

$$U_{tot} = U_e + U_k = \frac{1}{2} sx(t)^2 + \frac{1}{2} m\dot{x}(t)^2 = \text{constant} \qquad [6.11]$$

6.2.4 Damping of Oscillation

A real oscillating system is opposed by dissipative forces, such as air resistance and friction. This results in a proportion of the total energy gradually being removed from the spring-mass system. This process is called damping of the oscillation and its effect is to finally bring the

oscillation to a halt. The reduction of the total energy in a damped oscillation is shown in the graph of figure 6.6.

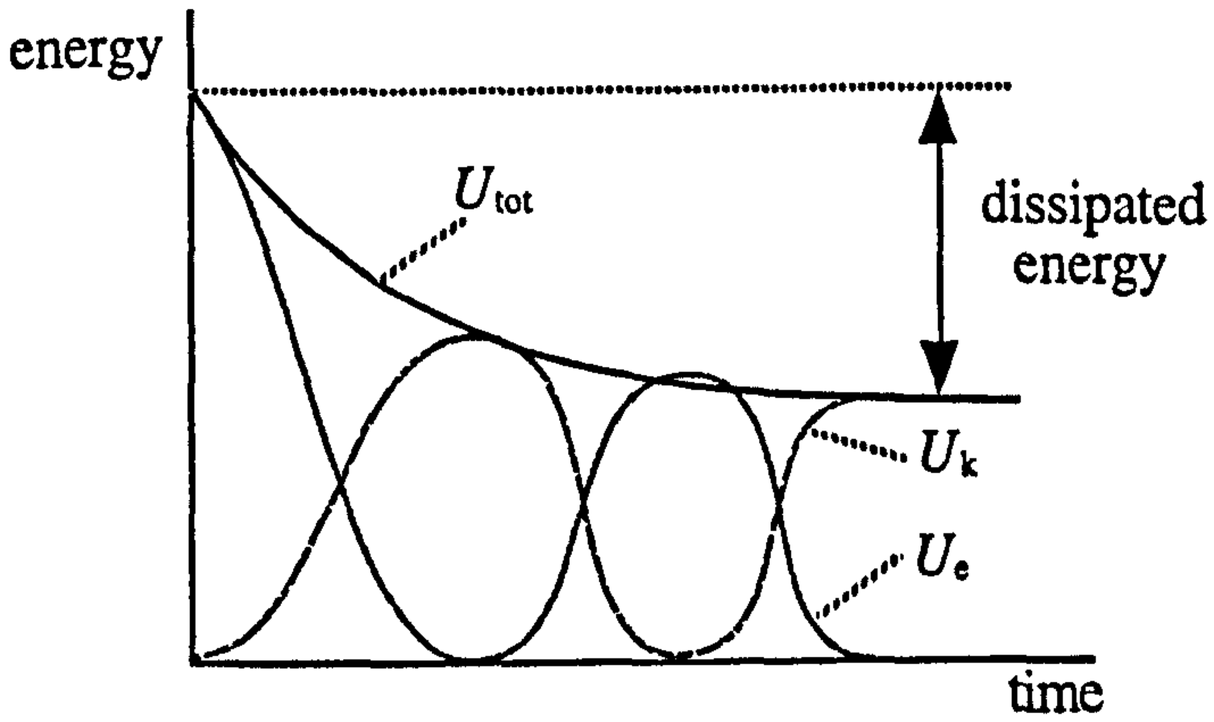


Figure 6.6 Energy / time graph of a damped oscillation.

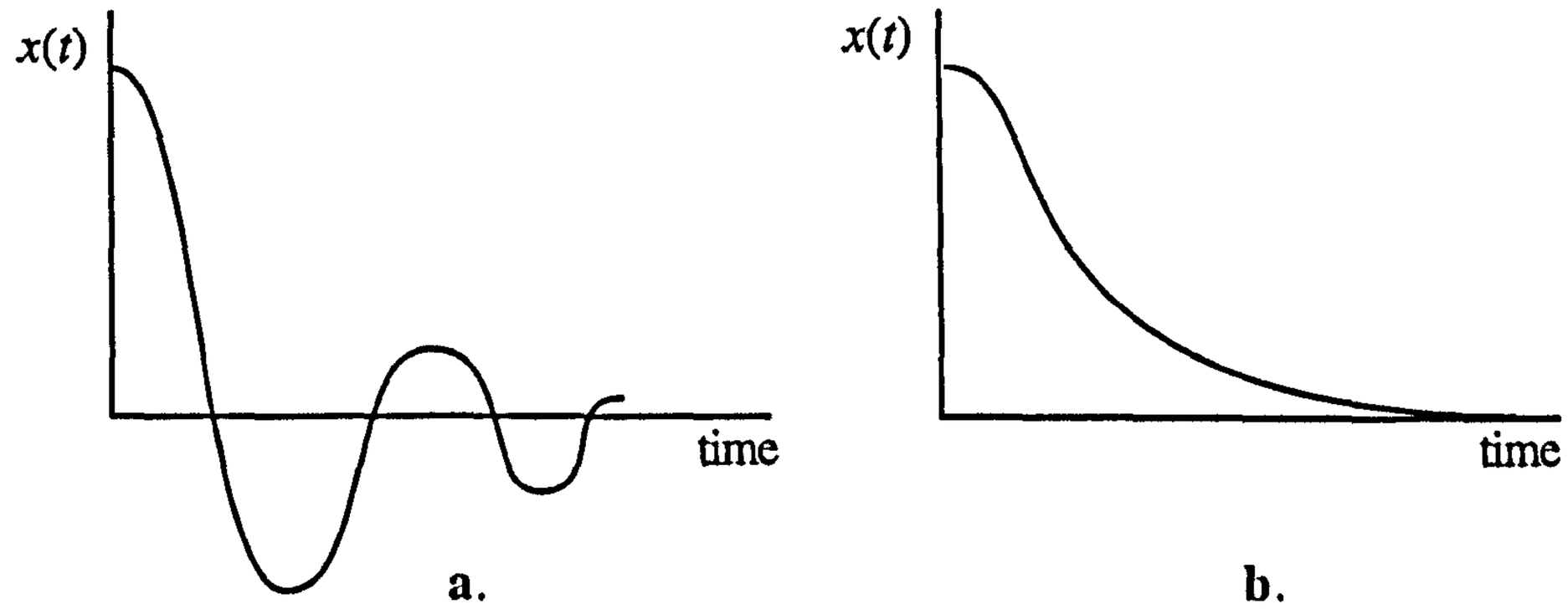


Figure 6.7 Displacement / time graphs of damped oscillations.

An object which is subjected to a damped oscillation may experience several oscillations before it settles back to its undeformed state (see figure 6.7a) or it may return to its original state smoothly without oscillating (see figure 6.7b).

6.2.5 Elastic Restoration of the VOO

In interactive manipulation and animation of elastic deformable objects it is often desirable to constrain the duration of the elastic restoration process. For this purpose, elastic restoration is

usually implemented as a damped oscillation. A mechanism for controlling the time related damping of elastic restoration is introduced in the following sections.

6.2.5.1 The Initial State of the VOO

Returning to the simple case of one deformed VOO (see section 6.2.1), the initial state of the system, at time t_0, will be described.

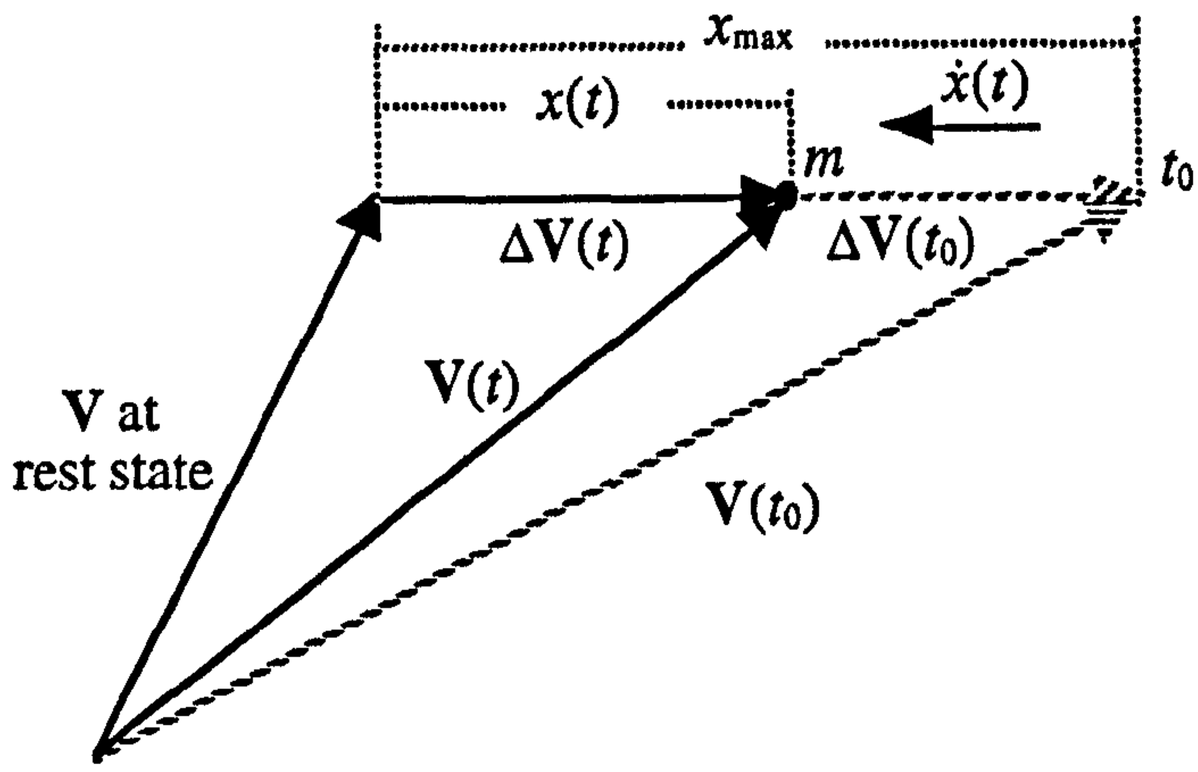


Figure 6.8 Elastic restoration of a VOO.

At time t_0, the external force, $\mathbf{F}_{ext}$, is removed and the elastic restoration process begins. Starting with the two known vectors $\mathbf{V}$ and $\mathbf{V}(t_0)$ their difference $\Delta\mathbf{V}(t_0)$ is calculated as (see figure 6.8):

$$\Delta\mathbf{V}(t_0) = \mathbf{V}(t_0) - \mathbf{V} \qquad [6.12]$$

Vector $\Delta\mathbf{V}(t_0)$ has a maximum length of $x(t_0)$ which is given by:

$$x(t_0) = |\Delta\mathbf{V}(t_0)| = x_{max} \qquad [6.13]$$

Equation [6.13] is used for the calculation of x_{max} which is equal to half the initial amplitude, a, of the oscillation:

$$a = 2x_{max} \qquad [6.14]$$

The velocity of oscillation at time t_0 is equal to zero:

$$\dot{x}(t_0) = 0 \qquad [6.15]$$

The total energy of the system at time t_0, comprises of a maximum amount of elastic energy and no kinetic energy because the velocity is equal to zero. Therefore, the total energy may be calculated as follows:

$$U_{tot} = U_e(t_0) = \frac{1}{2} s x_{max}^2 \qquad [6.16]$$

6.2.5.2 Velocity of Oscillation

Combining equations [6.11] (see section 6.2.3) and [6.16] (see section 6.2.5.1) the following equation may be derived:

$$U_e + U_k = U_{tot} \qquad [6.17]$$

$$\Rightarrow \quad \frac{1}{2} s x(t)^2 + \frac{1}{2} m \dot{x}(t)^2 = \frac{1}{2} s x_{max}^2 \qquad [6.18]$$

$$\Rightarrow \quad \dot{x}(t) = \sqrt{\frac{s}{m}\left(x_{max}^2 - x(t)^2\right)} \qquad [6.19]$$

Equation [6.19] may be used to calculate the absolute value of velocity of the undamped harmonic oscillation as a function of position $x(t)$.

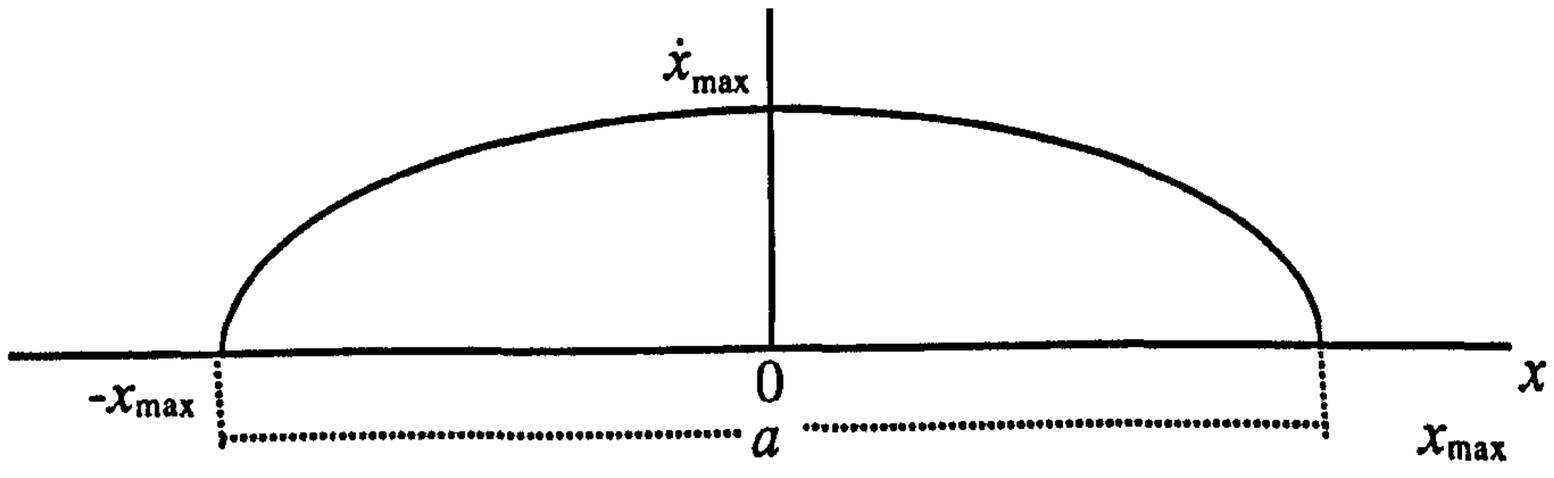


Figure 6.9 Graph of velocity / position in harmonic oscillation.

The graph in figure 6.9 represents the absolute value of velocity as derived from equation [6.19]. When the position is equal to x_{max} or $-x_{max}$ the velocity becomes zero and when the position is equal to zero then the velocity receives its maximum value:

$$\dot{x}_{max} = x_{max}\sqrt{\frac{s}{m}} \qquad [6.20]$$

6.2.5.3 Energy Damping

In order to damp the oscillation process and achieve complete elastic restoration of the deformation an energy damping function, $\zeta(t)$, is introduced. A typical graph of an energy damping function is shown in figure 6.10.

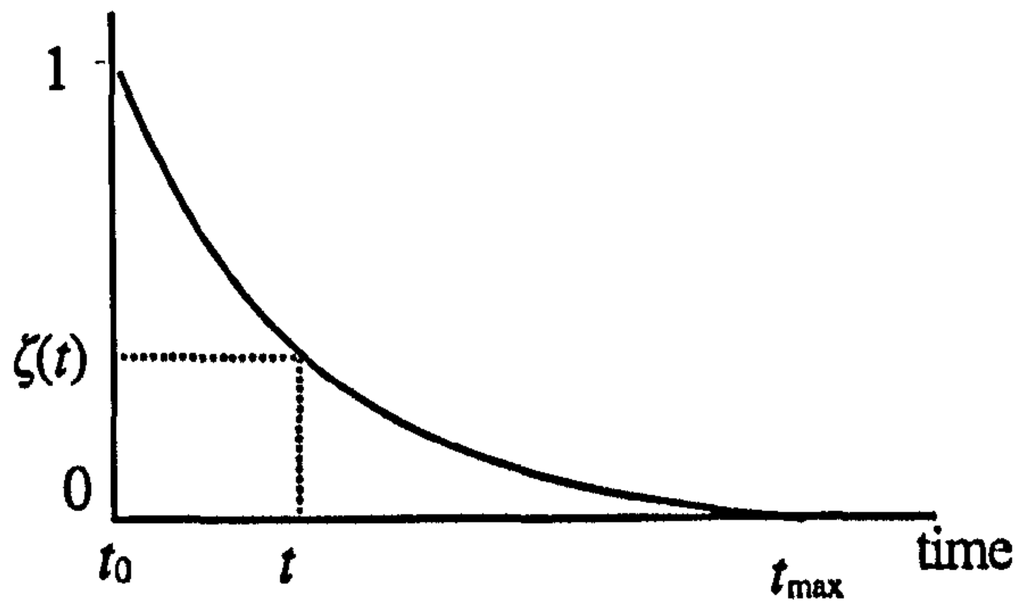


Figure 6.10 Energy damping function.

An energy damping function describes how the energy of a spring-mass system is dissipated over time. The function $\zeta(t)$ returns the value of 1 at time t_0 and the value of 0 at a time t_{max}. The

shape of the graph may differ from the one in figure 6.10 but the initial and final conditions are fixed. The user has control over the shape of the graph, as well as the value of the maximum time t_{max}. The parameter t_{max} determines the duration of the elastic restoration process: (t_{max} - t_0). The relationship between the duration (t_{max} - t_0) and the period, T, of the oscillation (see equation [6.8]) determines whether the elastic restoration process will result in a complete restoration of the vector $\mathbf{V}(t)$. For complete restoration, where $\mathbf{V}(t_{max}) = \mathbf{V}$, the following rule must be obeyed (see figure 6.11):

$$t_{max} - t_0 = n\frac{T}{4} \qquad \text{where} \quad n \bmod 2 = 1 \qquad \textbf{[6.21]}$$

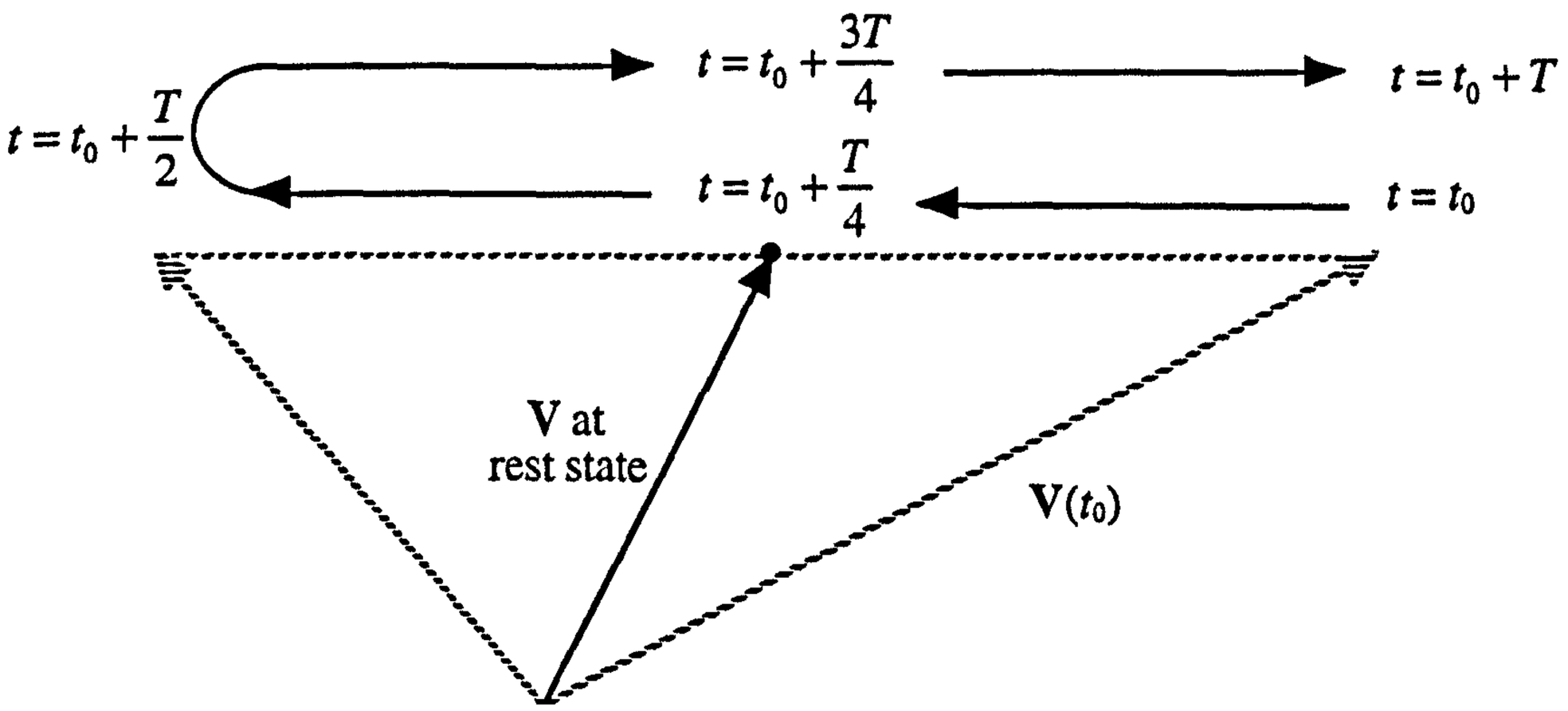


Figure 6.11 The period of oscillation.

A damping function, $\zeta(t)$, may be used to damp the total energy of a spring-mass system or just the kinetic or elastic energy element.

6.2.5.4 Damping of the Total Energy

Equation [6.17] may be extended to incorporate damping by multiplying the total energy element with a damping function as follows:

$$U_k + U_e = U_{tot} \cdot \zeta(t) \qquad [6.22]$$

$$\Rightarrow \quad \dot{x}(t) = \sqrt{\frac{s}{m}\left(x_{max}^2 \cdot \zeta(t) - x(t)^2\right)} \qquad [6.23]$$

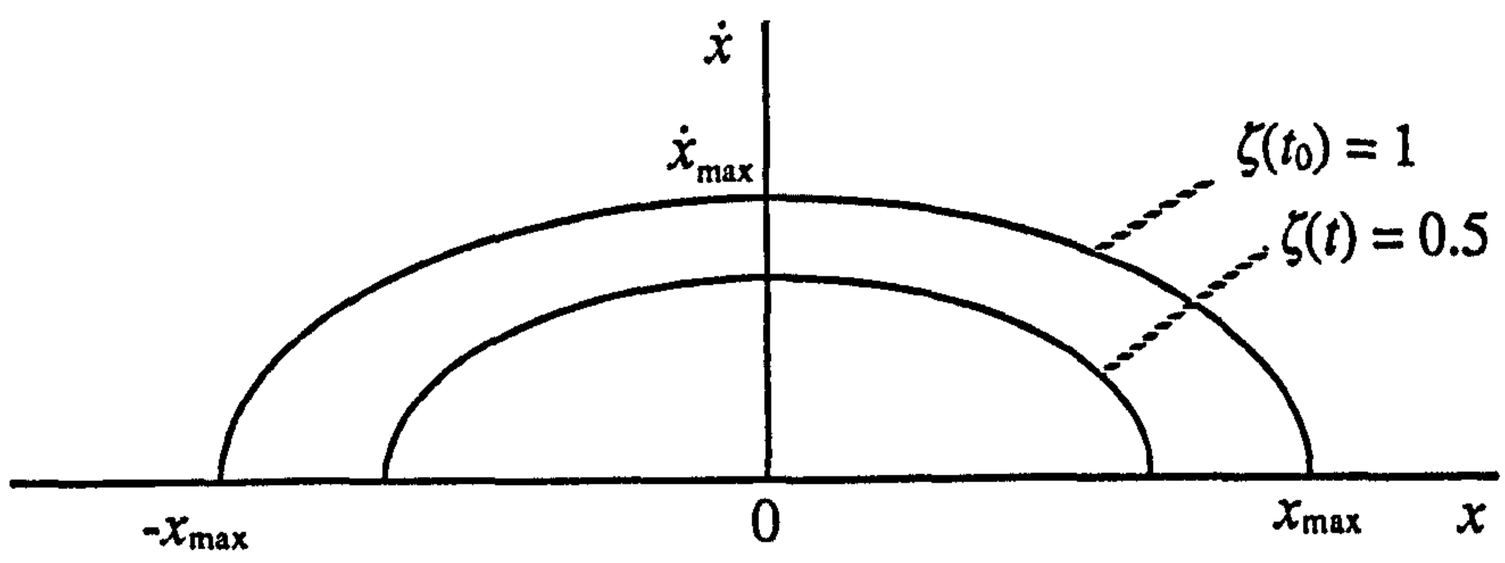


Figure 6.12 Graph of velocity / position in an elastic restoration process, where the total energy is being damped.

Equation [6.23] may be used to calculate the absolute value of the velocity of the damped elastic restoration as a function of the position $x(t)$. Damping the total energy of an oscillation causes a simultaneous reduction of amplitude and velocity (see figure 6.12).

6.2.5.5 Damping of the Elastic Energy

Alternatively, equation [6.17] may be extended to incorporate damping by multiplying the elastic energy element with a damping function as follows:

$$U_e = \left(U_{tot} - U_k\right) \cdot \zeta(t) \qquad [6.24]$$

$$\Rightarrow \quad \dot{x}(t) = \sqrt{\frac{s}{m}\left(x_{max}^2 - \frac{1}{\zeta(t)} x(t)^2\right)} \quad \text{where } \zeta(t) \neq 0 \qquad [6.25]$$

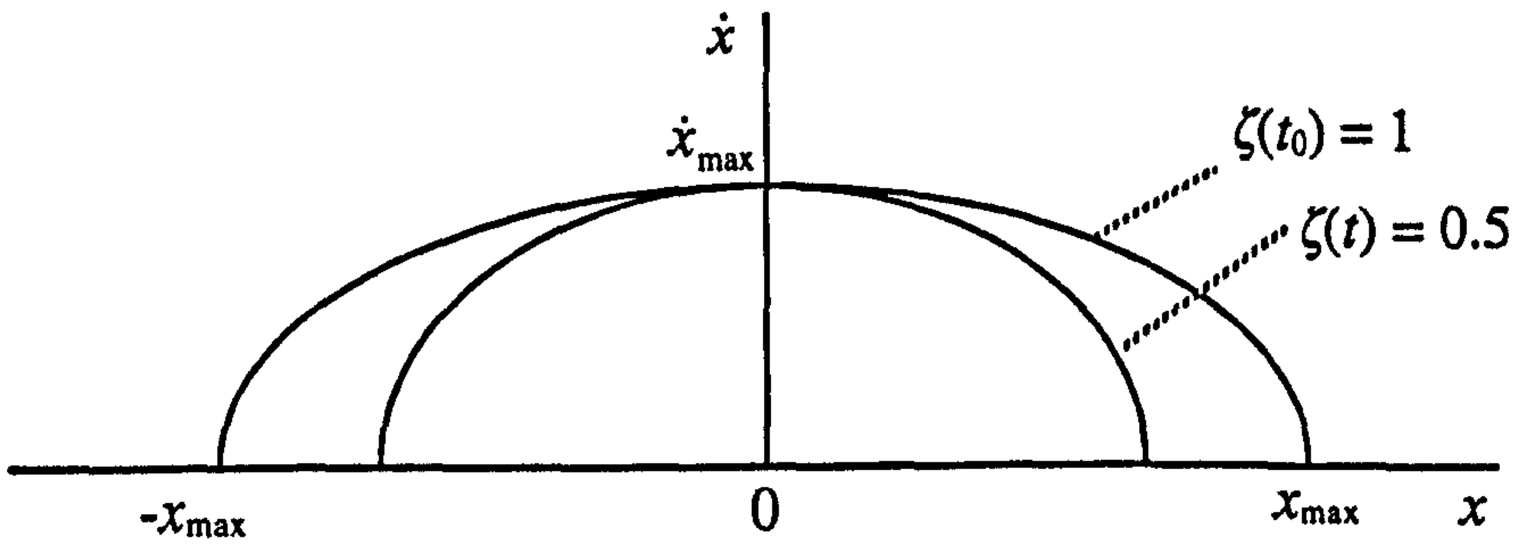


Figure 6.13 Graph of velocity / position in an elastic restoration process, where the elastic energy is being damped.

Damping the elastic energy of an oscillation causes a reduction of amplitude while the maximum velocity remains unaffected (see figure 6.13).

6.2.5.6 Damping of the Kinetic Energy

Finally, damping may also be applied to the kinetic energy element as follows:

$$U_k = (U_{tot} - U_e) \cdot \zeta(t) \qquad [6.26]$$

$$\Rightarrow \quad \dot{x}(t) = \sqrt{\frac{s}{m}\left(x_{max}^2 - x(t)^2\right) \cdot \zeta(t)} \qquad [6.27]$$

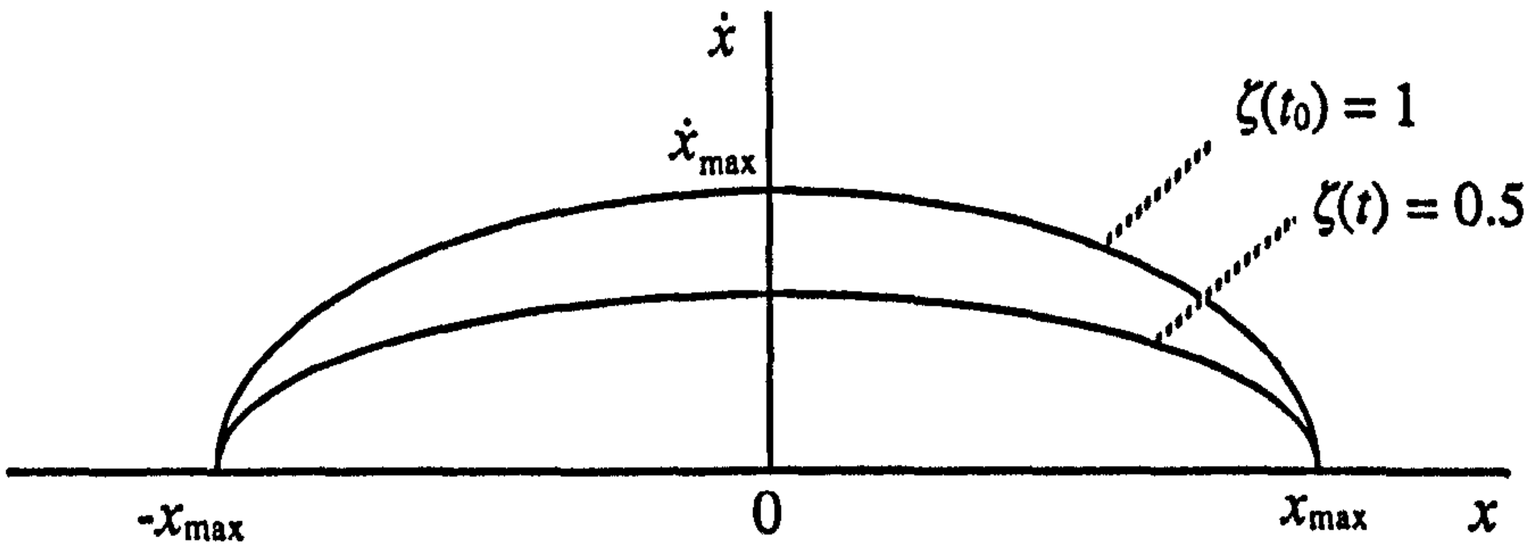


Figure 6.14 Graph of velocity / position in an elastic restoration process, where the kinetic energy is being damped.

Damping the kinetic energy of an oscillation causes a reduction of the absolute value of the velocity while the amplitude of the oscillation remains constant (see figure 6.14).

The user may choose to apply energy damping to either of the total, the elastic or the kinetic energy. The model will then deploy either of equations [6.23], [6.25] or [6.27] respectively, for the calculation of the absolute value of the velocity.

6.2.5.7 Direction of the Velocity Vector

Equations [6.23], [6.25] and [6.27] produce an absolute value of the velocity of restoration at time *t*. The direction of the velocity vector must be reversed every time a local minimum of the velocity graph is reached (see figure 6.15).

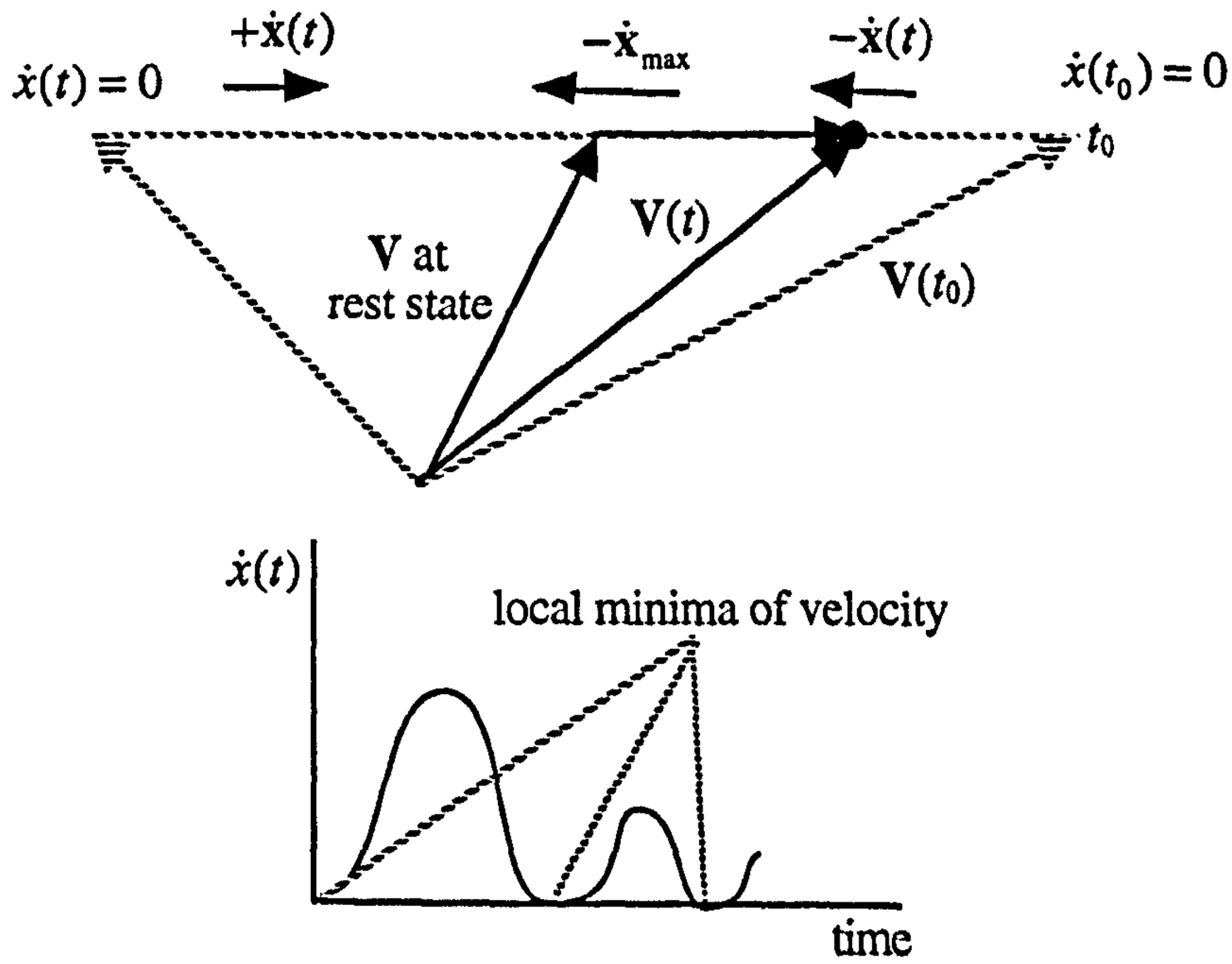


Figure 6.15 The velocity direction is reversed at the local minima of the velocity graph.

Depending on which of the equations [6.23], [6.25] or [6.27] is used for the calculation of velocity, the minimum velocity may be detected as follows:

If equation [6.23] is used:

$$x(t)^2 = x_{\max}^2 \zeta(t) \quad \Rightarrow \quad \dot{x}(t) = 0 \qquad [6.28]$$

If equation [6.25] is used:

$$\frac{1}{\zeta(t)} x(t)^2 = x_{\max}^2 \quad \Rightarrow \quad \dot{x}(t) = 0 \qquad [6.29]$$

And finally, if equation [6.27] is used:

$$x(t)^2 = x_{\max}^2 \quad \Rightarrow \quad \dot{x}(t) = 0 \qquad [6.30]$$

6.2.5.8 The State of a VOO During Elastic Restoration

The length $x(t)$ at time t (i.e. the magnitude of the difference vector $\Delta \mathbf{V}(t)$) may be calculated as follows:

$$x(t) = x(t_0) \, \mathrm{m} \int_{t_0}^{t} \dot{x}(t) dt \qquad [6.31]$$

The sign is initially set to negative and then its is reversed each time the velocity reaches a local minimum (see section 6.2.5.7). The integration of equation [6.31] from time t_0 to a time t, may be performed using the following pseudocode:

$time = t_0 + dt$ (dt is the integration time step)
$x = x_{max} - dx$ (to kick-start the restoration process)
$\zeta(time)$ = read value from energy graph

$$\dot{x} = \sqrt{\frac{s}{m}\left(x_{max}^2 \zeta(time) - x^2\right)}$$

(assume total energy damping)
sign = -1 (start with velocity opposite to $\Delta V(t_0)$)
repeat

$$x = x + \text{sign} \cdot \dot{x}$$

$time = time + dt$
$\zeta(time)$ = read value from energy graph

$$\dot{x} = \sqrt{\frac{s}{m}\left(x_{max}^2 \zeta(time) - x^2\right)}$$

if $\dot{x} \approx 0$ then sign = -sign (change direction of velocity vector)
until $time \geq t$ **[6.32]**

The integration produces a signed value for $x(t)$. The absolute value $|x(t_0)|$ represents the magnitude of vector $\Delta V(t)$. The direction of vector $\Delta V(t)$ is the same or opposite to the direction of $\Delta V(t_0)$, depending on the sign of $x(t)$. At the beginning, the elastic restoration process is being kick-started by subtracting a small amount dx, from x_{max}. Otherwise, the velocity remains equal to 0. The integration step dt, is usually equal to 1 frame of animation (i.e. 1/25th of a second). By increasing the step to 2 or more frames, the elastic restoration process becomes more efficient, though, it may easily result to unstable motion. In the course of the integration, every time the velocity reaches values near zero, the direction of the velocity vector is reversed. Finally, the restored state of the VOO at time t, is calculated as:

$$V(t) = V + \Delta V(t) \qquad \textbf{[6.33]}$$

Having calculated the state of a VOO at a time t, during an elastic restoration process the local deformation model is then activated in order to restore the deformation on the objects that are influenced by the VOO.

6.2.5.9 The Final State of the VOO

The integration process may be executed until time t_{max}, when the damping function $\zeta(t)$ reaches

the value of zero. If the value of t_{max} satisfies equation [6.21] (see section 6.2.5.3) then there will be a complete restoration of the VOO and its final state will be:

$$\mathbf{V}(t_{max}) = \mathbf{V} \qquad \dot{x}(t_{max}) = 0 \qquad x(t_{max}) = 0 \qquad \textbf{[6.34]}$$

6.3 The Elastoplastic Model

In the previous section a VOO was equipped with a perfectly elastic spring which tended to completely restore any amount of strain applied to the VOO itself and to the influenced objects. This mechanism may be used to simulate perfectly elastic objects which have a linear stress / strain relationship (see figure 6.16a). Observe that the slope of the graph represents the elasticity coefficient *s* (i.e. spring constant).

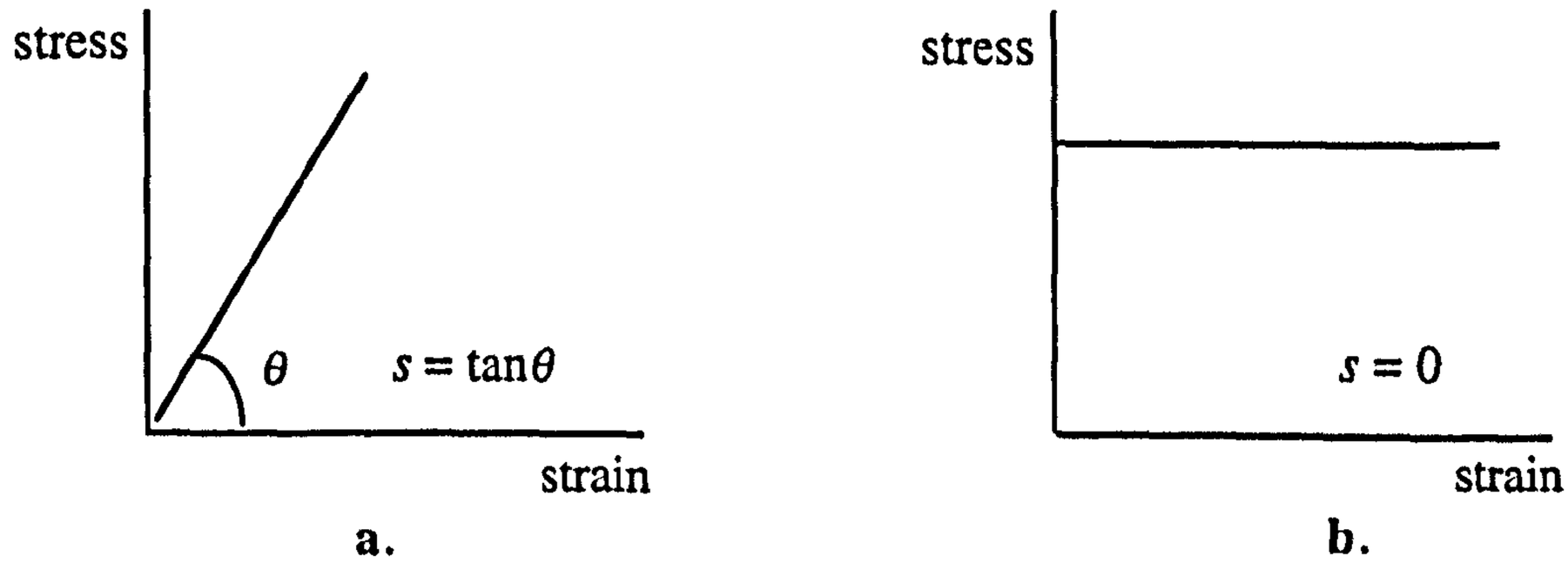


Figure 6.16 a. Perfectly elastic material, b. perfectly plastic material.

On the other hand, a perfectly plastic object may be subjected to any amount of strain under the influence of the same stress (see figure 6.16b). The stress / strain graph is a line parallel to the strain axis and the elasticity coefficient, *s*, is equal to zero. Therefore, there is no elastic energy stored inside a plastic object and no elastic restoration after removal of the external stress.

Most natural materials, however, are neither perfectly elastic nor perfectly plastic. They exhibit elastic behaviour under small stresses and plastic behaviour when stresses exceed a certain limit. The stress / strain relationship of such materials describes their complex mechanical

behaviour (see figure 6.17).

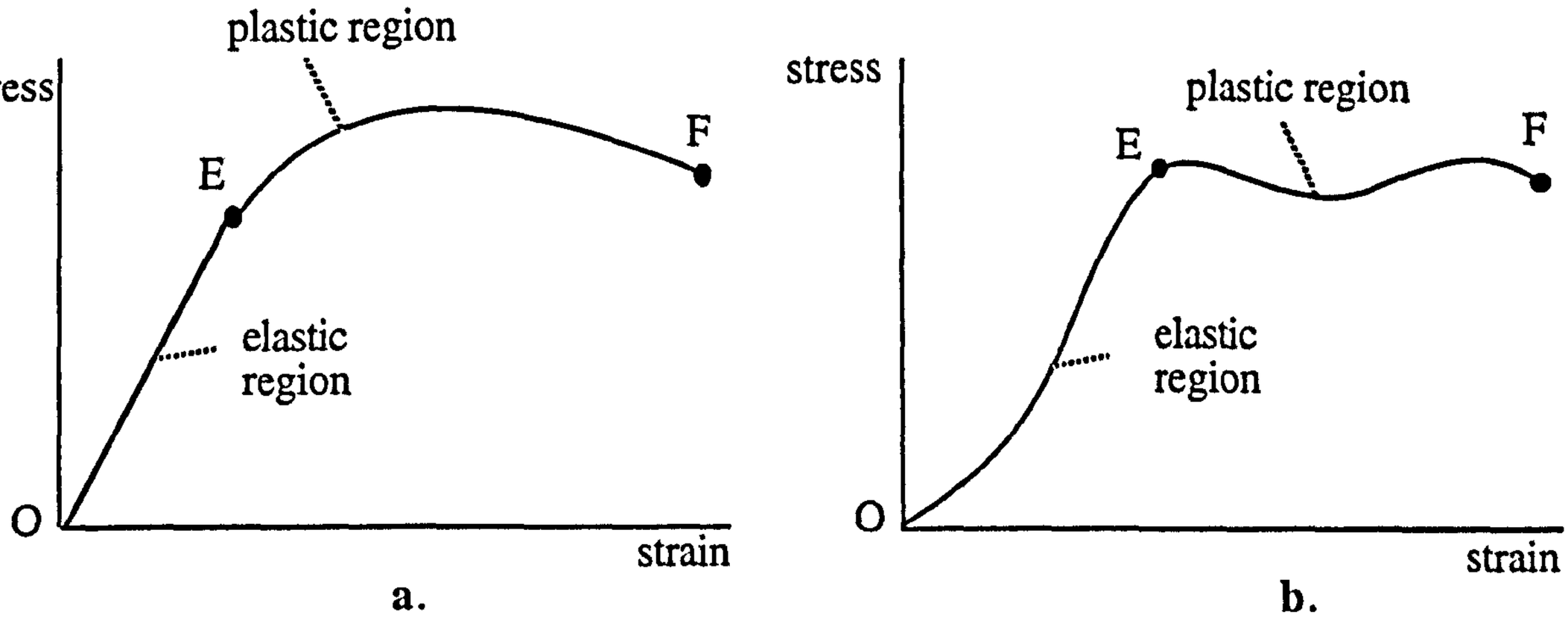


Figure 6.17 Simplified stress / strain relationship in natural materials.
a. metal, b. organic material.

The relationship between stress and strain inside the elastic region of such objects may or may not be linear. Most metals exhibit approximately linear elasticity (see figure 6.17a). Most organic materials, however, exhibit non-linear elasticity (see figure 6.17b) (see also figures 3.13, 3.16 and 3.17 in section 3.5). In fact, linear elasticity is an idealised material behaviour and it is rarely encountered in real life materials.

6.3.1 The Elastic Region

For stresses and strains that fall inside the elastic region, OE, of a stress / strain graph (see figure 6.18) the object is subjected to complete elastic restoration after removal of the external stress. The elasticity coefficient, $s(e)$, that corresponds to a point $P = (\sigma, e)$, may be derived as the tangent of the graph at point P (see figure 6.18). This elasticity coefficient, $s(e)$, is then kept constant during the elastic restoration process.

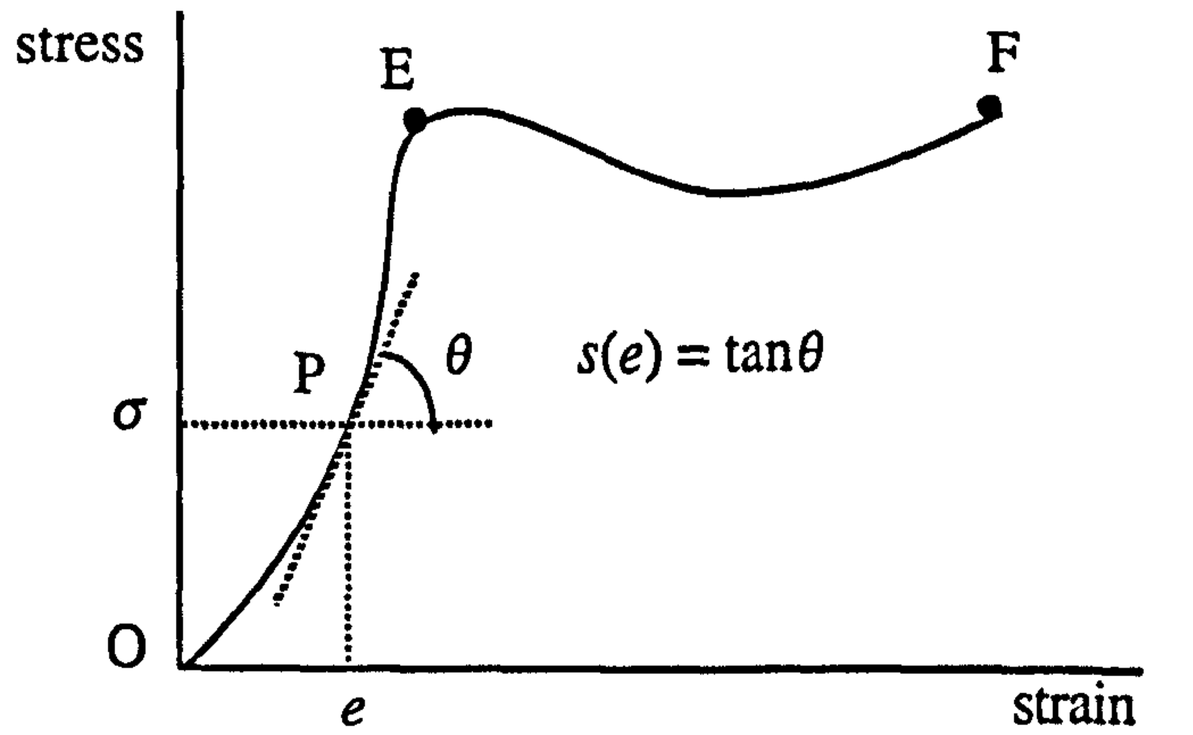


Figure 6.18 Elasticity coefficient, $s(e)$, inside the elastic region.

The slope of the graph inside the elastic region must always be greater than zero. If the slope of the graph does become zero, then $s(e) = 0$, which would cause the velocity of oscillation to become zero (see equation [6.19]) and the period of oscillation to become infinite (see equation [6.8]). In other words, the deformation would become perfectly plastic (see figure 6.16b). Also the slope of the graph inside the elastic region should never become equal to 90° because this would produce an infinite elasticity coefficient and a period of oscillation equal to zero.

6.3.2 The Plastic Region

In figure 6.19, point E represents the elastic limit and point F the fracture limit. For stresses and strains that fall inside the plastic region, EF, of the graph the object is subjected to partial elastic restoration which leaves an amount of permanent plastic strain on the object (see figure 6.19).

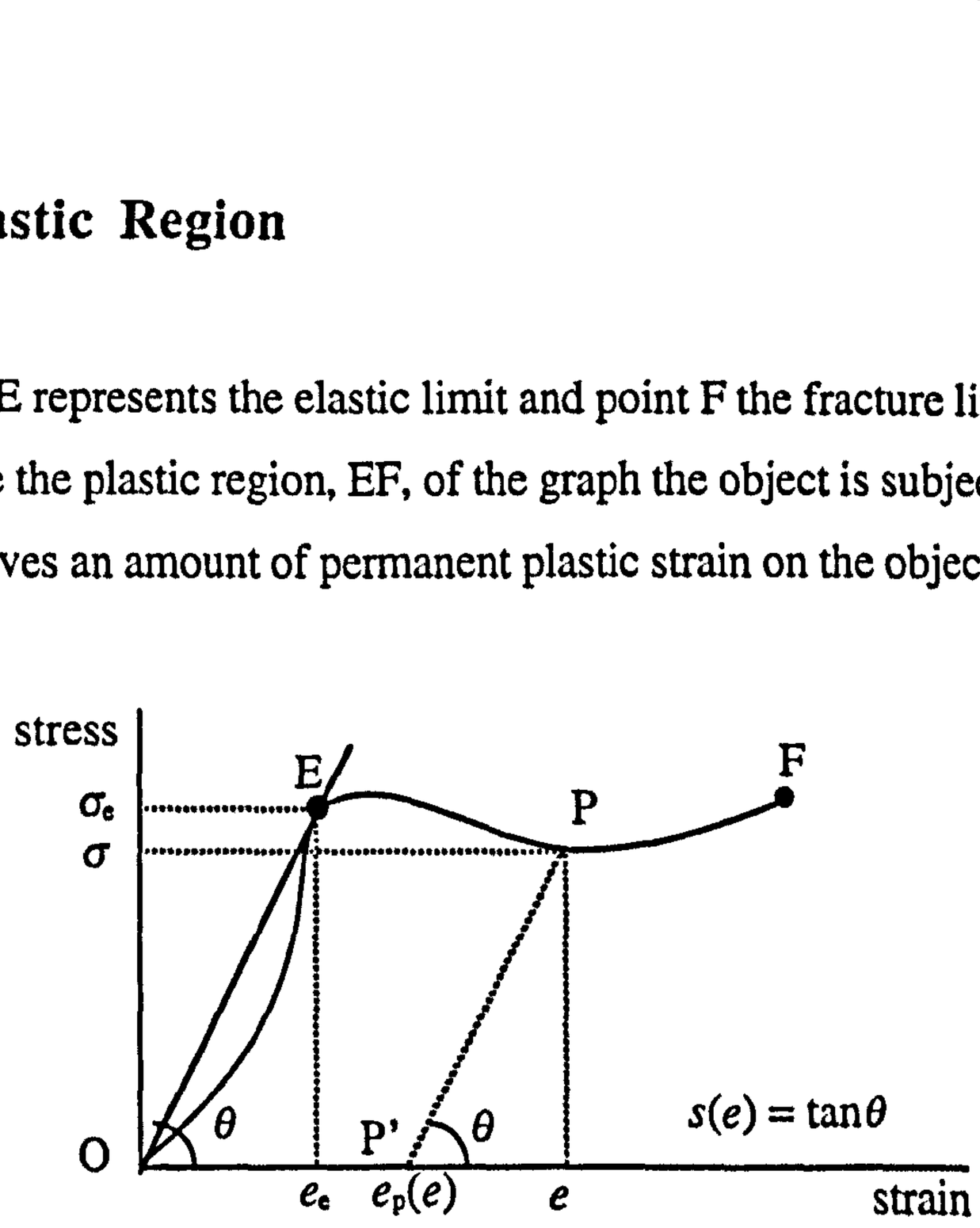


Figure 6.19 Partial elastic restoration with permanent strain in plastic region.

A point P = (σ, e) that lies inside the plastic region corresponds to a stress σ that will cause an initial strain e. Partial elastic restoration will take place with an elasticity coefficient, $s(e)$, and it will result in a permanent plastic strain, $e_p(e)$. An approximation of the values for $s(e)$ and $e_p(e)$ may be obtained by assuming that line PP′ is parallel to line OE (see figure 6.19). The elasticity coefficient, $s(e)$, is assumed to be constant during the elastic restoration and equal to the slope of the line PP′ (or of line OE), thus:

$$s(e) = \frac{\sigma}{\left(e - e_p(e)\right)} = \frac{\sigma_e}{e_e} \qquad [6.35]$$

where σ_e and e_e are the stress and strain that correspond to the elastic limit point E. Equation [6.35] may then be solved for $e_p(e)$ as follows:

$$e_p(e) = e - \frac{\sigma}{s(e)} \qquad [6.36]$$

In such a case, elastic restoration will proceed until the strain reaches the value of $e_p(e)$.

6.3.3 Implementation of the Elastoplastic Model

The elastic model which was introduced in section 6.2 may be extended to simulate a mixture of the properties of elasticity and plasticity of an object. To this end, a user defined stress / strain graph provides the information needed for the implementation of complex elastoplastic material behaviour of an object.

In a typical stress / strain graph (see figure 6.20), the two points that correspond to the elastic and fracture limits E and F respectively, may be defined by the pairs (σ_e, e_e) and (σ_f, e_f). For strains between 0 and e_e, the object experiences complete elastic restoration. The graph is only used for the calculation of the elasticity coefficient, $s(e)$, as a function of strain (see section 6.3.1).

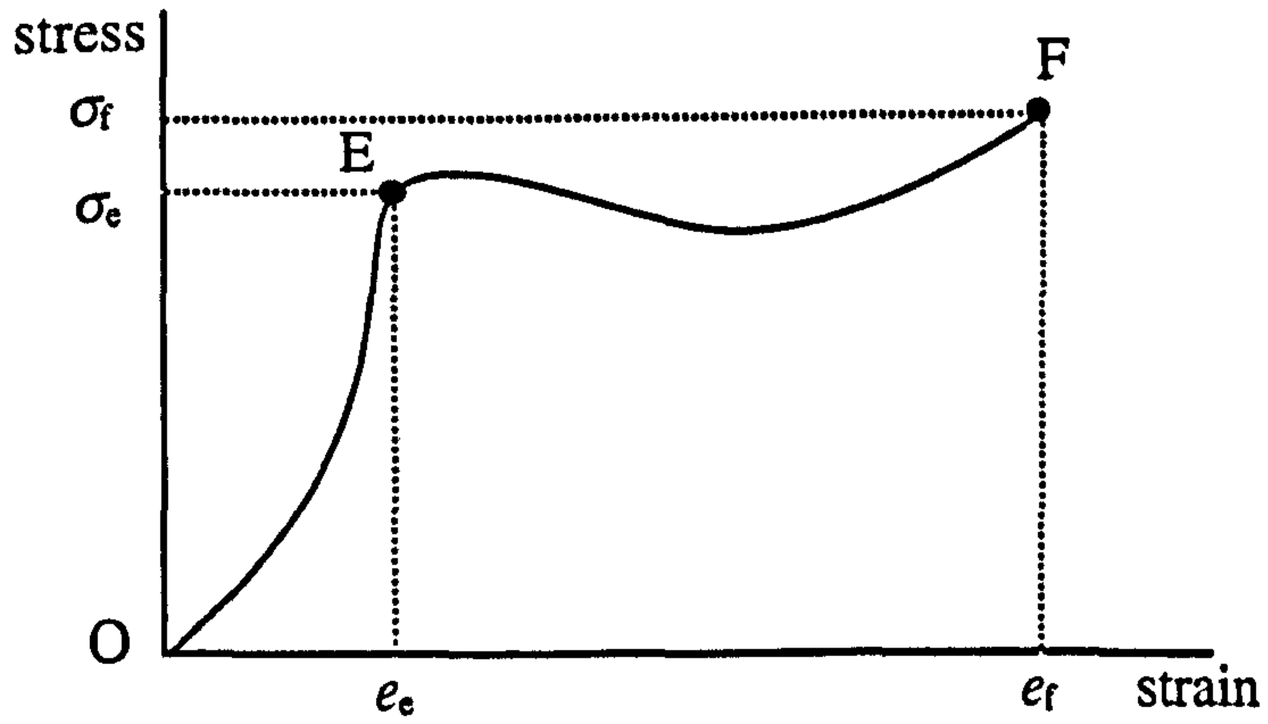


Figure 6.20 A stress / strain graph with elastic and fracture strain limits.

$$0 \le e \le e_e \quad \Rightarrow \quad \text{complete elastic restoration} \qquad \textbf{[6.37]}$$

Whereas, for strains between e_e and e_f, there is a partial elastic restoration resulting in a portion of permanent strain. The graph is used for the calculation of the permanent strain, $e_p(e)$ and the elasticity coefficient, $s(e)$ (see section 6.3.2).

$$e_e < e < e_f \quad \Rightarrow \quad \text{partial elastic restoration with permanent strain} \qquad \textbf{[6.38]}$$

Although, fracture of objects is not considered by the suggested model, the fracture limit, e_f, is simply used as a maximum permitted strain.

$$e \ge e_f \quad \Rightarrow \quad \text{no elastic restoration (fracture)} \qquad \textbf{[6.39]}$$

A stress / strain graph is not used for the derivation of absolute stress values. So, the stress axis of the graph is only there to provide relative information for the calculation of the elasticity coefficient ,$s(e)$ and the permanent strain, $e_p(e)$.

In the presentation of the elastic model (see section 6.2), it was mentioned that the process of elastic restoration aims to diminish the difference vector $\Delta\mathbf{V}(t)$ between $\mathbf{V}(t)$ and $\mathbf{V}$. In the elastoplastic model, the strain e is assumed to be equal to the initial length x_{max} of vector $\Delta\mathbf{V}(t_0)$. This is done to simplify the calculations and improve the efficiency of the model.

The pseudocode [6.32] (see section 6.2.5.8) may be extended for the elastoplastic model. In the following pseudocode it is assumed that the user has opted for total energy damping.

$e = x_{max}$
if $0 \le e \le e_e$ (inside elastic region)
$s(e)$ = read tangent from stress / strain graph
$time = t_0 + dt$
$\zeta(time)$ = read value from energy graph
$x = x_{max} - dx$

$$\dot{x} = \sqrt{\frac{s(e)}{m}\left(x_{max}^2 \zeta(time) - x^2\right)}$$

sign = -1
repeat
$x = x + \text{sign} \cdot \dot{x}$
$time = time + dt$
$\zeta(time)$ = read value from energy graph

$$\dot{x} = \sqrt{\frac{s(e)}{m}\left(x_{max}^2 \zeta(time) - x^2\right)}$$

if $\dot{x} \approx 0$ then sign = -sign
until $time \ge t$
else
if $e_e < e < e_f$ (inside plastic region)

$$s(e) = \frac{\sigma_e}{e_e}$$

$$e_p(e) = e - \frac{\sigma}{s(e)}$$

$time = t_0 + dt$
$x = x_{max} - dx$
$\zeta(time)$ = read value from energy graph

$$\dot{x} = \sqrt{\frac{s(e)}{m}\left((x_{max} - e_p(e))^2 \cdot \zeta(time) - (x - e_p(e))^2\right)}$$

sign = -1
repeat
$x = x + \text{sign} \cdot \dot{x}$
$time = time + dt$
$\zeta(time)$ = read value from energy graph

$$\dot{x} = \sqrt{\frac{s(e)}{m}\left((x_{max} - e_p(e))^2 \cdot \zeta(time) - (x - e_p(e))^2\right)}$$

if $\dot{x} \approx 0$ then sign = -sign
until $time \ge t$ **[6.40]**

The elasticity coefficient, $s(e)$ and the permanent strain, $e_p(e)$ are only calculated once in the

beginning of each integration process. The rest of the calculations inside the integration loop are similar to the elastic model (see pseudocode [6.32]). The integration method in pseudocode [6.32] and in [6.40] is likely to result in numerical instability. The current solution is good enough for demonstration purposes but for a commercial implementation a more robust method could be implemented (see Taylor and Euler methods (Press *et al.* 1986)).

6.4 Results

6.4.1 Effectiveness of the Elastoplastic Model

In this chapter, the suggested deformation model has been extended to incorporate the physical properties of elasticity and plasticity. The main objective of this chapter was to achieve realistic elastoplastic organic material behaviour using an efficient and constrainable model.

An important feature in the mechanical behaviour of organic materials is their elastoplastic response to external stresses. In chapter 3, it was shown that organic materials such as animal skin and flesh do not exhibit perfectly elastic neither perfectly plastic behaviour. These materials exhibit mechanical behaviours that may be described by complex stress / strain graphs (see figures 3.13, 3.16 and 3.17 in section 3.5). These stress / strain graphs differ considerably from that of an idealised linearly elastic model (see figure 6.16).

The suggested elastoplastic model is based on a user defined stress / strain graph. This enables the customisation of the material behaviour of a CG model according to data derived from real-life experiments. Alternatively, the user may experiment by modifying the shape of a stress / strain graph and visually customise the elastoplastic behaviour of an object. The user may also apply time constraints to the elastic restoration process by defining an energy damping graph and by directly controlling the duration of the process. Other user controlled parameters are the property of mass and the elastic and fracture strain limits of the material. The stress / strain graph, the energy damping graph, the duration of elastic restoration, the mass and the elastic and fracture strain limits of a material may or may not be related to the respective properties of

real materials. Manipulation of these graphs and parameters provides a mechanism which is capable of generating a whole range of natural and artificial elastoplastic material behaviours.

Several researchers have suggested elastically deformable models based on local deformation and linear elasticity. A characteristic example is a fast elastically deformable model introduced by Witkin and Welch (1990). Their method produces puppets whose parts are made of jelly like, elastic material with specified points under the full control of the animator. The puppet moves with correct passive dynamics by moving these control points as functions of time. The advantage of their approach is the great efficiency of their model. However, the realism of the movement is restricted by the material behaviour of synthetic elastic objects. Terzopoulos *et al.* (1987) also suggested a model of elastically deformable objects based on elasticity theory. Their model can be made to behave like string, rubber, cloth, paper, flexible metal, etc. They also incorporate physical properties such as mass and damping. In their method the shape of a body is determined by the Euclidian distances between nearby mass point elements. As the body is forced to deform these distances change. Assuming that all the mass points of an object are interconnected with elastic springs any shape deformation would result in an amount of elastic potential energy stored inside the object. This energy of elasticity generates internal forces that constantly attempt to restore each point to its rest state. Their model is capable of realistic simulation of elastic material behaviour. However, their model is not efficient because its solution results in a large number of simultaneous differential equations of motion, one for each mass point. This leads to a substantial computational overhead and prohibits interactive manipulation of their model.

An advantage of the suggested elastoplastic model is the freedom offered to a user / animator to design the elastoplastic material behaviour through stress / strain and energy damping graphs. This enables the user to experiment visually in order to achieve the desired deformable behaviour. The user may also approximate real life organic material behaviour using experimentally derived graphs and thus improve the realism in deformable motion.

Another advantage of the suggested model is the use of VOOs as an intermediate layer between the elastoplastic model and the actual deformation model. This allows the application of the elastoplastic model directly on VOOs. An elastically restored VOO transfers the deformation

inside its field of influence and onto the influenced objects. This two-layer approach helps separate the deformation problem from the elastoplastic behaviour problem and it greatly improves the efficiency of the overall model.

Colour plates relevant to this chapter can be found in section 6.6. Also, see animation tests 4, 5, 6 and 7 in video tape, time code: from 10:04:46:00 to 10:05:38:00.

6.4.2 Performance of the Elastoplastic Model

The efficiency of the proposed method is based on the fact that the elastic restoration process is only applied to the time related state of a VOO. So, the process does not attempt to restore individual vertices. A further element which improves the efficiency of the model is the use of a simplified method for deriving the elasticity coefficient and the permanent plastic strain from a stress / strain graph. Also, a simple integration from time t_0 to current time t, is used for the calculation of the restored state of vector $\mathbf{V}(t)$ of a VOO. The local deformation model is then activated to restore the influenced objects (see chapter 5).

The following performance tests have been carried out to measure the speed of execution of the elastoplastic model.

6.4.2.1 Performance Test 1

The basic performance test (see section 5.9.2) was modified as follows:

The tip of the VOO was offset to a new position inside the plastic region of the stress / strain graph (the plastic region of a stress / strain graph represents the case where the most calculations are involved). The duration of the elastic restoration process was set to 25 frame steps. At a time t_0, the elastic restoration process was triggered while the process was being timed. The speed of the elastic restoration was derived in frames per second. This speed

represents the amount of time per frame step, used for calculations by the elastoplastic model, including the simple deformation model and the drawing routines. The test was repeated by adding extra copies of the same object inside the influence field of the VOO. The same elastic restoration process was performed and the speed of the process was measured each time. The deformation was performed with precalculation of damping elements. The results are shown in the graph in figure 6.21 (see blue line). For comparison, the speed graph of a simple plastic deformation test is superimposed (see figure 6.21 red line). The computation overhead created by the elastoplastic model is due to calculations executed once per VOO per time step. Figure 6.21 shows that the elastoplastic model (blue line) initially presents a drop of performance of approximately 6% (for 200 vertices) when compared to the simple plastic model (red line). This performance drop eventually disappears as the number of vertices increases.

6.4.2.2 Performance Test 2

The same performance test was repeated by adding extra VOOs performing a similar elastic restoration and the speed of the process was measured. The deformation was performed with precalculation of damping elements. The results are shown in the graph of figure 6.22 (see blue line). For comparison, the speed graph of a plastic deformation test is superimposed (see red line). The increase in number of VOOs operating on an object reduces the performance of the elastoplastic restoration process. This is due to the fact that the computation overhead of the elastoplastic model increases as the number of VOOs increases. With one VOO there is a drop of performance of approximately 6% and with six VOOs there is a drop of performance of approximately 8%.

6.5 Summary and Conclusion

A VOO is a strain application tool which may be used to deform space interactively or in animation. By using a VOO, there is no need to apply stress directly on to objects. In this chapter we have seen that a VOO may also be engaged in a time-based elastic restoration of the

afflicted strain. Assuming that an amount of mass, m, is concentrated at the tip or the head of a VOO and by assigning an elasticity coefficient, s, a VOO becomes capable of harmonic oscillation. Elastic restoration is achieved by damping the energy of the oscillation using an energy damping function, $\zeta(t)$, over a time period of $(t_{max} - t_0)$.

The shape of the stress / strain graph with the elastic and fracture strain limits are properties of a VOO and its surrounding local space. All three, may be directly controlled by the user. These elements when combined with a user defined mass, an energy damping function and a duration of elastic restoration enable the creation of custom-made elastoplastic material behaviour. Furthermore, a stress / strain graph may be based on real experimental data and thus lead to an approximation of realistic behaviour of specific materials. The combination of an elastoplastic model with a local deformation model, leads one step closer to the automation of organic material behaviour in computer animated objects. As a result, much of animation time and effort may be dedicated to other more important animation problems.

The original concepts introduced in this chapter are:

- the use of user defined stress / strain graphs for the specification of elasticity coefficients and permanent plastic strains,
- the incorporation of energy damping graphs for constraining the elastic restoration process.

In the following chapter, the deformation model will be extended to incorporate viscoelastic material behaviour.

6.6 Colour Plates

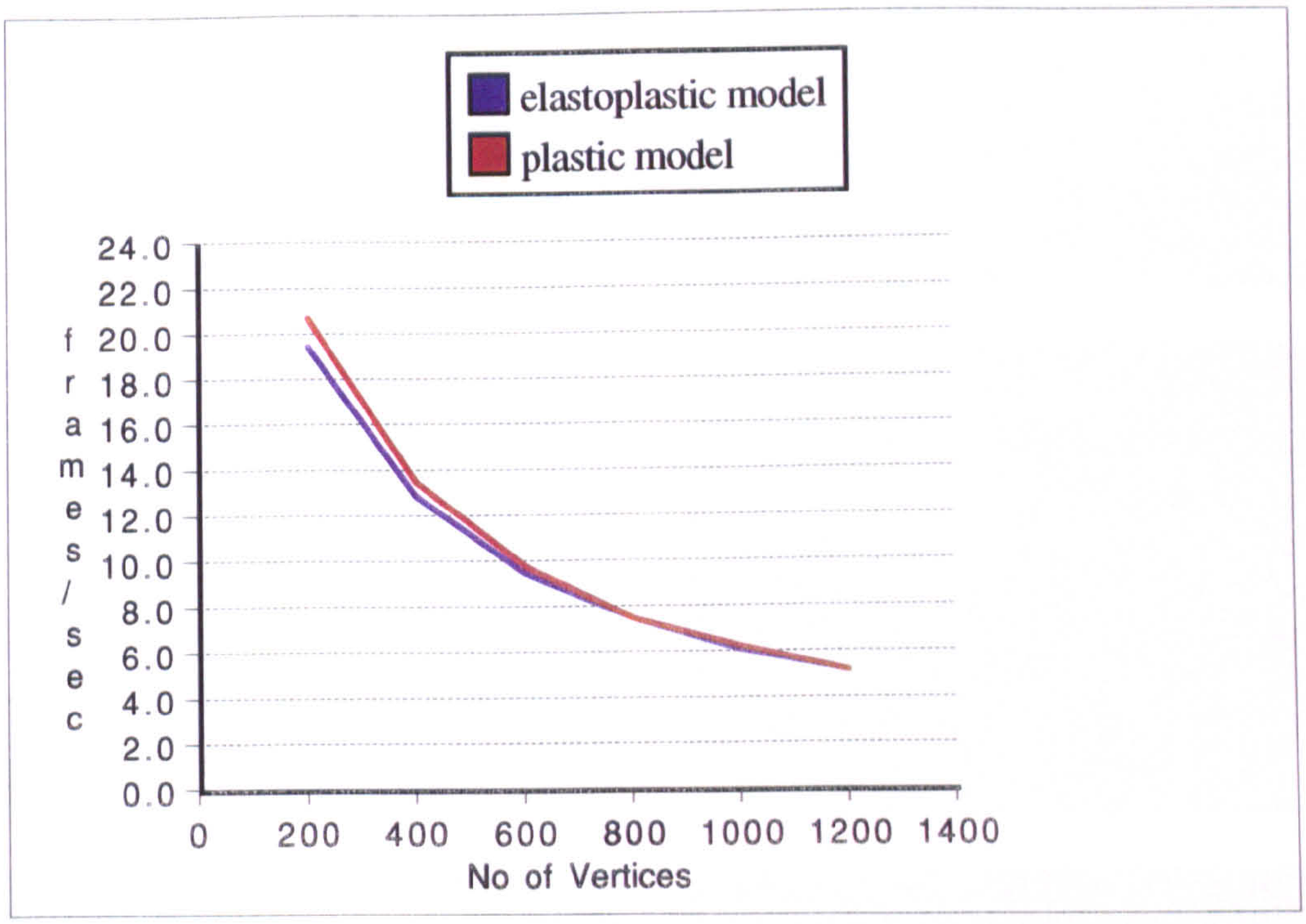


Figure 6.21 Graph of speed of elastoplastic deformation over the number of vertices.

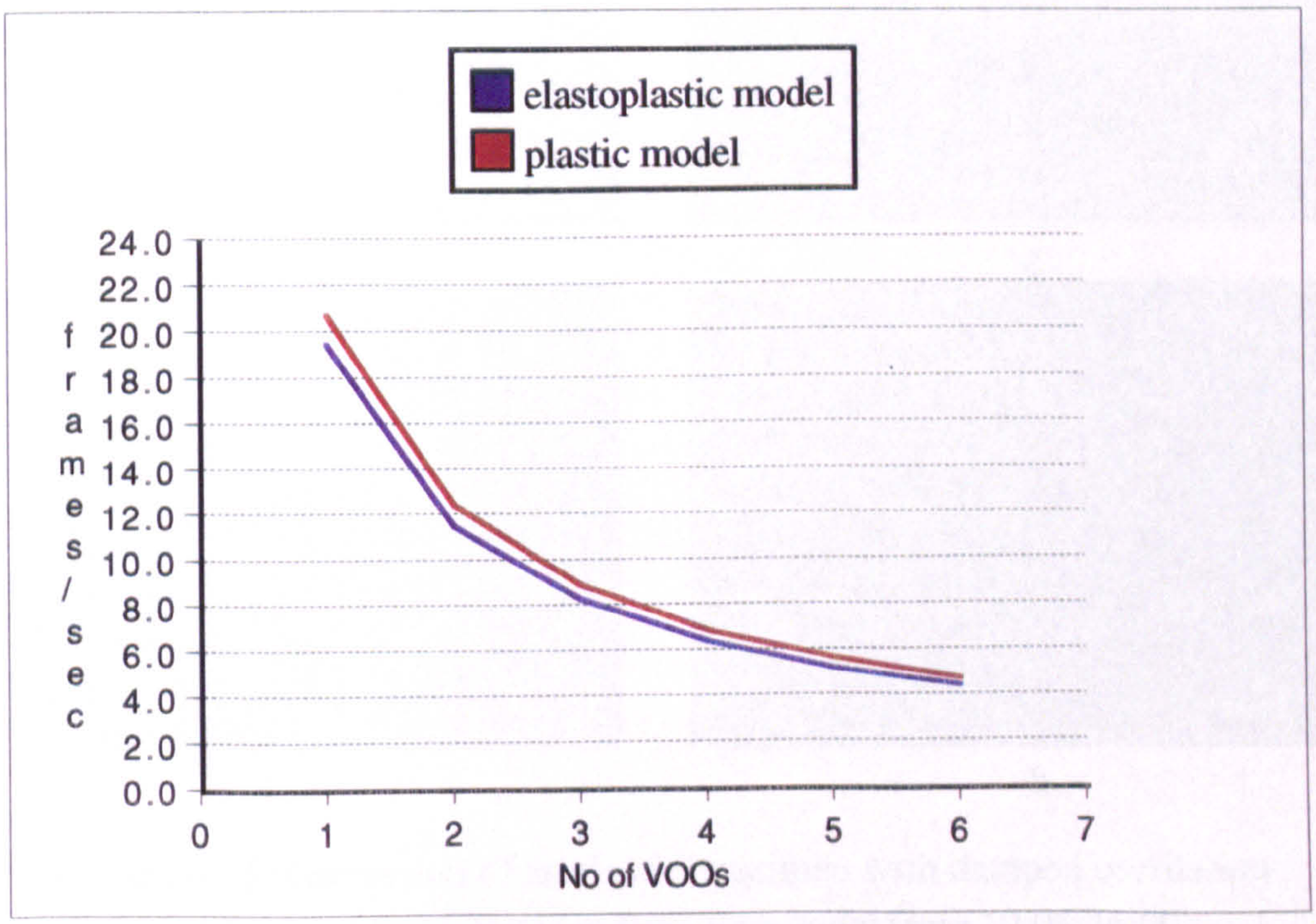
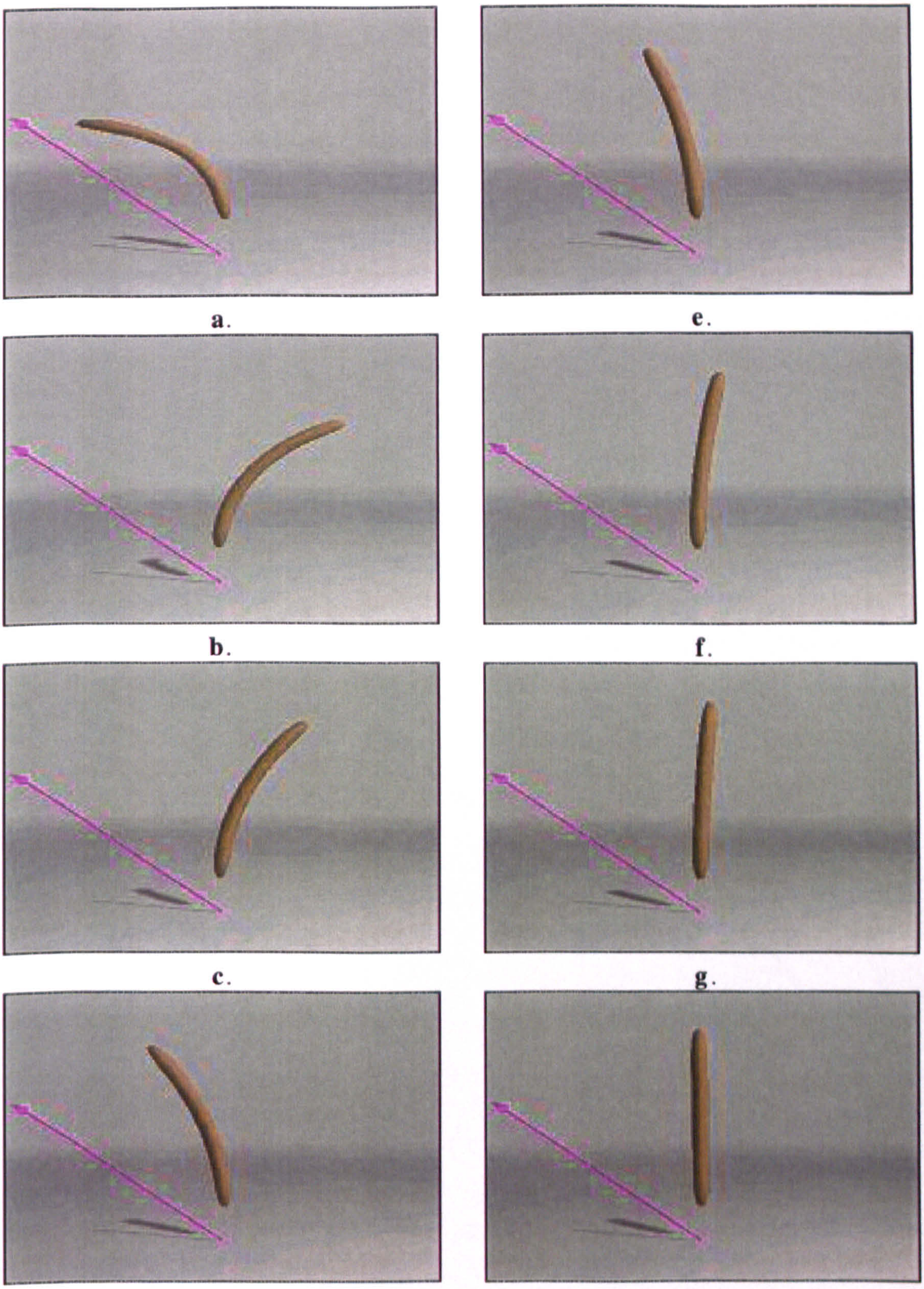


Figure 6.22 Graph of speed of elastoplastic deformation over the number of VOOs.



a. **e.**

b. **f.**

c. **g.**

d. **h.**

Figure 6.23 Restoration of an elastic specimen with damped oscillation (see animation test 4 in video tape, time code: from 10:04:46:00).

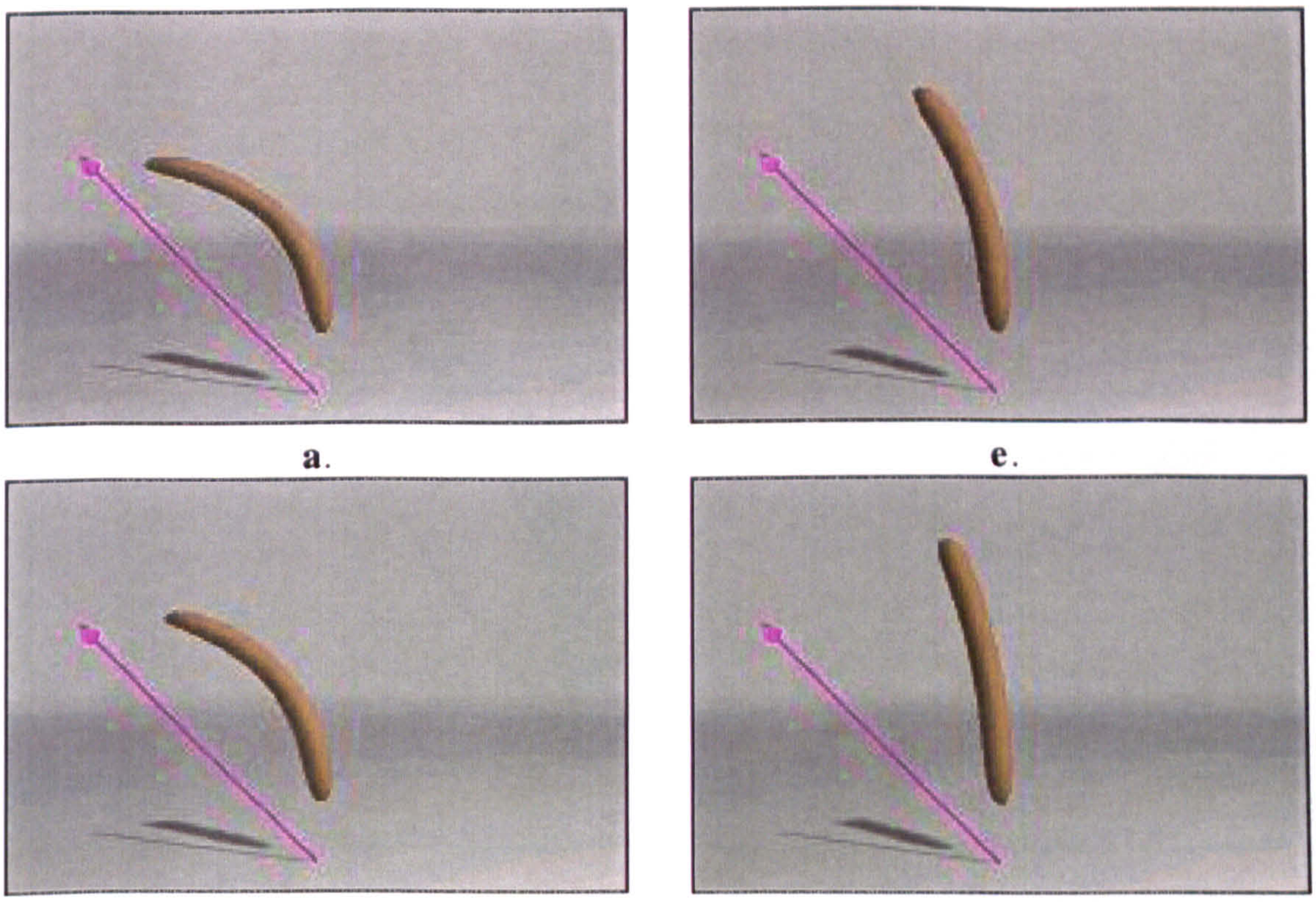


Figure 6.24 Restoration of an elastic specimen without oscillation (see animation test 5 in video tape, time code: from 10:04:59:00).

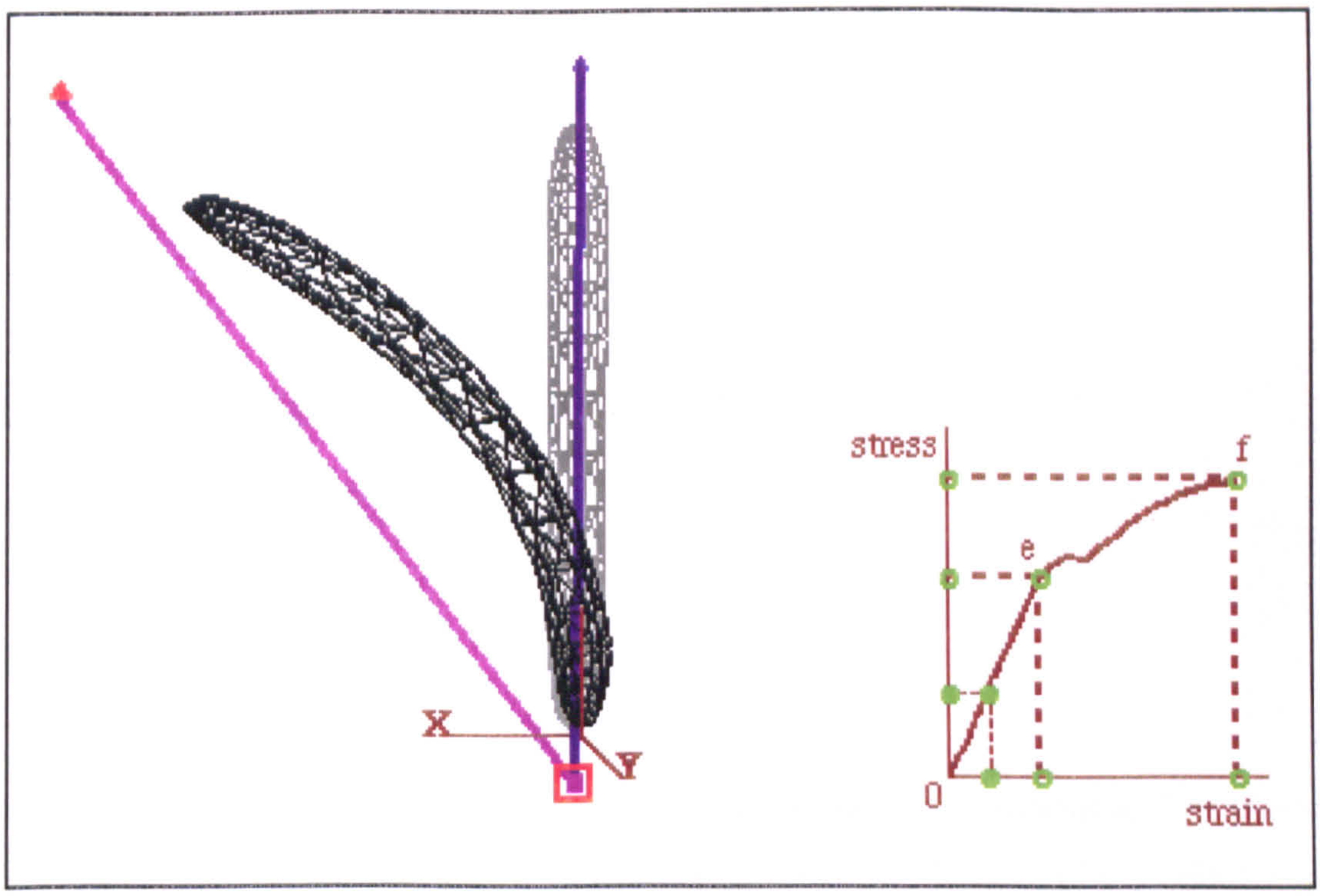


Figure 6.25 An elastoplastic VOO equipped with a stress / strain graph.

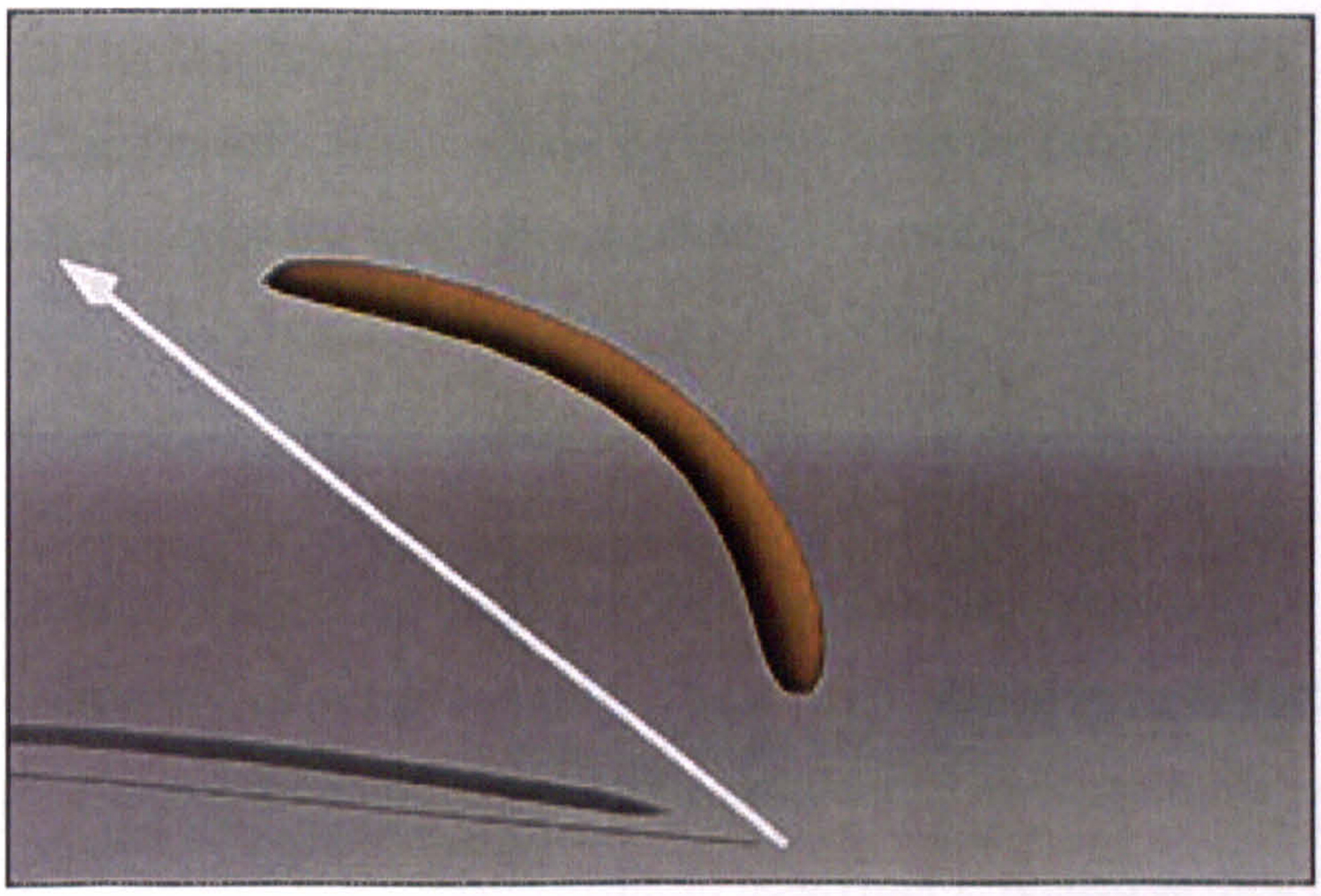


Figure 6.26 A still from animation test 4
(see video tape, time code: from 10:04:46:00).

7

The Viscoelastic Model

7.1 Introduction

Organic materials, such as living tissue in physiological state are viscoelastic. The phenomena of strain creep, stress relaxation and hysteresis are characteristic features of viscoelastic material behaviour. A self-activated model for animation of viscoelastic objects could replace a great amount of repetitive animation work. As a result, the creation of animated characters based on living creatures would be greatly facilitated.

The objectives of this chapter are:

- to extend the deformation model by incorporating the property of viscoelasticity,
- to enable interaction with viscoelastically deformable objects,
- to enable efficient animation testing and rendering,
- to enable application of temporal constraints to viscoelastic objects.

The deformation model introduced in the previous chapters (see chapters 5 and 6) is extended to incorporate the property of viscoelasticity. The overall objective is to further improve the realistic look of animated deformable objects by providing a mechanism for generating natural organic material behaviour. Viscoelastic material behaviour is related to time and therefore, the model must provide a mechanism for temporal constraints. The model must also be efficient, so as to enable effective interactive manipulation of viscoelastically deformable objects.

7.2 Strain Creep

In chapter 3, it was shown that organic materials exhibit the viscoelastic feature of strain creep (see figure 3.14 in section 3.5). So far in this study, it has been assumed that strain is applied on the entire body of an object instantaneously. However, in organic viscoelastic materials strain develops over time. Therefore, strain creep is an inherently time related phenomenon. From the initial moment of stress application onwards the propagation of strain inside the body of a deformable object generates a time based deformation which is visually very interesting. Strain creep creates a fluid like motion and if constrained properly, may be used in the animation of organic characters or natural phenomena.

7.2.1 The Plastic Strain Rate

A VOO may be subjected to a time based motion, either through interactive manipulation or by participating in an animated sequence. As a result, for every time t, an object which is influenced by a moving VOO will experience a strain and a strain rate (i.e. velocity of deformation). The strain vector depends on the deformation characteristics of the VOO (see chapter 5). The strain rate vector depends on the motion characteristics of the VOO.

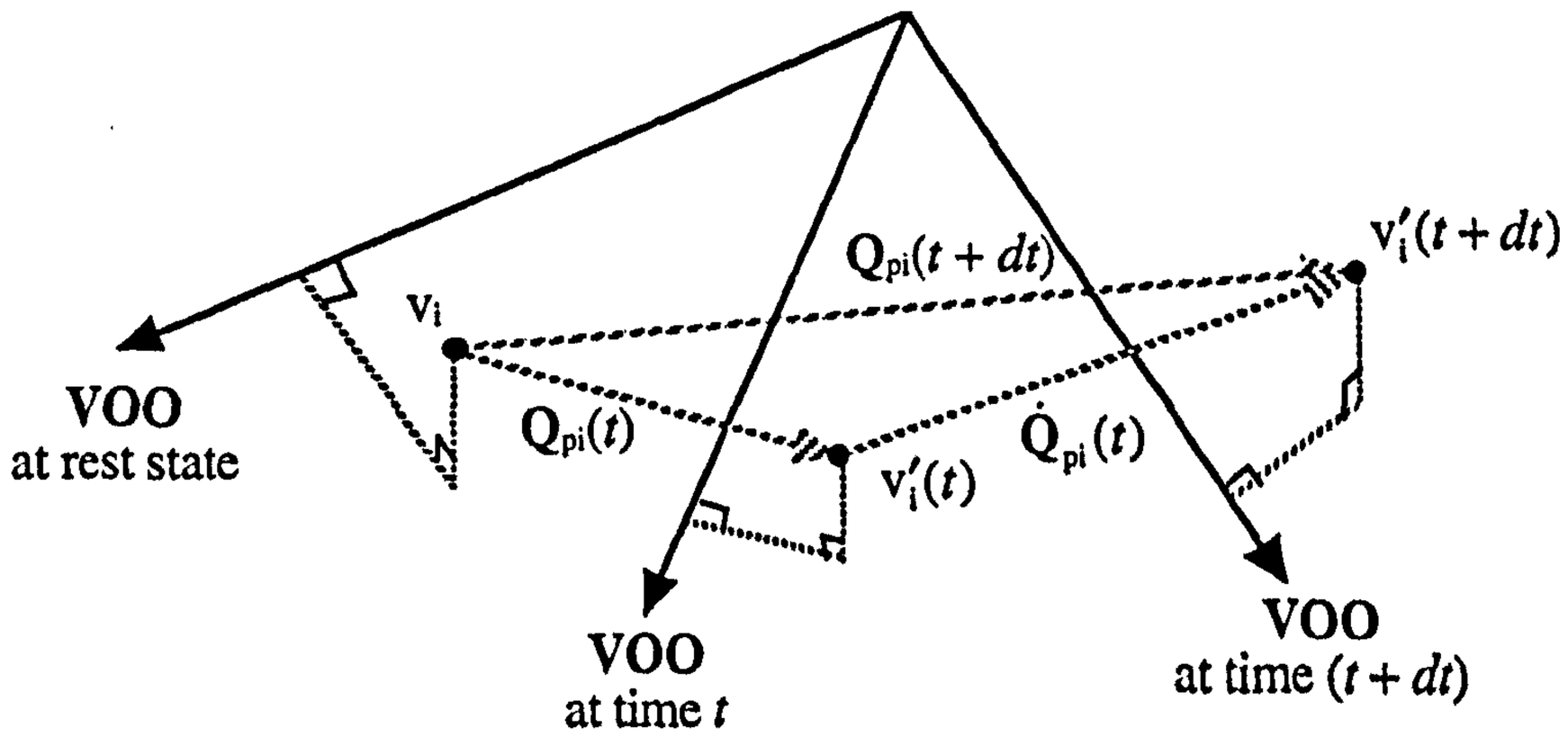


Figure 7.1 Plastic strain $\mathbf{Q}_{pi}(t)$ and plastic strain rate $\dot{\mathbf{Q}}_{pi}(t)$ of vertex $\mathbf{v}_i$ at time t.

In figure 7.1, a VOO is depicted at rest state, at time t and at time $(t + dt)$, where dt is a very small time step. For simplicity, only the tip of the VOO is moved. At times t and $(t + dt)$ an active vertex v_i has received amounts of strain which are represented by the plastic strain vectors $\mathbf{Q}_{pi}(t)$ and $\mathbf{Q}_{pi}(t + dt)$, respectively (see equation [5.47] in section 5.7 for the calculation of the plastic strain vector $\mathbf{Q}_{pi}(t)$). The plastic strain rate of vertex v_i may be given by the following equation:

$$\dot{\mathbf{Q}}_{pi}(t) = \frac{\left(\mathbf{Q}_{pi}(t+dt) - \mathbf{Q}_{pi}(t)\right)}{dt} = \frac{d\mathbf{Q}_{pi}(t)}{dt} \qquad [7.1]$$

A strain rate is imposed on every active vertex by the motion of the VOO. This motion may be interactively specified by the user. Also, in an animated sequence a VOO may be moved by interpolating between key positions or by applying a motion path or other spatial and temporal constraints to the tip and head of a VOO (see chapter 8).

7.2.2 The Viscous Strain Rate

In a viscous fluid, a steady external stress produces a steady rate of strain flow. Also, in a viscous fluid there is a spatial gradient of strain rate which is proportional to the distance from the stress application area (see figure 3.5 in section 3.4).

7.2.2.1 The Spatial Gradient of the Viscous Strain Rate

A spatial gradient may be applied to the plastic strain rate of equation [7.1] in order to produce a viscous strain rate for every vertex v_i. The field of influence of a VOO may be used as the basis for the application of this strain rate gradient.

In a polar VOO, vertices that lie close to the centre of strain application (i.e. the tip or head of a VOO) will receive a maximum strain rate, whereas vertices which lie close to the extremities of

the field will receive a minimum strain rate (see figure 7.2).

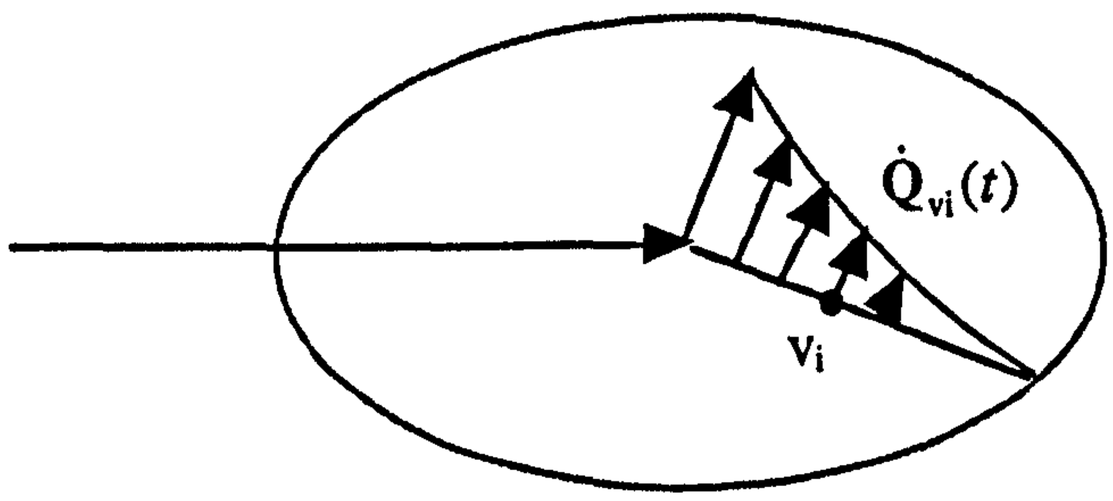


Figure 7.2 A viscous strain rate develops a spatial gradient inside the field of influence of a polar VOO.

In a length VOO, vertices that lie close to the VOO itself will receive a maximum strain rate, whereas vertices which lie close to the edges of the field will receive a minimum strain rate (see figure 7.3).

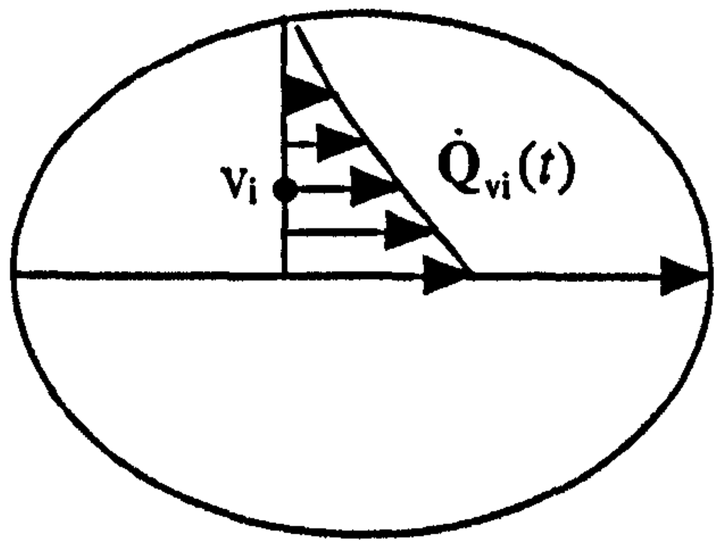


Figure 7.3 A viscous strain rate develops a spatial gradient inside the field of influence of a length VOO.

The viscous spatial gradient may be derived from a user defined graph (see figure 7.4). Every vertex v_i has a normalised distance δ_i, which represents the relative position of a vertex inside the field of influence of a VOO (see section 5.5).

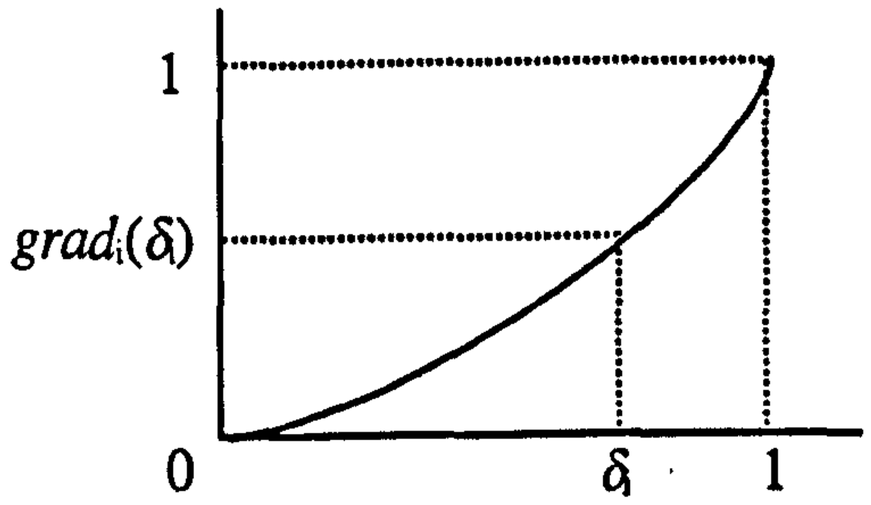


Figure 7.4 Graph of spatial gradient of viscous strain rate.

This distance δ_i, may be used in conjunction with a gradient graph to generate viscous strain rate for a vertex.

$$\dot{\mathbf{Q}}_{vi}(t) = \dot{\mathbf{Q}}_{pi}(t) \cdot \left(1 - v \cdot grad_i(\delta_i)\right) \qquad [7.2]$$

In equation [7.2], subscript $_v$ stands for viscous and v is the viscosity coefficient.

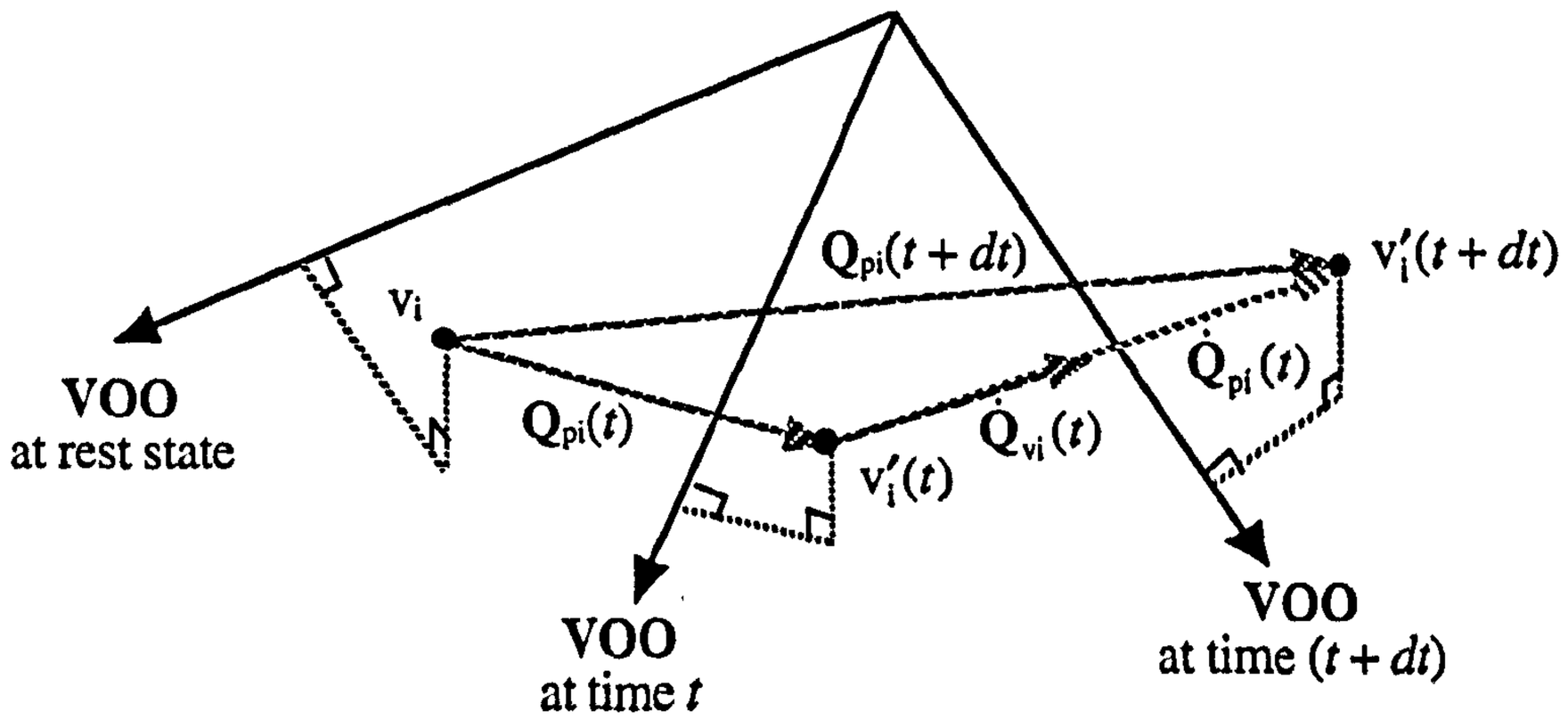
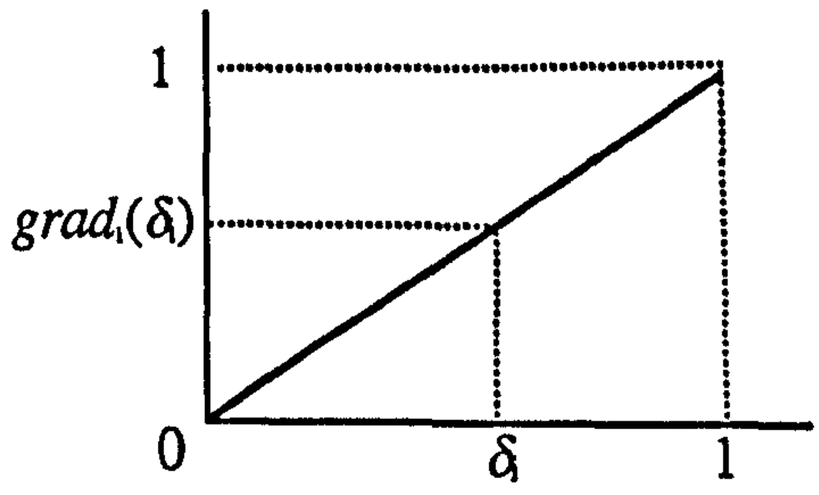


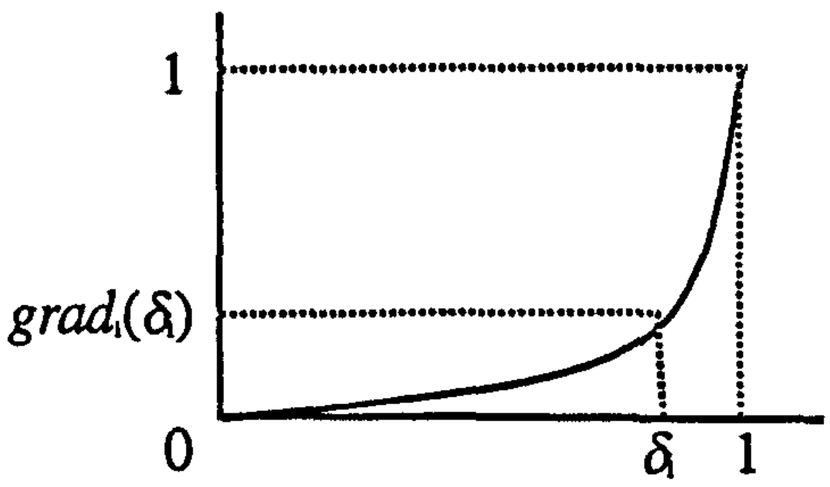
Figure 7.5 Viscous strain rate $\dot{\mathbf{Q}}_{vi}(t)$ of vertex v_i at time t.

In figure 7.5, vector $\dot{\mathbf{Q}}_{vi}(t)$ represents the viscous strain rate of vertex v_i at time t and it is equal to a portion of the plastic strain rate vector $\dot{\mathbf{Q}}_{pi}(t)$. So, each vertex receives a different viscous strain rate depending on its position inside the field of influence of a VOO. The result is a viscous fluid motion inside the volume of an object.

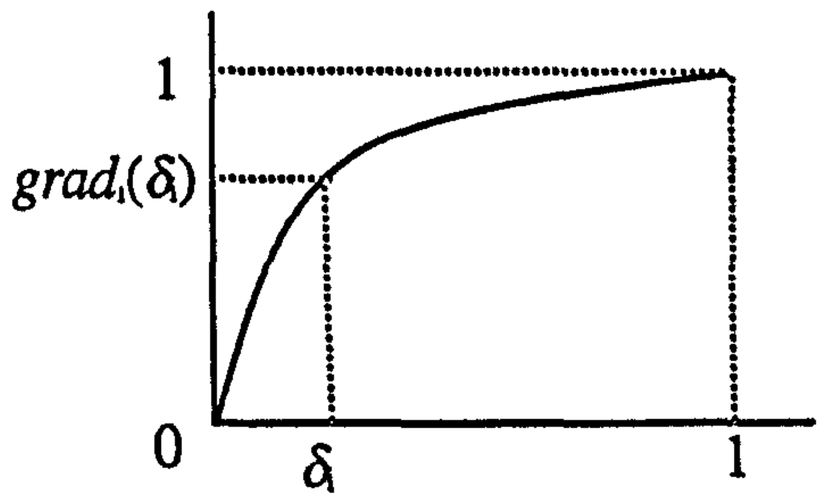
a. Elastoplastic material.



b. Newtonian viscous fluid.



c. Thick, sticky fluid.



d. Thin, runny fluid.

Figure 7.6 Various viscous strain rate gradients.

The user may manipulate the strain rate gradient graph in order to achieve the desired viscous or non-viscous material behaviour (see figure 7.6).

7.2.2.2 The Viscosity Coefficient

The viscosity coefficient, v, may be used as a scaling factor of the effect of a user defined graph of strain rate gradient.

$$v = 0 \quad \Rightarrow \mathbf{Q}_{vi}(t) = \mathbf{Q}_{pi}(t) \quad \text{(elastoplastic material)} \qquad [7.3]$$

$$0 < v \leq 1 \Rightarrow \dot{\mathbf{Q}}_{vi}(t) < \dot{\mathbf{Q}}_{pi}(t) \quad \text{(viscous material)} \qquad [7.4]$$

If the value of v becomes greater than 1 then in certain areas inside the field of influence the direction of the viscous strain rate vector will become opposite to the plastic strain rate vector. This may cause unpredictable yet visually interesting results. Also, if the value of v becomes less than 0 then the magnitude of the viscous strain rate may become greater than the magnitude

of the plastic strain rate, again causing unpredictable results. The viscosity coefficient in the presented model is a simple scaling factor and has no direct relationship with the viscosity of real materials.

7.2.2.3 Inertia in a Viscous Fluid

Normally, the magnitude of the viscous strain rate is smaller than the magnitude of the corresponding plastic strain rate. Therefore, some active vertices under the influence of a moving VOO may move slower than others. As a consequence, at a certain time, parts of a viscous object may not have yet reached their corresponding plastic strain. When a VOO motion is suddenly interrupted, certain parts of a viscous object will keep on moving under the influence of inertia. Assuming that a mass is concentrated on each such vertex, then each vertex will follow a decelerated motion due to its initial momentum until it reaches its corresponding plastic strain at rest state. This means that parts of an object may continue to deform even after a VOO has ceased moving. However, the philosophy of the suggested model is based on a VOO being used as a strain application tool suitable for animation. A VOO may be manipulated via user specified spatial and temporal constraints. In animation, basic spatial and temporal constraints such as start and end position and time are of vital importance. Allowing independent movement of objects due to inertia would lead to situations which are difficult to constrain and which result in inefficient formulations. Therefore, the suggested model focuses on the viscoelastic response of an object only during the constrained motion of a VOO. For a detailed analysis on spatial and temporal constraints for VOOs see chapter 8.

7.2.3 The Viscoelastic Strain Rate

A viscoelastic material initially exhibits a viscous fluid behaviour which gradually turns into ordinary elastoplastic material behaviour. During this period of transition the strain creep phenomenon takes place. Assuming that a stress is applied at a time t_0, the strain creeps from t_0 to a time t_{max}. If stress application continues after t_{max}, from that moment onwards the material

behaves as an elastoplastic object. When the stress is finally removed elastic restoration takes place (see chapter 6).

7.2.3.1 The Strain Creep Function

The strain creep phenomenon may be implemented using a *creep*(*t*) function of time in conjunction with the viscous strain rate from equation [7.2]:

$$\dot{\mathbf{Q}}_{vei}(t) = \dot{\mathbf{Q}}_{pi}(t) \cdot \left(1 - \left(\nu \cdot grad_i(\delta_i) \cdot (1 - \mu \cdot creep(t))\right)\right) \qquad [7.5]$$

where the subscript $_{ve}$ stands for viscoelastic and μ is the creep coefficient. The *creep*(*t*) function may be based on a user defined creep / time graph (see figure 7.7).

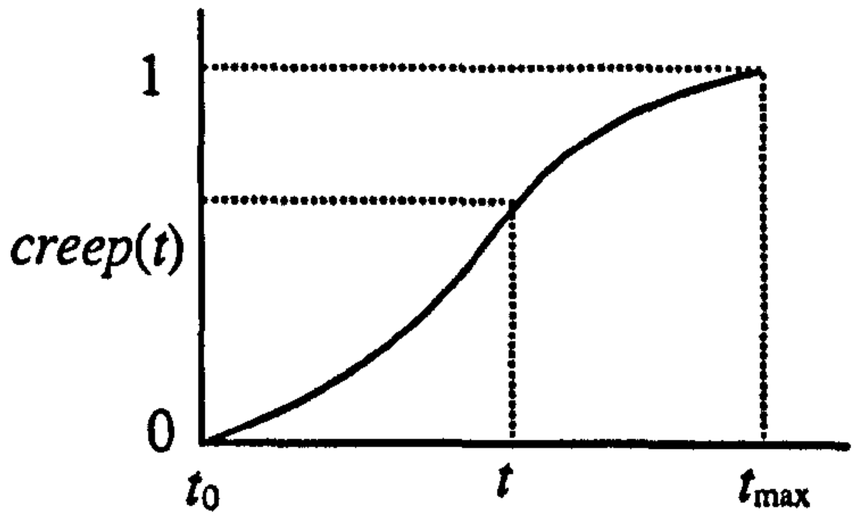
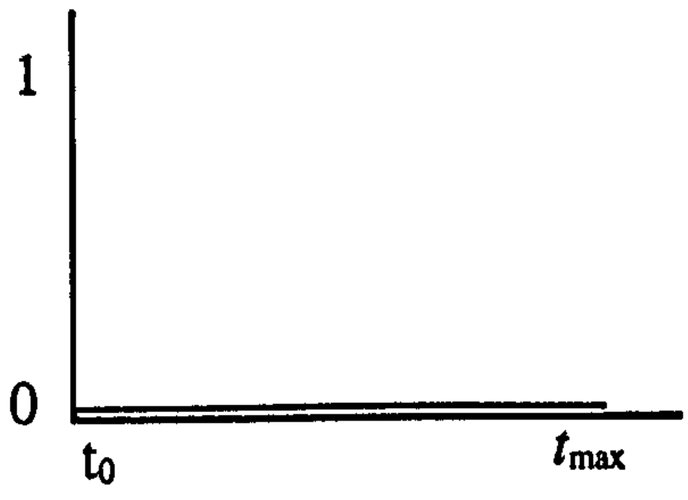


Figure 7.7 A strain creep graph.

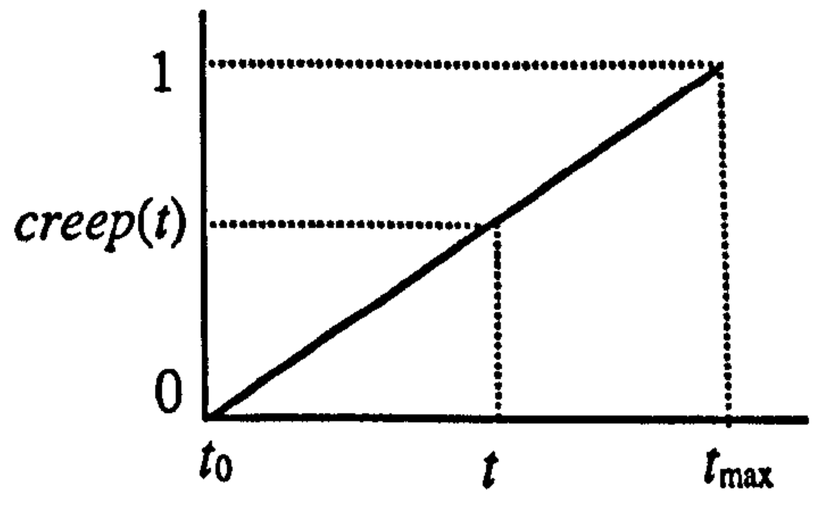
Between times t_0 and t_{max}, the *creep*(*t*) function receives values between 0 and 1 (see figure 7.7). As a result, the viscoelastic strain rate starts from being equal to the viscous strain rate and is accelerated to finally become equal to the plastic strain rate:

$$t_0 \le t \le t_{max} \quad \Rightarrow \quad 0 \le creep(t) \le 1 \quad \Rightarrow \quad \dot{\mathbf{Q}}_{vi}(t) \le \dot{\mathbf{Q}}_{vei}(t) \le \dot{\mathbf{Q}}_{pi}(t) \qquad [7.6]$$

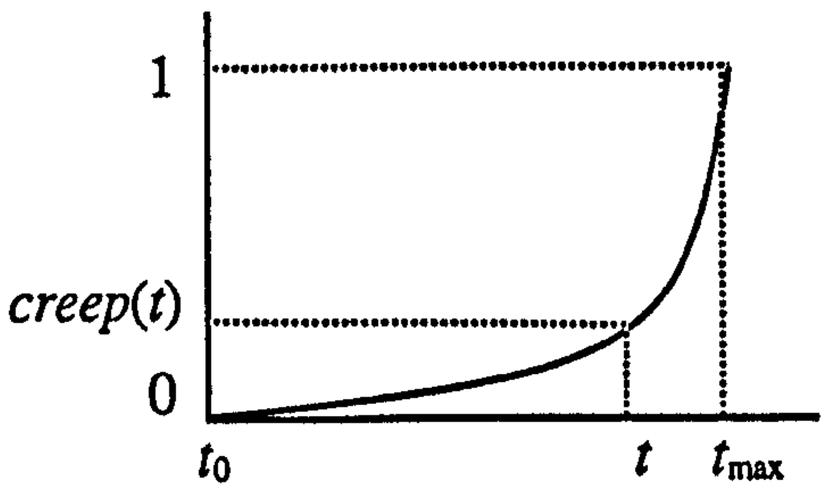
The *creep*(*t*) function in equation [7.5], is in fact used to accelerate or decelerate the viscous strain rate. This tends to gradually diminish the difference between the viscous and the plastic strain rates and thus to transform any viscous behaviour into an elastoplastic behaviour.



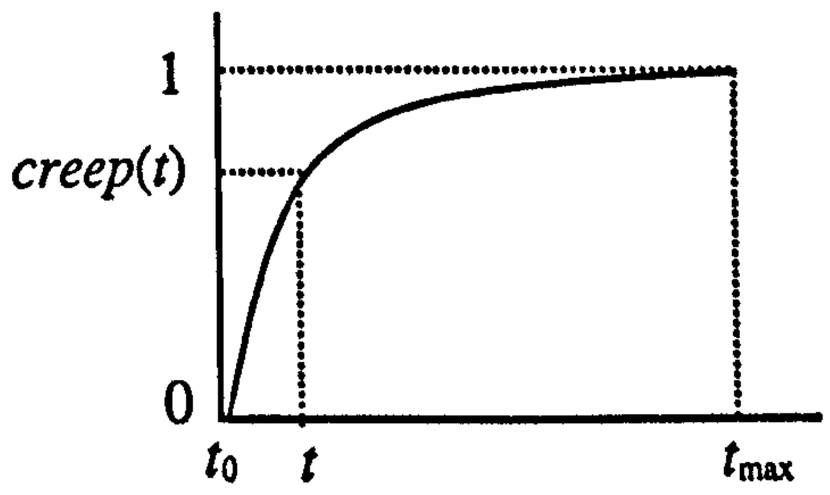
a. Viscous material with no strain creep.



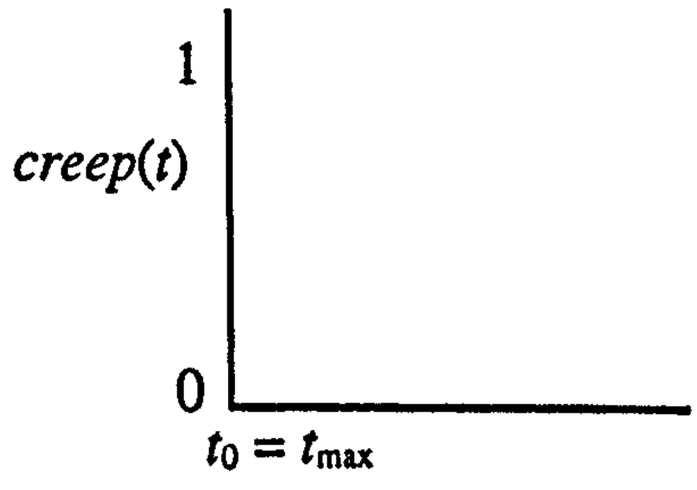
b. Viscoelastic material with linearly accelerated strain.



c. Viscoelastic material with slowly accelerated strain.



d. Viscoelastic material with rapidly accelerated strain.



e. Elastoplastic material.

Figure 7.8 Various strain creep / time graphs.

The user may manipulate a *creep*(*t*) graph. This offers the user the freedom to experiment visually and to custom design the relationship between strain and time. A wide range of material behaviours may thus be generated from perfectly viscous to perfectly elastoplastic (see figure 7.8).

7.2.3.2 Duration of the Strain Creep Process

The duration of the strain creep process (t_{max} - t_0) may also be directly specified by the user. Time t_0 represents the moment at which a VOO is set in motion. Time t_{max} represents the moment by which all active vertices have reached their corresponding plastic strain rate. If a VOO keeps on moving from t_{max} onwards an influenced object will simply exhibit elastoplastic behaviour as described by the stress / strain curve.

Although it may seem somewhat arbitrary, the specification of an appropriate maximum strain creep time is very important for animation. The duration of an event in animation is usually predetermined and influenced by other events or temporal constraints. So, the user specified duration of the strain creep process acts as a user imposed temporal constraint for viscoelastic behaviour. Choosing a long strain creep duration may cause interactive manipulation of viscoelastic objects to become very slow and awkward to control. So, it is preferable to keep the creep duration as short as possible during interactive manipulation.

7.2.3.3 The Strain Creep Coefficient

The strain creep coefficient, μ, may be used to scale the effect of the strain *creep*(t) function (see equation [7.5]).

$$\mu = 0 \Rightarrow \text{viscous material with no strain creep} \qquad [7.7]$$

$$\mu = 1 \Rightarrow \text{viscoelastic material with strain}creep(t) \qquad [7.8]$$

The user may experiment with different values for the strain creep coefficient and combine them with various strain *creep*(t) graphs. This allows the user to generate a broad range of material behaviours from viscous to viscoelastic. The strain creep coefficient is a simple scaling factor with no physical counterpart.

7.2.4 The Implementation of Strain Creep

The objective is to calculate the deformed position of every active vertex of a viscoelastic object at a time *t*. The plastic strain and strain rate for every vertex are calculated as described in chapter 5. The normalised distance, δ_i, of every vertex v_i is used for the precalculation of a strain rate gradient value, $grad_i(\delta_i)$, which is derived from the user defined strain rate gradient graph (see figure 7.4). This precalculation takes place along with the precalculation of strain damping elements (see section 5.6) and its purpose is to improve the efficiency of the time related calculations. The strain creep value, *creep*(*t*), is derived from the user defined strain creep graph (see figure 7.7) once per time step per VOO for times between t_0 and t_{max}. The viscosity coefficient, ν and the creep coefficient, μ, are directly specified by the user. The viscoelastic strain rate is calculated as a portion of the plastic strain rate using equation [7.5]. The viscoelastic strain $\mathbf{Q}_{vei}(t)$ of a vertex v_i at time *t* may then be calculated as follows:

$$\mathbf{Q}_{vei}(t) = \int_{t_0}^{t} \dot{\mathbf{Q}}_{vei}(t)dt \qquad \text{(assuming that} \quad \mathbf{Q}_{vei}(t_0) = 0) \qquad [7.9]$$

The global position vector of a viscoelastically deformed vertex may be calculated as follows:

$$\mathbf{r}_{vei}(t) = \mathbf{r}_i + \mathbf{T}(t) + \mathbf{Q}_{vei}(t) \qquad [7.10]$$

In equation [7.10], $\mathbf{r}_i$ is the global position vector of an undeformed vertex v_i at rest state and $\mathbf{T}(t)$ is the global translation vector. It should be noted that the viscoelastic creep phenomenon affects only the local strain $\mathbf{Q}_{vei}(t)$ and not the global translation $\mathbf{T}(t)$. The integration of equation [7.9] for the calculation of $\mathbf{Q}_{vei}(t)$ at time *t* may be performed using the following pseudocode:

$time = t_0 + dt$
repeat
    creep(*time*) = read value from strain creep graph
    for every vertex v_i
        calculate plastic strain $\mathbf{Q}_{pi}(time)$

$$\dot{\mathbf{Q}}_{pi}(time) = \frac{\left(\mathbf{Q}_{pi}(time) - \mathbf{Q}_{pi}(time - dt)\right)}{dt}$$

$$\dot{\mathbf{Q}}_{vei}(time) = \dot{\mathbf{Q}}_{pi}(time) \cdot \left(1 - \left(v \cdot grad_i(\delta_i) \cdot \left(1 - \mu \cdot creep(time)\right)\right)\right)$$

$$\mathbf{Q}_{vei}(time) = \mathbf{Q}_{vei}(time - dt) + \dot{\mathbf{Q}}_{vei}(time)$$

    $time = time + dt$
until $time > t$ **[7.11]**

In the above pseudocode, vectors $\mathbf{Q}_{pi}(time - dt)$ and $\mathbf{Q}_{vei}(time - dt)$ have been calculated and temporarily stored during the previous time step ($time - dt$). The spatial gradient factors, $grad_i(\delta_i)$, have been precalculated outside the integration loop. The integration in pseudocode [7.11] is performed for every active vertex of an object. So, the total number of vertices in an object can seriously affect the efficiency of the viscoelastic creep process. The size of the time step, *dt*, is another factor which can affect the efficiency of the model. The time step is normally set to one frame (usually equal to 1/25th of a second) but it may also be set to two or more frames in order to improve the performance of the interactive process. The integration method in pseudocode [7.11] is likely to result in numerical instability. The current solution is good enough for demonstration purposes but for a commercial implementation a more robust method could be implemented (see Taylor and Euler methods (Press *et al.* 1986)).

If there are more than one VOOs affecting the same vertex v_i then a viscoelastic strain $\mathbf{Q}_{veij}(t)$ must be calculated for every VOO $_j$. Equation [7.10], will then become:

$$\mathbf{r}_{vei}(t) = \mathbf{r}_i + \sum_{j=1}^{nv}\left(\mathbf{T}_j(t) + \mathbf{Q}_{veij}(t)\right) \qquad \textbf{[7.12]}$$

In equation [7.12], *nv* is the number of VOOs operating on vertex v_i.

7.3 Stress Relaxation

In chapter 3, it was shown that organic materials exhibit the viscoelastic feature of stress relaxation (see figure 3.15 in section 3.5). In such materials, internal elastic stresses induced inside the body of an object decrease with time. As a consequence, the amount of elastic energy stored inside an object is gradually being converted to permanent deformation. The elastic restoration process becomes slower and is finally diminished after a considerable time lapse. Therefore, the occurrence of stress relaxation in a viscoelastic material is only expressed indirectly through the elastic restoration process. The phenomenon of stress relaxation does not contribute directly to the visual behaviour of a deforming object and it is, therefore, not very useful in animation. Similarly to strain creep, stress relaxation is an inherently time related phenomenon. A user defined stress relaxation graph (see figure 7.9) may be used to "relax" the internal elastic stresses which cause elastic restoration.

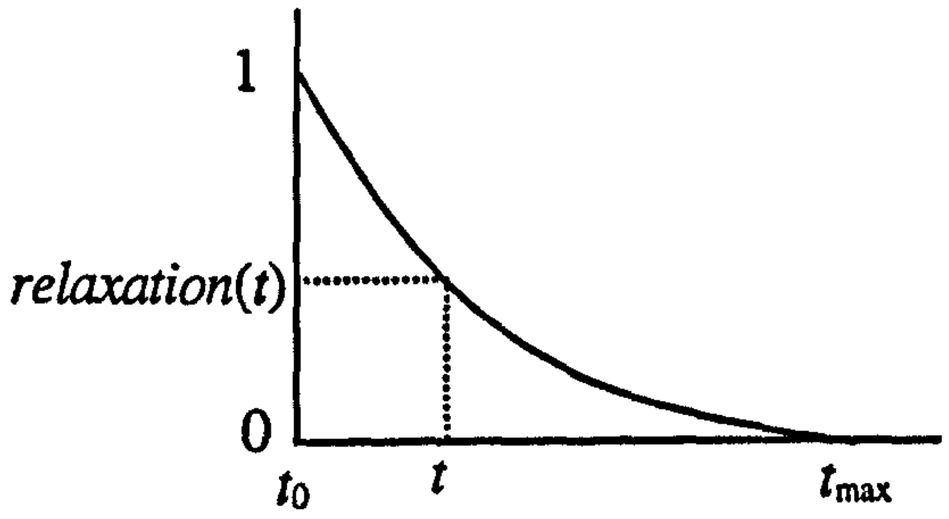


Figure 7.9 A stress relaxation graph.

The graph in figure 7.9, describes how stress relaxes from time t_0 of the initial stress application up to a user defined t_{max}. The values of the *relaxation*(t) function range between 0 and 1.

In chapter 6, the velocity of elastic restoration was calculated using either of the equations [6.23], [6.25] or [6.27]. The elasticity coefficient, $s(e)$, which is used in these equations is directly derived from the stress / strain graph of the material. So, if the object is viscoelastic the value of the elasticity coefficient will be affected by the stress relaxation phenomenon.

At a time t after the time of initial stress application t_0 the elasticity coefficient becomes:

- If inside the elastic region of the stress / strain graph:
  (see figure 6.18 in section 6.3.2)

$$s(e) = \tan(\theta) \cdot relaxation(t) \qquad [7.13]$$

- If inside the plastic region of the stress / strain graph:
  (see equation [6.35] in section 6.3.3)

$$s(e) = \frac{(\sigma_e \cdot relaxation(t))}{e_e} \qquad [7.14]$$

The permanent plastic strain become (see equation [6.36] in section 6.3.3):

$$e_p(e) = e - \frac{(\sigma \cdot relaxation(t))}{s(e)} \qquad [7.15]$$

In equations [7.13], [7.14] and [7.15], the stress relaxation function is in fact used to scale the stress axis of the stress / strain graph (see figure 7.10).

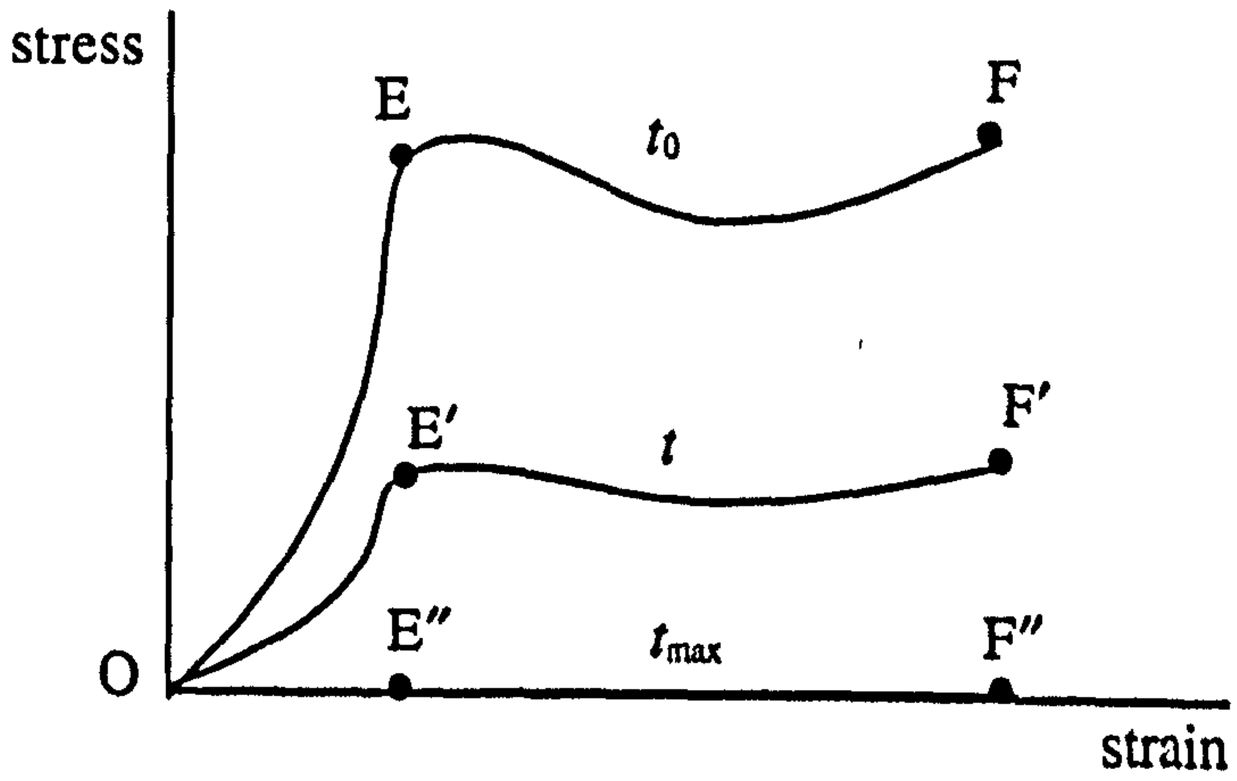


Figure 7.10 The effect of stress relaxation on a stress / strain graph from time t_0 to t_{max}.

At a time t, the stress axis of the graph appears to be scaled by the value of *relaxation*(t) (see

line OE′F′ in figure 7.10). Finally at time t_{max}, the stress axis is scaled by zero and the graph becomes a straight line coincident with the strain axis, which represents a perfectly plastic material (see line OE″F″ in figure 7.10) (see also figure 6.16b in section 6.3.1). Therefore, the effect of stress relaxation is expressed indirectly through the scaling of the elasticity coefficient and the permanent plastic strain during the elastic restoration period.

7.4 Hysteresis

A cyclic loading process causes the phenomenon of hysteresis. Stress / strain hysteresis is present in all materials to some extent but it is particularly obvious in soft organic tissues (see figures 3.16 and 3.17a and b in section 3.5). The phenomenon of hysteresis is expressed as a deviation from the stress / strain graph of a material during the unloading and reloading process (see figure 7.11).

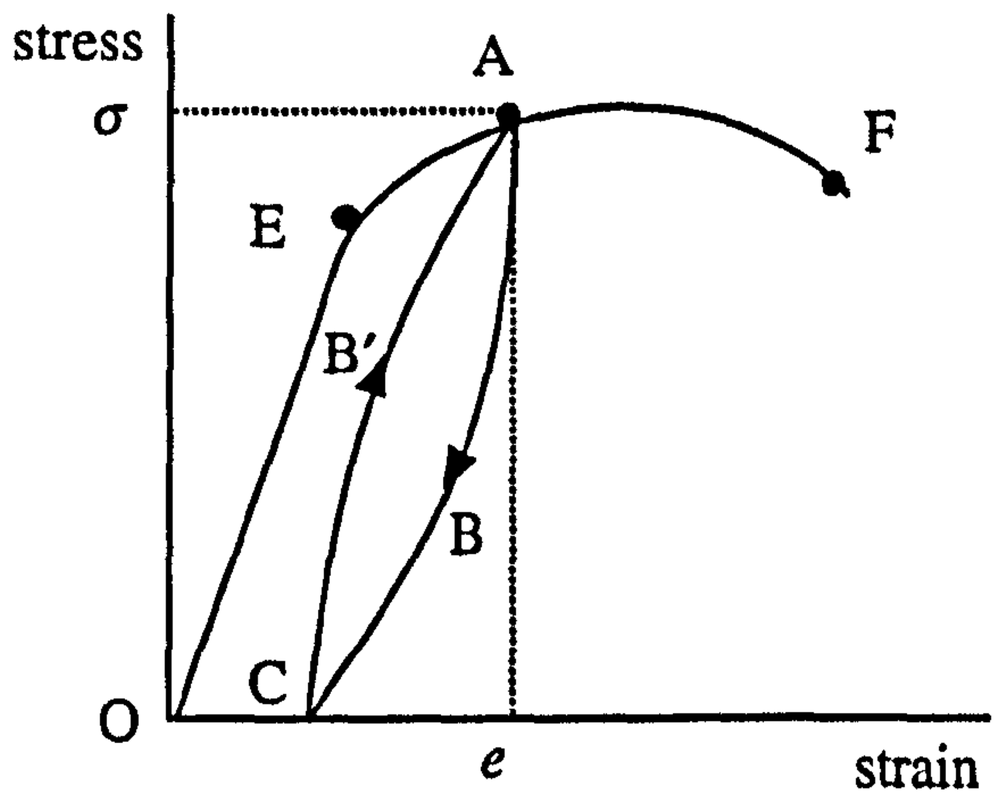


Figure 7.11 Hysteresis in a stress / strain graph.

In figure 7.11, the stress / strain path ABC describes the unloading process and CB′A the reloading process. The user may specify a hysteresis loop ABOB′ which is then scaled and superimposed on top of the stress / strain graph of the material (see figure 7.12).

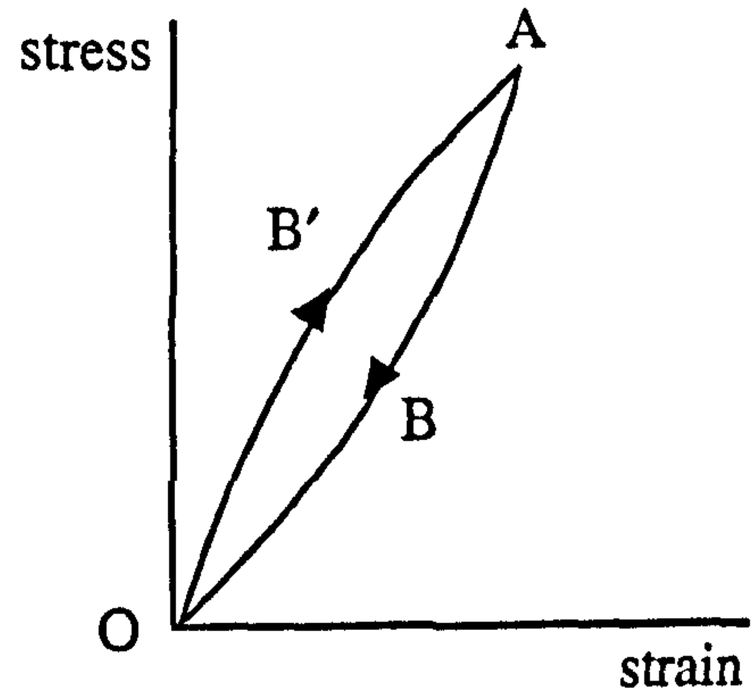


Figure 7.12 An hysteresis loop.

The hysteresis curve is then used for the derivation of the elasticity coefficient, $s(e)$, which is used for the calculation of the velocity of the elastic restoration process (see section 6.3.2 and 6.3.3). The phenomenon of hysteresis is not time dependent and it only affects the elastic restoration process. Hysteresis, much like stress relaxation, does not contribute directly to the visual behaviour of a deforming object and is not often used in animation.

7.5 Results

7.5.1 Effectiveness of the Viscoelastic Model

In this chapter, the deformation model has been extended to incorporate the property of viscoelasticity. The main objective was to provide an efficient and constrainable mechanism for generating realistic organic material behaviour in 3D computer graphics objects.

Viscoelasticity is a combination of viscosity and elasticity in the same material. A viscoelastic object behaves initially like a viscous fluid which is characterised by a spatial gradient of strain rate. The behaviour of the object gradually turns into elastoplastic which is characterised by a combination of elastic restoration and permanent plastic strain. Viscoelasticity is expressed through strain creep and stress relaxation, both time dependent phenomena and stress / strain hysteresis which is not time dependent. The suggested model provides a mechanism for the

implementation of these three characteristic viscoelastic features.

A spatial gradient of strain rate is used to generate a viscous effect in a deformable object. For this purpose, a strain rate gradient value, $grad_i(\delta_i)$, is precalculated for every vertex by incorporating a user defined spatial gradient graph (see figure 7.4). This value is then combined with a viscosity coefficient, v, in order to apply a spatial damping of strain rate inside the field of influence of a VOO (see equation [7.2]). The user is thus given the freedom to customise the viscous behaviour of an object by directly specifying the strain rate gradient and by manipulating the value of the viscosity coefficient. An object may be made to respond like a thin runny fluid or a thick sticky fluid or, at the other extreme, as an elastoplastic material with no viscosity at all (see figure 7.6). A user defined strain creep graph provides a temporal constraint for the development of viscoelastic creeping strain inside the volume of a 3D object (see figure 7.7). A time related creep value, $creep(t)$, is calculated once per time step and is used in conjunction with the creep coefficient, μ (see equation [7.5]). The strain creep mechanism is used to transform the initially viscous behaviour of an object into common elastoplastic behaviour over a user specified period of time (t_{max} - t_0). With this mechanism the user may customise the material behaviour of an object from perfectly viscous to viscoelastic and finally to elastoplastic (see figure 7.8). Also, a user defined stress relaxation graph is used to "relax" the internal elastic stresses over a period of time (see figure 7.9). For this purpose, a time related relaxation factor, $relaxation(t)$, is calculated once per time step and is used to scale the stress axis of the stress / strain graph (see figure 7.10). Finally, a user defined hysteresis graph is superimposed on top of the stress / strain graph of a material in order to describe the hysteresis loop during the loading and unloading process (see figure 7.11).

The suggested viscoelastic model is capable of reproducing an approximation of real organic material behaviour in CG deformable objects. This may be achieved by incorporating creep / time, relaxation / time and hysteresis graphs derived from experiments on real organic materials. Such graphs may be obtained from existing literature (Alexander 1962), (Yamada 1970), (Gosline 1971), (Koehl 1977), (Dimery *et al.* 1985), (Ker *et al.* 1986), (Fung 1990). The graphs may then be calibrated and modified to conform to specific animation constraints.

Compared to conventional animation techniques, such as key-framing, the introduction of the

viscoelastic model offers a greater degree of freedom. The animator is freed from the burden of having to specify key frames for every minute movement in an organically deforming object. By defining or modifying existing creep, relaxation and hysteresis graphs the user has a fine and high level control over the shape and the temporal behaviour of the deformation. Thus, the viscoelastic model offers extensive control of complex organic material behaviour and may be used to automate the animation process of organically deformable objects.

Compared to physically based simulation models of viscoelasticity, the suggested viscoelastic model has the advantage of efficiency. Terzopoulos and Fleischer (1988) developed a simulation model for idealised viscoelasticity based on mechanical viscoelastic models which consisted of elastic and viscous units (see section 3.4.2, the Maxwell, Voigt and Kelvin models). Their method results in large systems of simultaneous ordinary differential equations. This creates a computational overhead which becomes a prohibitive factor for the implementation of this method in fast interactive animation design applications. The viscoelastic model described in this chapter results in a fast implementation which is general, reusable, extendible, independent of geometric representation and capable of generating realistic looking organic material behaviour.

Colour plates relevant to this chapter can be found in section 7.7. Also, see animation tests 8, 9, 10 and 11 in video tape, time code: from 10:05:38:00 to 10:06:45:00.

7.5.2 Performance of the Viscoelastic Model

The efficiency of the viscoelastic model is based on the precalculation of a spatial gradient for the viscous strain rate per active vertex of an object. A further improvement on efficiency may be achieved by calculating the strain creep and stress relaxation factors once per time step. This adds only a minor computation overhead to the model. The integration time step may be increased to two or more frames in order to further speed up the integration process. This may reduce the smoothness of a deforming object but it enables interaction with viscoelastic objects of considerable size. Reducing the time step to a fraction of a frame improves the smoothness

of the flow of the deformation during the viscoelastic creep of strain. However, setting the time step to values less then half a frame will dramatically decrease the performance of the model.

The following performance tests have been carried out to measure the speed of execution of the viscoelastic model.

7.5.2.1 Performance Test 1

The basic performance test (see section 5.9.2) was modified as follows:

The VOO was equipped with a strain creep function and a spatial gradient of viscous strain rate. The maximum duration of the strain creep process (t_{max} - t_0) was set to 25 frame steps. An offsetting of the tip started at time t_0 and was performed over 25 frame steps while the actual duration of the process was being timed. The test was then repeated by adding extra copies of the same object inside the influence field of the VOO. The results are shown in figure 7.13 (see section 7.7). The same test was performed with precalculation of the spatial gradient of strain rate (blue line) and without precalculation (red line) (see figure 7.13 in section 7.2.4). The green graph in figure 7.13, represents the performance of the plastic model without strain creep and viscous strain rate and it is superimposed for comparison. Interactive manipulation of a viscoelastic object with precalculated viscous strain rate and strain creep presents a drop in performance of approximately between 19% (for 200 vertices) and 23% (for 1200 vertices) when compared with a similar test on a plastic object. If the spatial gradient of the viscous strain rate is not precalculated then there is a further drop in performance of approximately between 10% (for 200 vertices) and 15% (for 1200 vertices). Therefore, precalculation of the spatial viscous gradient, similarly to the precalculation of strain damping elements, considerably improves the performance of the viscoelastic deformation model.

7.5.2.2 Performance Test 2

The previous test was repeated by adding extra VOOs instead of extra objects. The results are shown in figure 7.14 (see section 7.7). The same test was performed with precalculation of the spatial gradient of strain rate (see blue) and without precalculation (see red line in figure 7.14). The green graph in figure 7.14, represents the performance of the plastic model without strain creep and viscous strain rate and it is superimposed for comparison. Interactive manipulation of a viscoelastic object with precalculated viscous strain rate and strain creep presents a drop in performance of approximately between 19% (for one VOO) and 25% (for six VOOs) when compared with a similar test on a plastic object. If the spatial gradient of the viscous strain rate is not precalculated then there is a further drop in performance of approximately 10%. Therefore, precalculation of the spatial viscous gradient, similarly to the precalculation of strain damping elements, considerably improves the performance of the viscoelastic deformation model.

7.5.2.3 Performance Test 3

Performance test 1 was repeated by setting the integration step to half, one and two frame steps. The results are shown in figure 7.15 (see section 7.7). Figure 7.15 shows that an increase of the integration step can dramatically improve the performance of the viscoelastic model (see green line in figure 7.15). However, setting the time step to three or more frame steps may result to noticeable discontinuities in the motion of a deforming object (temporal 3D aliasing). This may sometimes be ignored when working in the interactive mode for the sake of efficiently testing a deformation. It cannot though be ignored during animation rendering where smooth deformation is required. Reducing the time step to a value less than one frame step can cause a rapid deterioration of the performance of the model (see red line in figure 7.15). A small integration step however can contribute to smoother flow of deformation during the viscoelastic creep of strain.

7.6 Summary and Conclusion

In this chapter, the deformation model has been extended to incorporate the property of viscoelasticity. The property of viscoelasticity enhances the realistic look of deformable objects and enables the development of animated characters based on living vertebrate and invertebrate creatures. The objective was to provide an efficient and constrainable mechanism for generating realistic organic material behaviour.

The viscoelastic model provides a mechanism for the implementation of the three characteristic viscoelastic features of strain creep, stress relaxation and stress / strain hysteresis. A spatial gradient of strain rate and a viscosity coefficient are used to generate a viscous effect in a deformable object. A strain creep graph, a strain creep coefficient and a strain creep duration provide a temporal constraint for the development of viscoelastic creeping strain inside the volume of a 3D object. A stress relaxation graph is used to "relax" the internal elastic stresses over a period of time. Finally, an hysteresis graph is used to describe the hysteresis loop during the loading and unloading process. The user is given the freedom to customise the viscoelastic behaviour of an object by manipulating these graphs and coefficients. An object may be made to respond like a thin runny fluid or a thick sticky fluid or, at the other extreme, as an elastoplastic material with no viscosity at all.

The original concept introduced in this chapter is the combination of viscoelastic features based on user defined graphs with an efficient local deformation model.

The following chapter will extend the deformation model to incorporate spatial and temporal constraints for VOOs.

7.7 Colour Plates

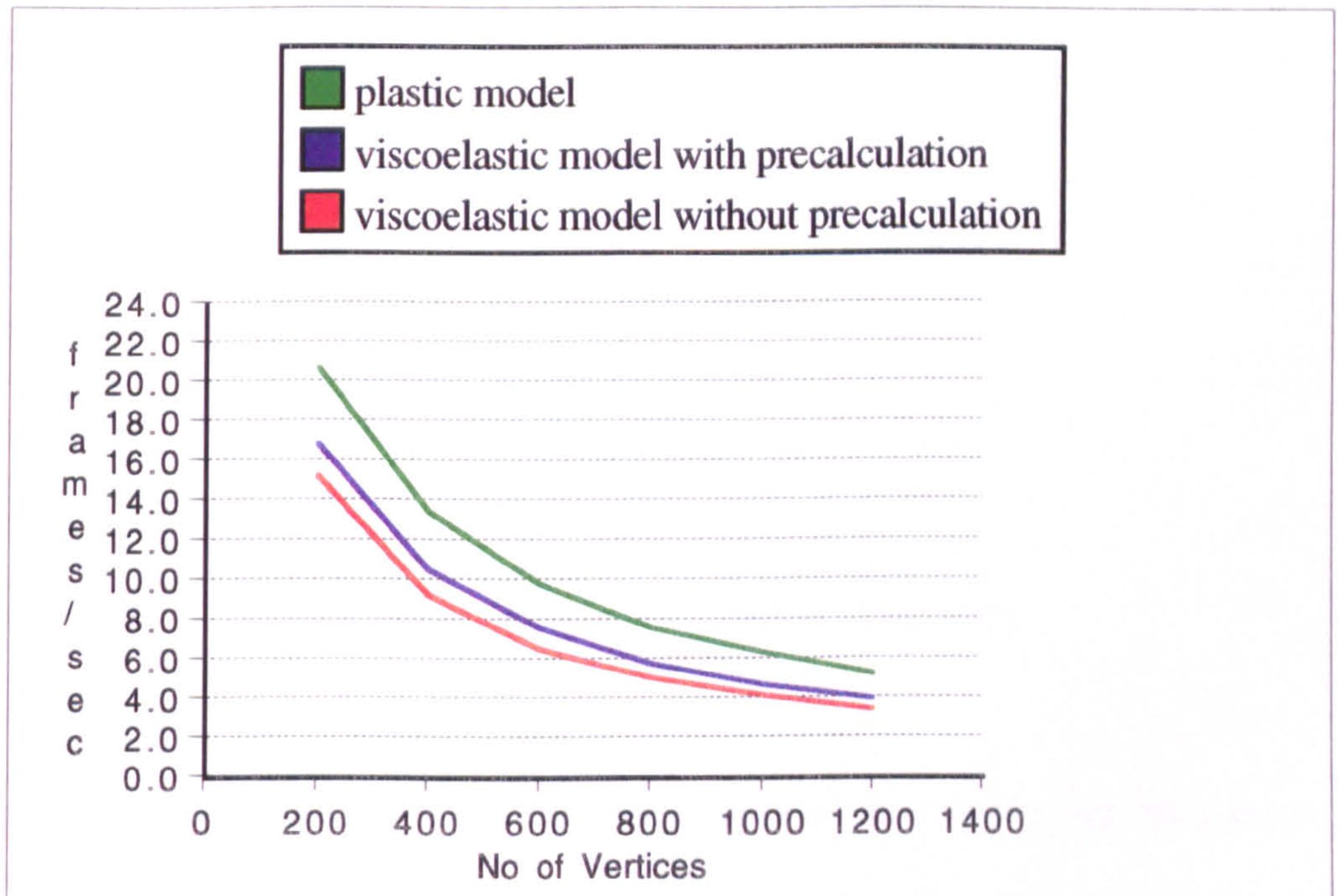


Figure 7.13 Graph of speed of viscoelastic deformation over the number of vertices.

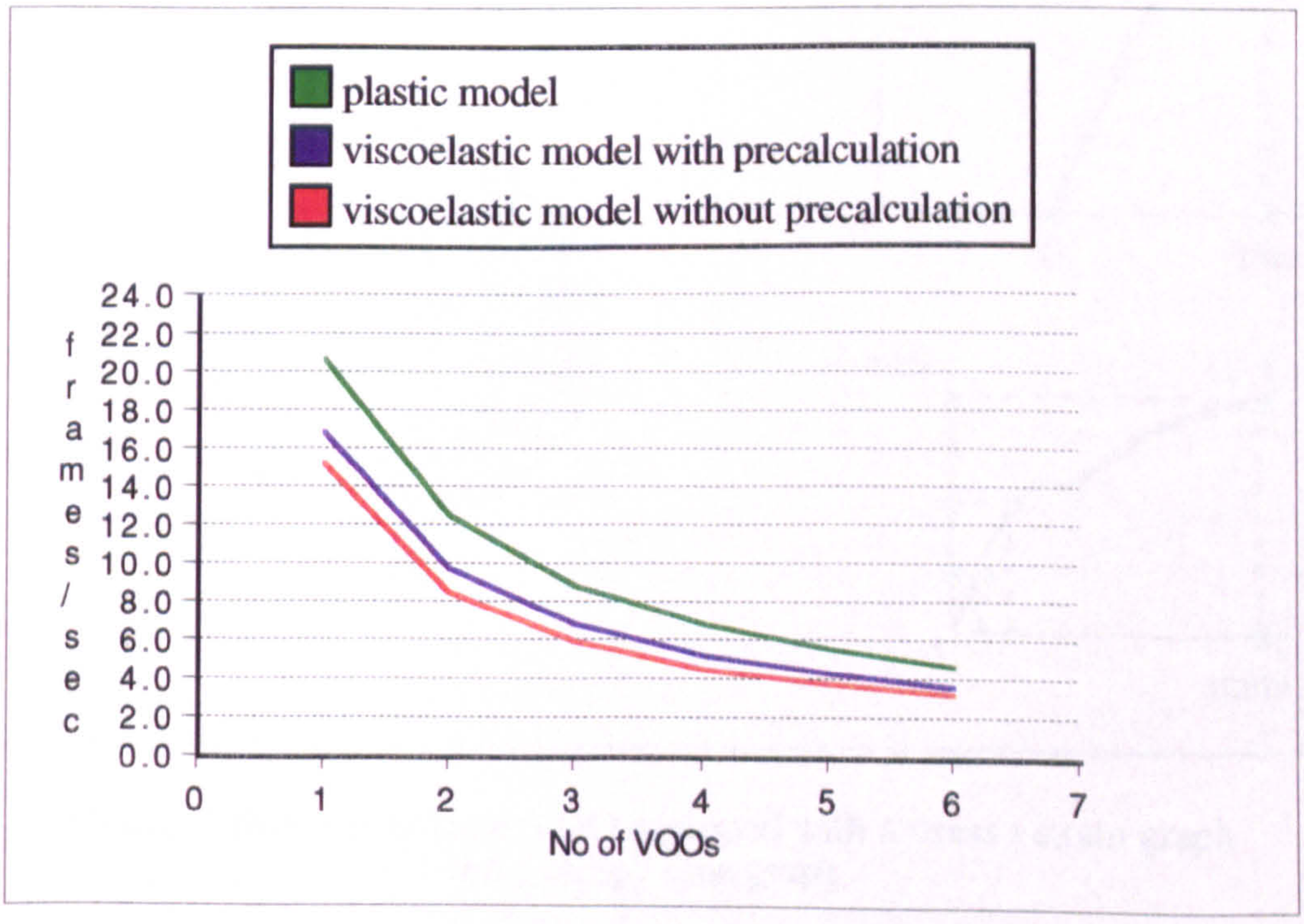


Figure 7.14 Graph of speed of viscoelastic deformation over the number of VOOs.

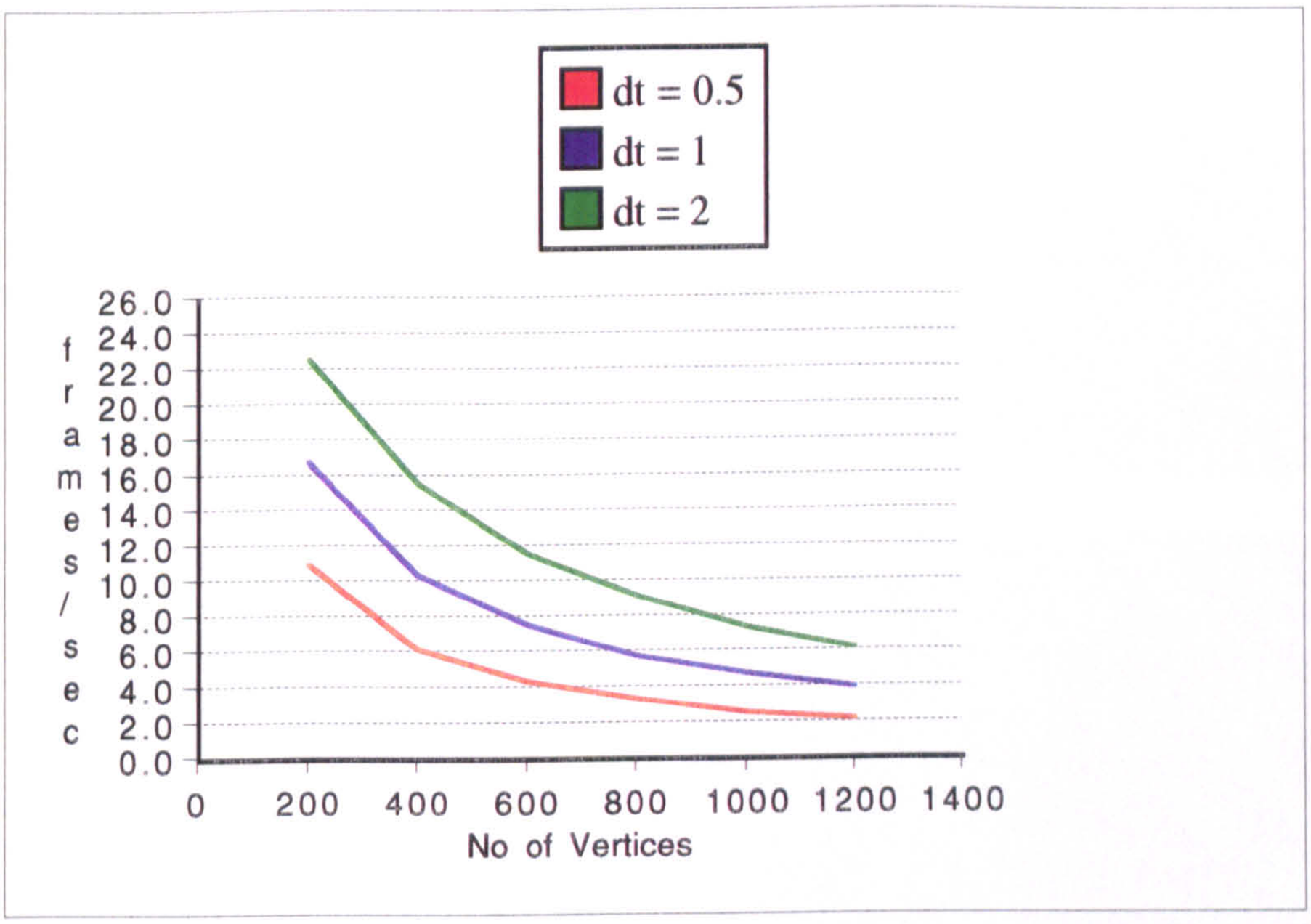


Figure 7.15 Graph of speed of viscoelastic deformation with varying integration step.

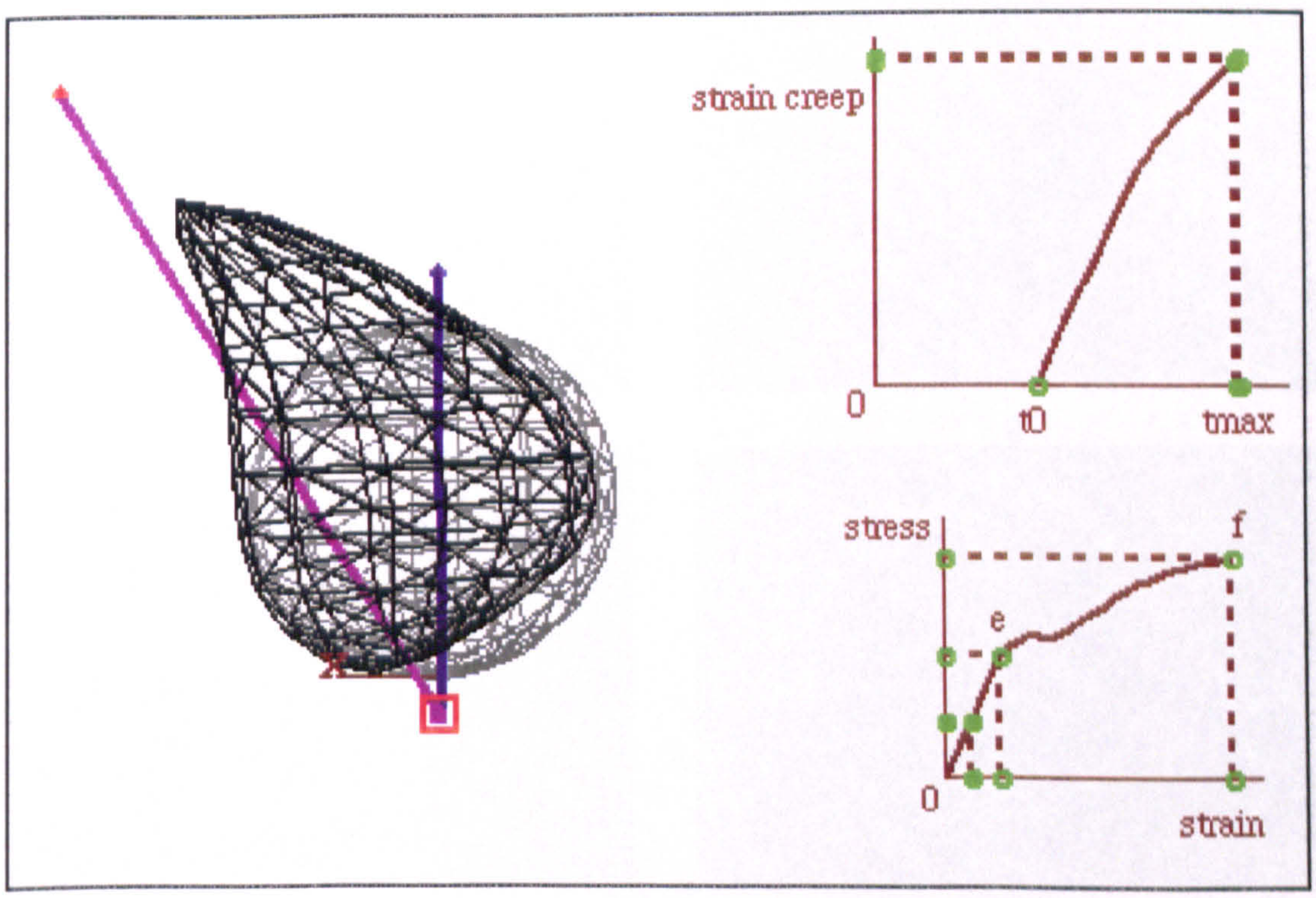
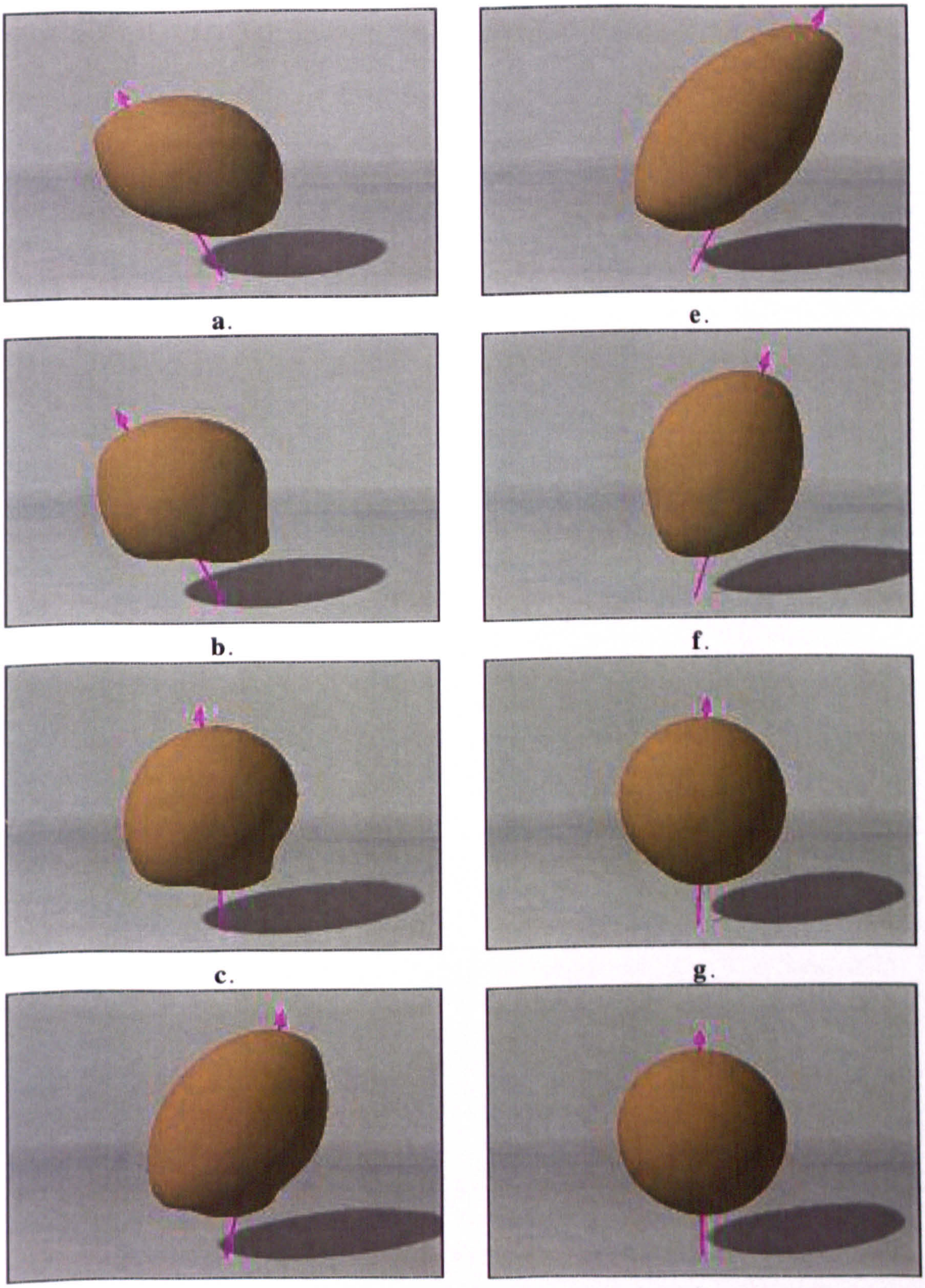


Figure 7.16 A viscoelastic VOO equipped with a stress / strain graph and a creep / time graph.



a. **e.**

b. **f.**

c. **g.**

d. **h.**

Figure 7.17 A viscoelastic object exhibits strain creep
(see animation test 8 in video tape, time code: from 10:05:38:00).

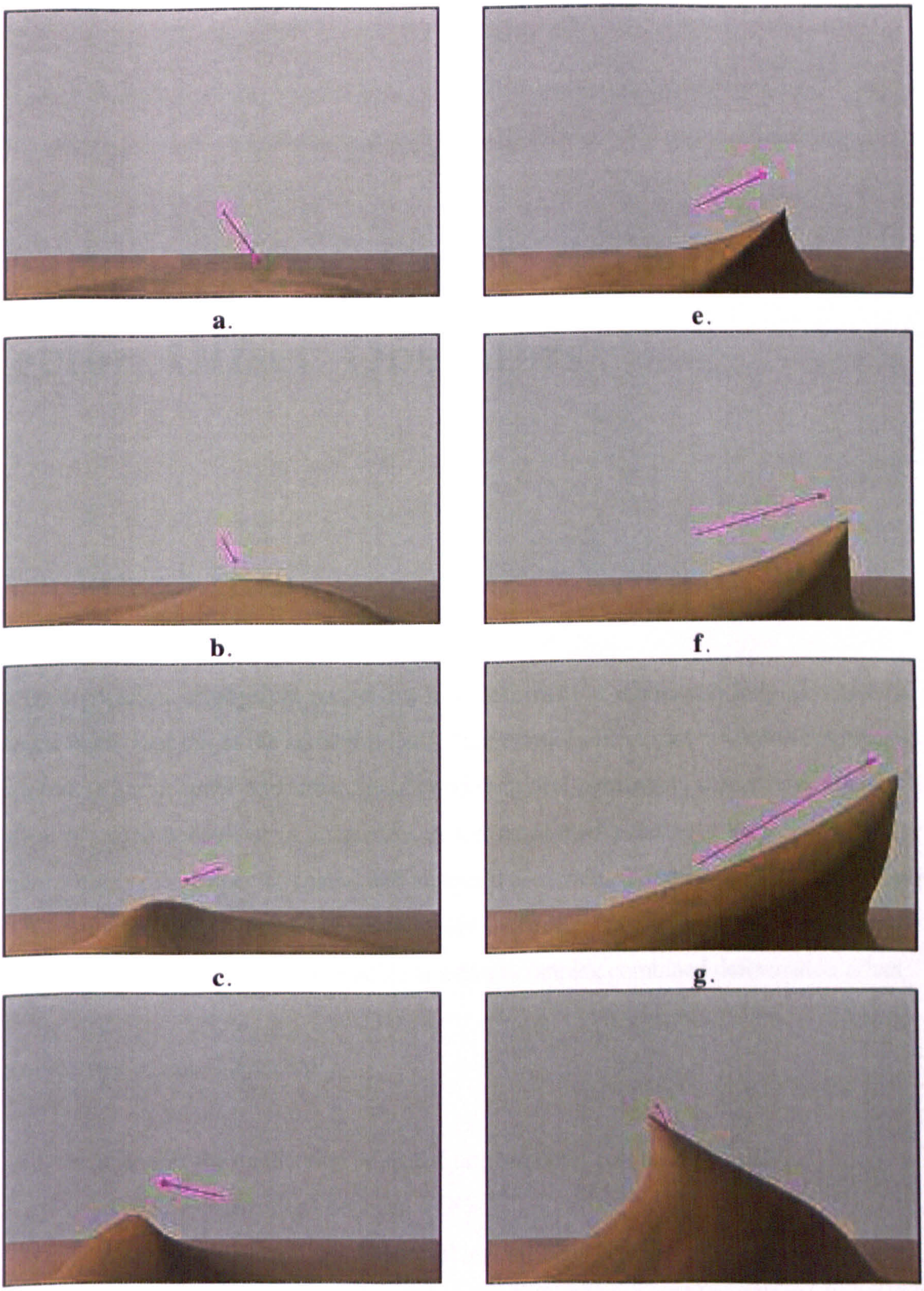


Figure 7.18 A viscoelastic object exhibits strain creep
(see animation test 11 in video tape, time code: from 10:06:27:00).

8

Spatiotemporal Constraints for Vector Offset Operators

8.1 Introduction

The incorporation of physical properties in a deformation model can help generate natural material behaviour but at the same time introduces major difficulties in controlling the state of the deformation in space and time. Spatial and temporal constraints are necessary, in order to confine physical behaviour in animation. Apart from manipulating a VOO interactively, an animator may wish to specify spatial and temporal constraints which will control the motion of a VOO, from the start to the end of an animated sequence. Also, several VOOs may need to be constrained to each other in order to achieve a more complex combined deformation effect.

The objectives of this chapter are:

- to enable the application of spatial and temporal constraints to VOOs,
- to enable connections between VOOs,
- to enable interaction with a general and efficient constraint model.

In the suggested approach, the focus is concentrated on fast, effective and intuitive ways of dynamic constraint specification that are well suited to the animation practice.

8.2 Motion Paths

Often in animation objects must follow predetermined spatial paths whose shape and dimensions are dictated by other objects or other animation parameters. Spatial paths may be accompanied by simple temporal constraints such as start and end times. Occasionally, certain motion characteristics of an object are also predetermined. Velocity and acceleration or deceleration are often required to assume certain values at certain times during the traversal of a path. A motion path is a combination of a spatial path with motion characteristics which are necessary for the description of the desired motion of an object along the path.

8.2.1 A Mass Point on a Motion Path

A point of constant mass, m, may be forced to travel on an existing spatial path under the influence of a force function, $\mathbf{F}(t)$, provided that the direction of the force vector remains tangential to the curve of the path (see figure 8.1).

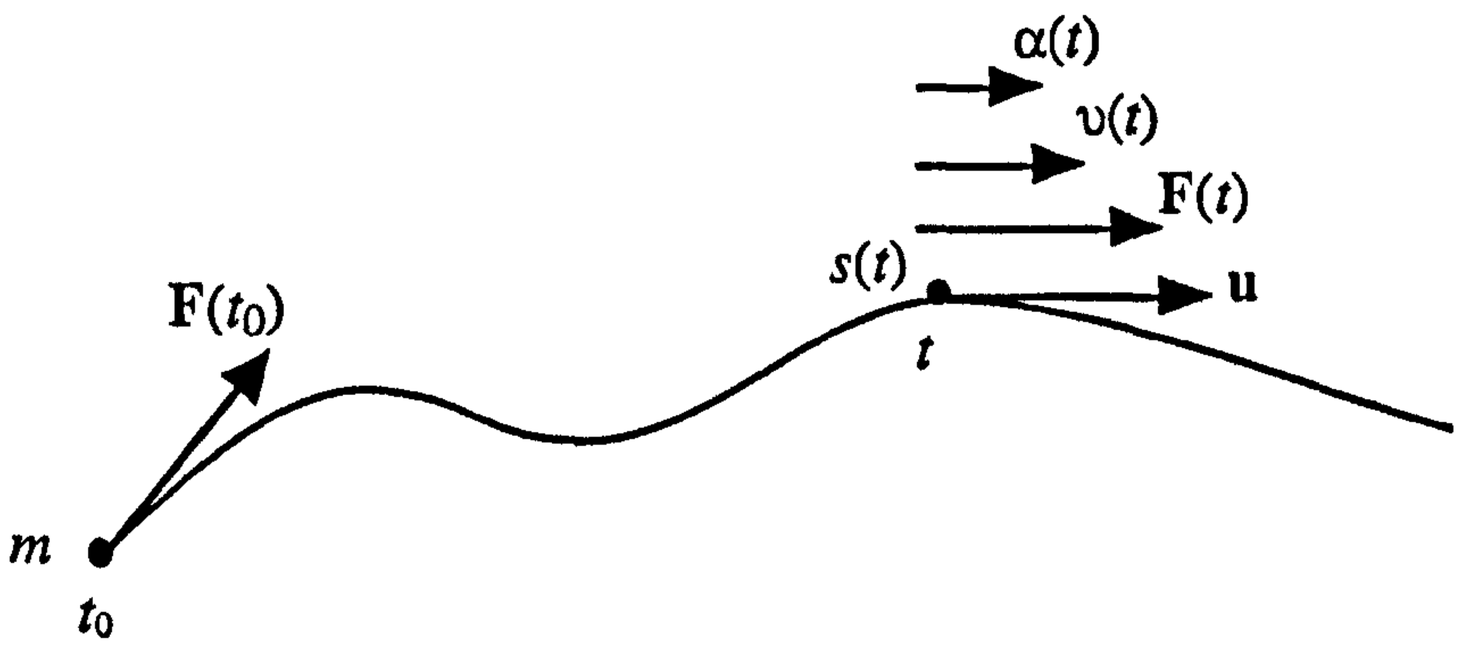


Figure 8.1 A mass point on a motion path.

In figure 8.1, a mass point starts at time t_0 at the beginning of the path. At a time t, the distance travelled on the path is $s(t)$. The velocity vector $\upsilon(t)$ of the mass point at time t, is given by:

$$\upsilon(t) = \frac{ds(t)}{dt} \cdot \mathbf{u} \qquad [8.1]$$

where **u** is a unit vector tangent to the curve of the path at the position of the mass point. The acceleration vector $\alpha(t)$, at time t, is given by:

$$\alpha(t) = \frac{d\upsilon(t)}{dt} \qquad [8.2]$$

According to Newton's second law, the force vector acting on the mass point may be derived as the first time derivative of the momentum of the mass point

$$\mathbf{F}(t) = \frac{d\mathbf{M}(t)}{dt} \quad \Rightarrow \qquad [8.3]$$

$$\mathbf{F}(t) = \frac{d(m \cdot \upsilon(t))}{dt} \quad \Rightarrow \qquad [8.4]$$

$$\mathbf{F}(t) = m \cdot \alpha(t) \qquad [8.5]$$

In the above equations, it is assumed that $\mathbf{F}(t)$ is the only force acting on the mass point and that there is no gravity, friction, wind resistance, etc.

8.2.2 The Momentum / Time Graph

A momentum / time graph may contribute to the specification of a force function acting on a mass point which is constrained to travel on a motion path. At every time t, between start time t_0 and end time t_1, the graph may provide the magnitude of the momentum of the mass point (see figure 8.2).

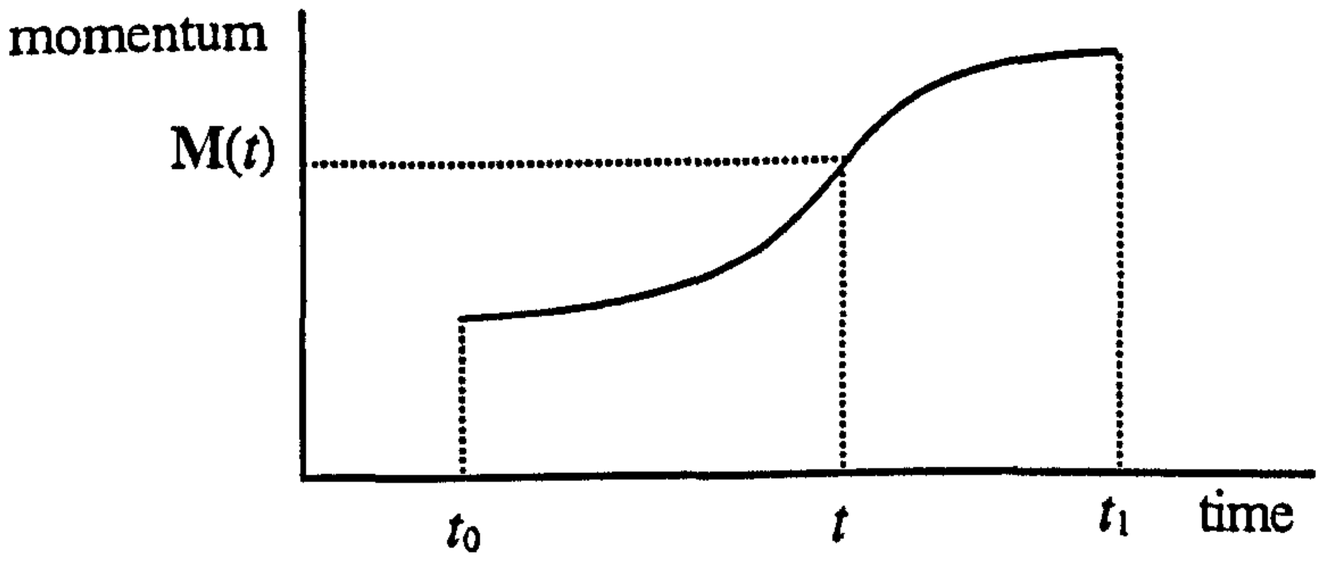


Figure 8.2 A momentum / time graph.

The tangent on the curve of the graph will provide the magnitude of the force acting on the mass point (see figure 8.3).

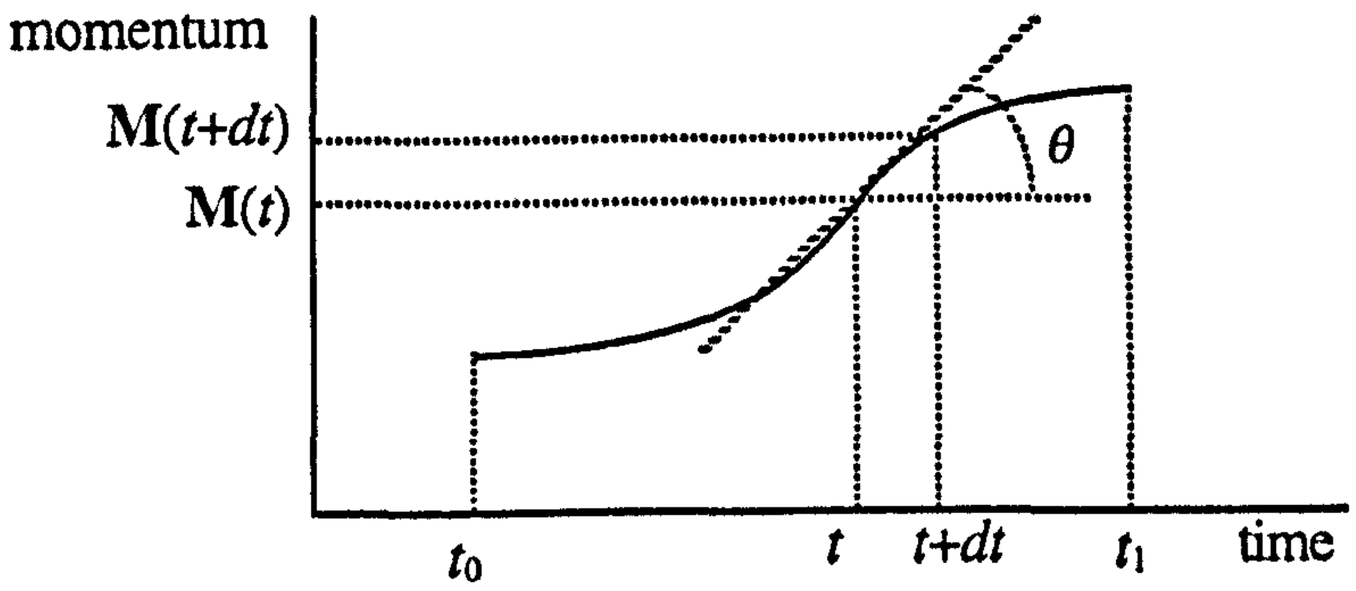


Figure 8.3 Extracting force from a momentum graph.

Assuming that for a short time step, *dt*, after time *t*, the change of momentum is linear, the force magnitude may be obtained as the first time derivative of the momentum (see equation [8.3]).

$$F(t) = \frac{dM(t)}{dt} = \tan(\theta) \qquad [8.6]$$

So, between the start and end times the momentum graph may help generate a force function acting on the mass point. Therefore, the dynamic motion of a mass point on a path may be specified via a user defined momentum / time graph.

8.2.3 Implementing Motion Paths

A user defined motion path may consist of a spatial path, a mass, m, initial velocity, $v(t_0)$, initial position, $s(t_0)$, start time t_0, end time t_1 and a momentum / time graph (see figure 8.4a and b).

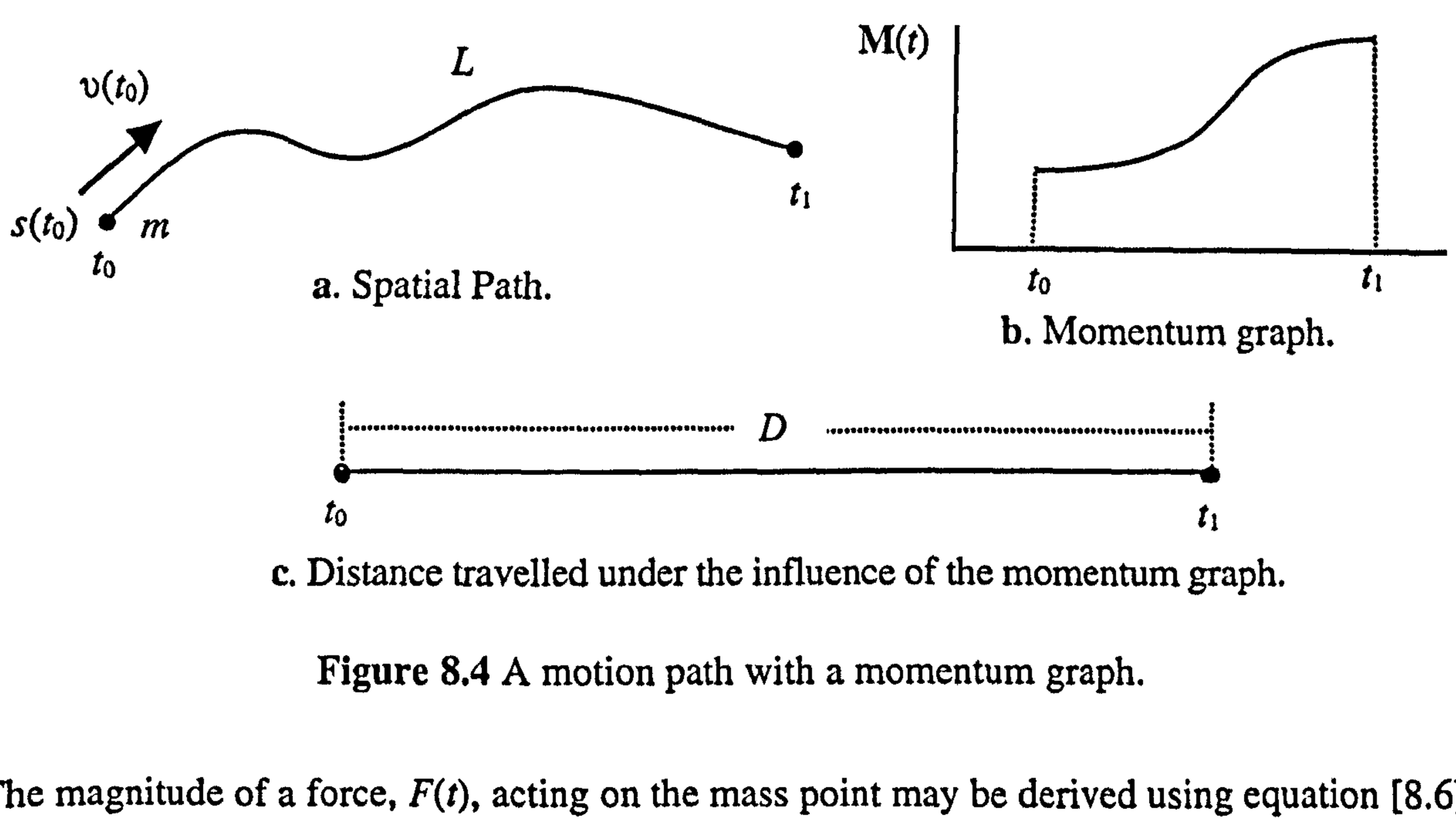


Figure 8.4 A motion path with a momentum graph.

The magnitude of a force, $F(t)$, acting on the mass point may be derived using equation [8.6] (see section 8.2.2). The magnitude of acceleration, $\alpha(t)$, at time t, may then be derived from equation [8.5], as follows:

$$\alpha(t) = \frac{F(t)}{m} \qquad [8.7]$$

The magnitude of velocity, $v(t)$, at time t, may be calculated using equation [8.2] as follows:

$$v(t) = v(t_0) + \int_{t_0}^{t} \alpha(t)dt \qquad [8.8]$$

where $v(t_0)$ represents the initial velocity of the mass point at time t_0. The distance, $s(t)$, travelled on the path from t_0 to t, may be calculated using equations [8.1] and [8.8] as follows:

$$s(t) = s(t_0) + v(t_0) + \int_{t_0}^{t}\int \alpha(t)dt^2 \qquad [8.9]$$

where $s(t_0)$ represents the user defined initial position on the path and is usually set to the beginning of the path.

The integration in equation [8.9] may generate a travelled distance $s(t_1)$ which is different from the total length, L, of the spatial path. So, at the end time t_1, the mass point will not necessarily appear at the end of the path. To ensure that the mass point does reach the end of the path at time t_1, a further step must be taken. The integration in equation [8.9] may be precalculated from t_0 to t_1, in order to find the total distance D that the mass point would have travelled under the influence of the user defined momentum graph (see figure 8.4c). At every time t, the travelled distance on the path and the velocity must be scaled by a scaling factor (L / D) as follows:

$$D = s(t_0) + v(t_0) + \int_{t_0}^{t_1}\int \alpha(t)dt^2 \qquad [8.10]$$

$$s(t) = \frac{L}{D}\left(s(t_0) + v(t_0) + \int_{t_0}^{t}\int \alpha(t)dt^2 \right) \qquad [8.11]$$

$$v(t) = \frac{L}{D}\left(v(t_0) + \int_{t_0}^{t} \alpha(t)dt \right) \qquad [8.12]$$

8.2.4 A Numerical Solution

The differential equations of motion (see equations [8.1], [8.2], [8.11] and [8.12]) may be solved using the Runge-Kutta numerical approximation method (Mylonas 1978) (Press *et al.* 1986). A pseudocode which describes this method follows:

$t = t_0 \quad , \quad \upsilon(t) = \upsilon_0 \quad , \quad s(t) = s_0$

repeat

- step a:

$$t_a = t$$

$$\alpha_a = \alpha(t_a) = \frac{F(t_a)}{m} \qquad \text{(extract } F(t_a) \text{ from momentum graph)}$$

$$\upsilon_a = \upsilon(t) + \frac{1}{2}\alpha_a \Delta t$$

- step b:

$$t_b = t + \frac{1}{2}\Delta t$$

$$\alpha_b = \alpha(t_b) = \frac{F(t_b)}{m} \qquad \text{(extract } F(t_b) \text{ from momentum graph)}$$

$$\upsilon_b = \upsilon(t) + \frac{1}{2}\alpha_b \Delta t$$

- step c:

$$t_c = t_b$$

$$\alpha_c = \alpha_b$$

$$\upsilon_c = \upsilon(t) + \alpha_c \Delta t$$

- step d:

$$t_d = t + \Delta t$$

$$\alpha_d = \alpha(t_d) = \frac{F(t_d)}{m} \qquad \text{(extract } F(t_d) \text{ from momentum graph)}$$

$$\upsilon(t) = \upsilon(t) + \left(\alpha_a + 2\alpha_b + 2\alpha_c + \alpha_d\right)\frac{\Delta t}{6}$$

$$s(t) = s(t) + \left(\upsilon(t) + 2\upsilon_a + 2\upsilon_b + \upsilon_c\right)\frac{\Delta t}{6}$$

$t = t + \Delta t$

until $t > t_{cur}$ **[8.13]**

The integration in pseudocode [8.13] starts at time t_0 when the mass point lies at position s_0 on the path and has initial velocity υ_0. The integration proceeds until a current time t_{cur} ($t_{cur} \leq t_1$). At t_{cur} the mass point lies at a position $s(t_{cur})$ on the path curve and has a velocity $\upsilon(t_{cur})$. The approximation of the integration is achieved using four intermediate steps and by averaging the results. This integration method is likely to result in numerical instability. The current solution is good enough for demonstration purposes but for a commercial implementation a more robust method could be implemented (see Taylor and Euler methods (Press *et al.* 1986)).

The position and velocity must then be scaled by a scale factor of (L / D) (see section 8.2.3).

$$s(t_{cur}) = \frac{L}{D} s(t) \qquad [8.12]$$

$$v(t_{cur}) = \frac{L}{D} v(t) \qquad [8.13]$$

where L represents the total length of the path curve. Length D must be precalculated from equation [8.10] using the same approximation method. Equation [8.12] generates a length $s(t)$, which is then mapped on the path curve to define the exact position of the mass point on the curve. The time step, Δt, is usually set to one frame step. Reducing the time step to values less than one frame step increases the accuracy of the integration and hence the smoothness of motion. However, this causes a deterioration of the performance of the dynamic constraint model. On the other hand, an increase of the time step to two or more frame steps would improve the performance of the model but it would, also, reduce the accuracy of the model.

8.2.5 Obtaining Motion Paths

Motion paths may be derived from a variety of sources. There is an abundance of reference material on motion of living creatures (Muybridge 1899, Burton 1990, Ward-Smith 1984, Slijper 1961, Gray 1968, etc.). Photographic sequences may be used to plot subsequent positions of parts of an animal's body. These positions may then be connected to form continuous spatial paths. Walking, running, galloping, flying and swimming motion path cycles may be derived in this way (see figures 8.5, 8.6, 8.7 and 8.8).

Recent developments in 3D motion capture technology have allowed the direct capture of motion paths of specific moving joints on actors performing in real time (Witkin and Popoviç 1995) (see figure 8.9). Motion capture systems may be used to produce realistic human figure animation with more ease and efficiency than traditional key-framing techniques. Motion capture may also be used to create libraries of reusable motion. Motion capture techniques are

best suited for obtaining motion paths for human or generally anthropomorphised animated characters.

However, observation of natural motion has always been the most commonly used technique among animators who attempt to generate animated characters with life like motion behaviour.

Existing motion paths derived from library data, motion capture or, simply from observation, may be modified in order to conform to specific requirements and may be reused in different animation situations. Customisation and fine tuning of existing paths is widely used in animation. For example, a standard bipedal walk cycle may be customised and embedded in animated characters of any shape and size. A standard wing flap cycle may be adapted for a variety of flying characters, etc.

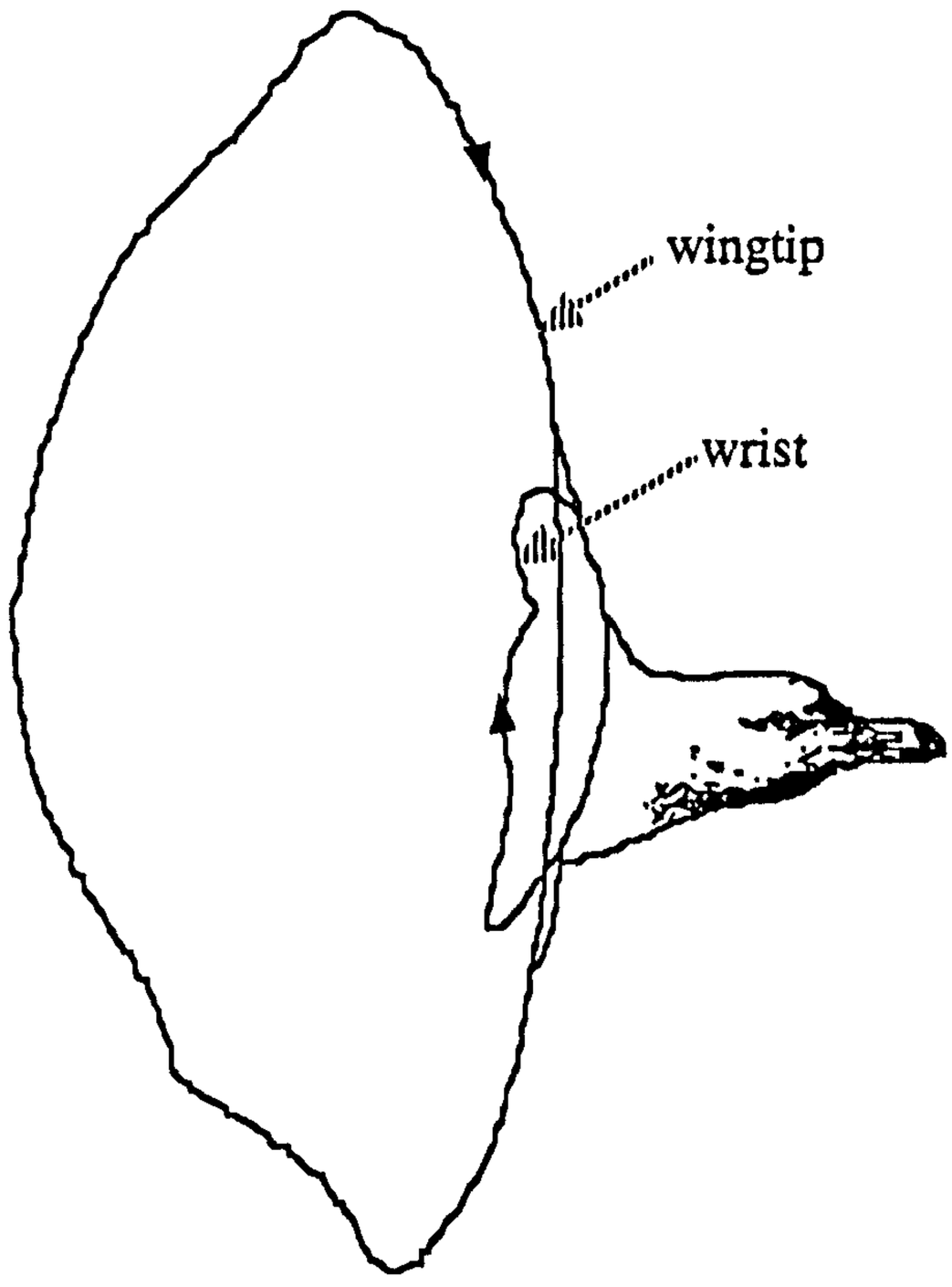


Figure 8.5 Motion paths of wingtip and wrist of a seagull (Burton 1990).

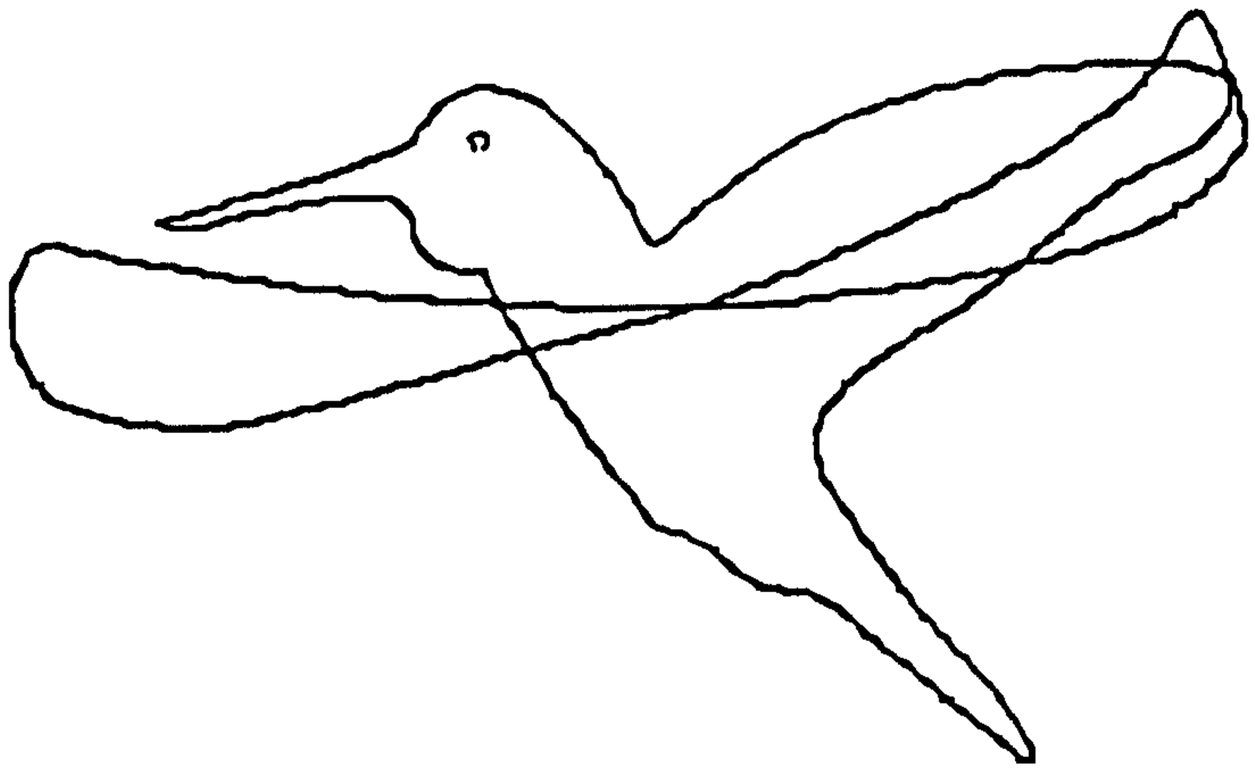


Figure 8.6 Motion path of wingtip of a hummingbird (Ward-Smith 1984).

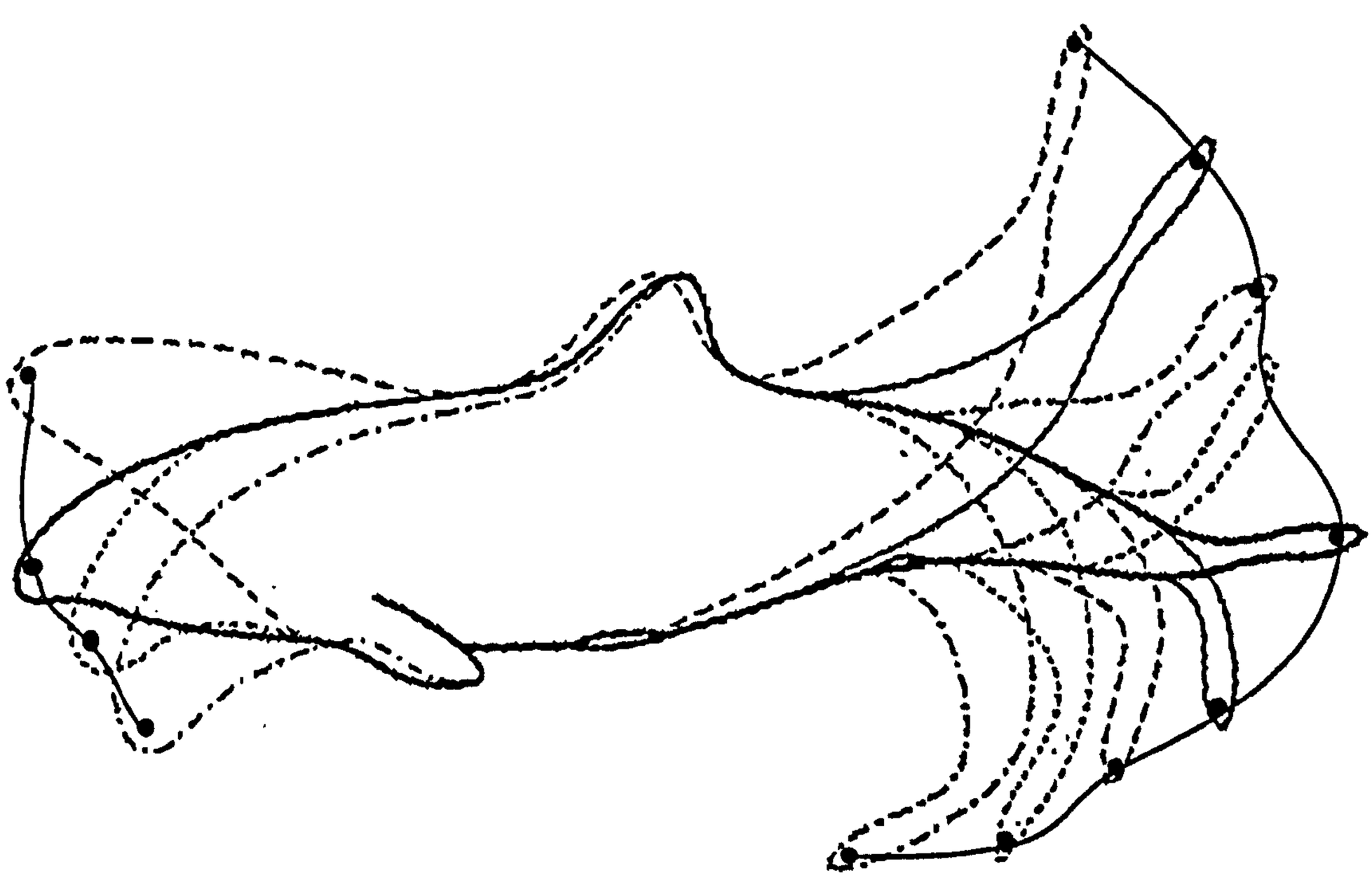
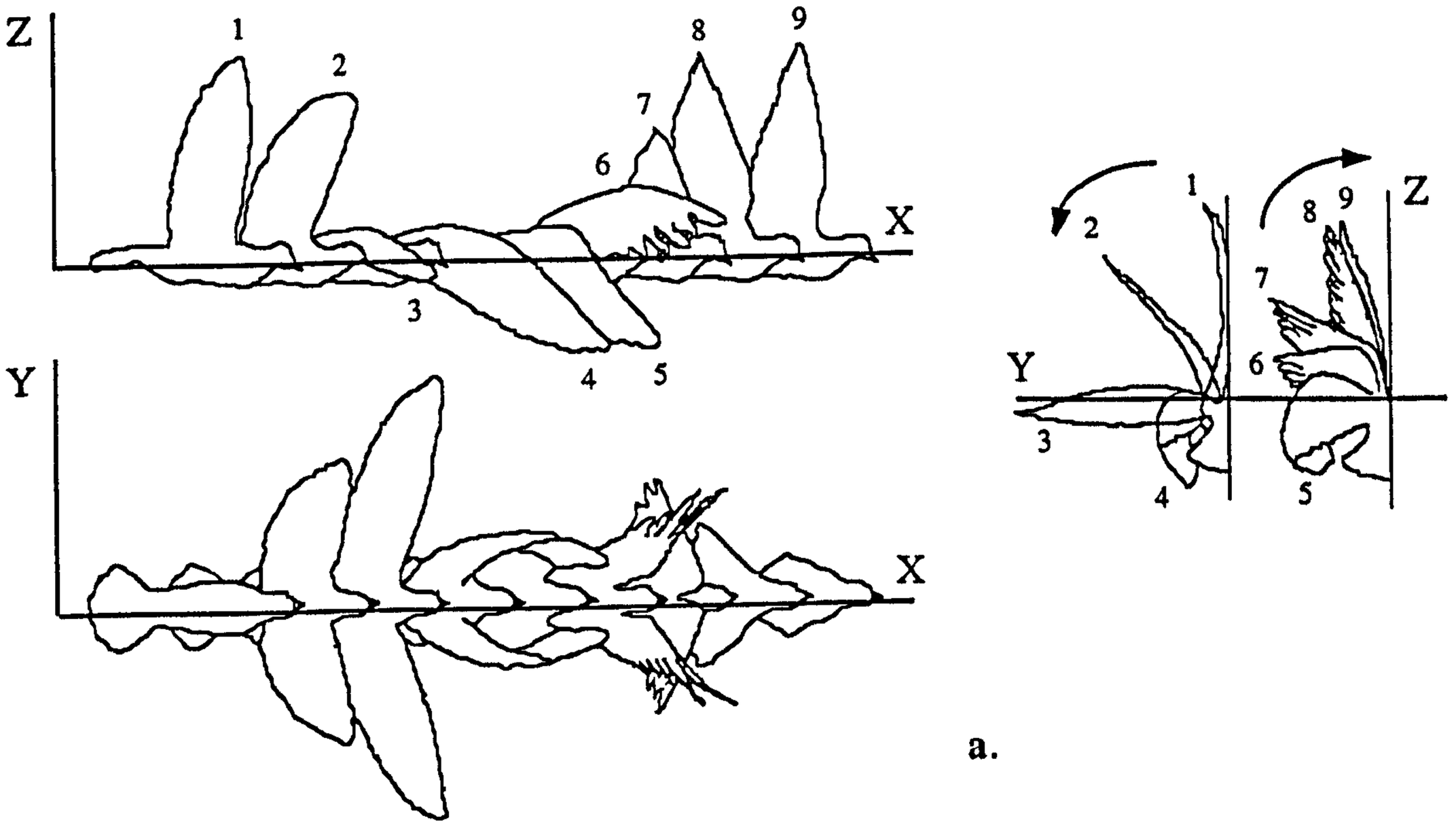
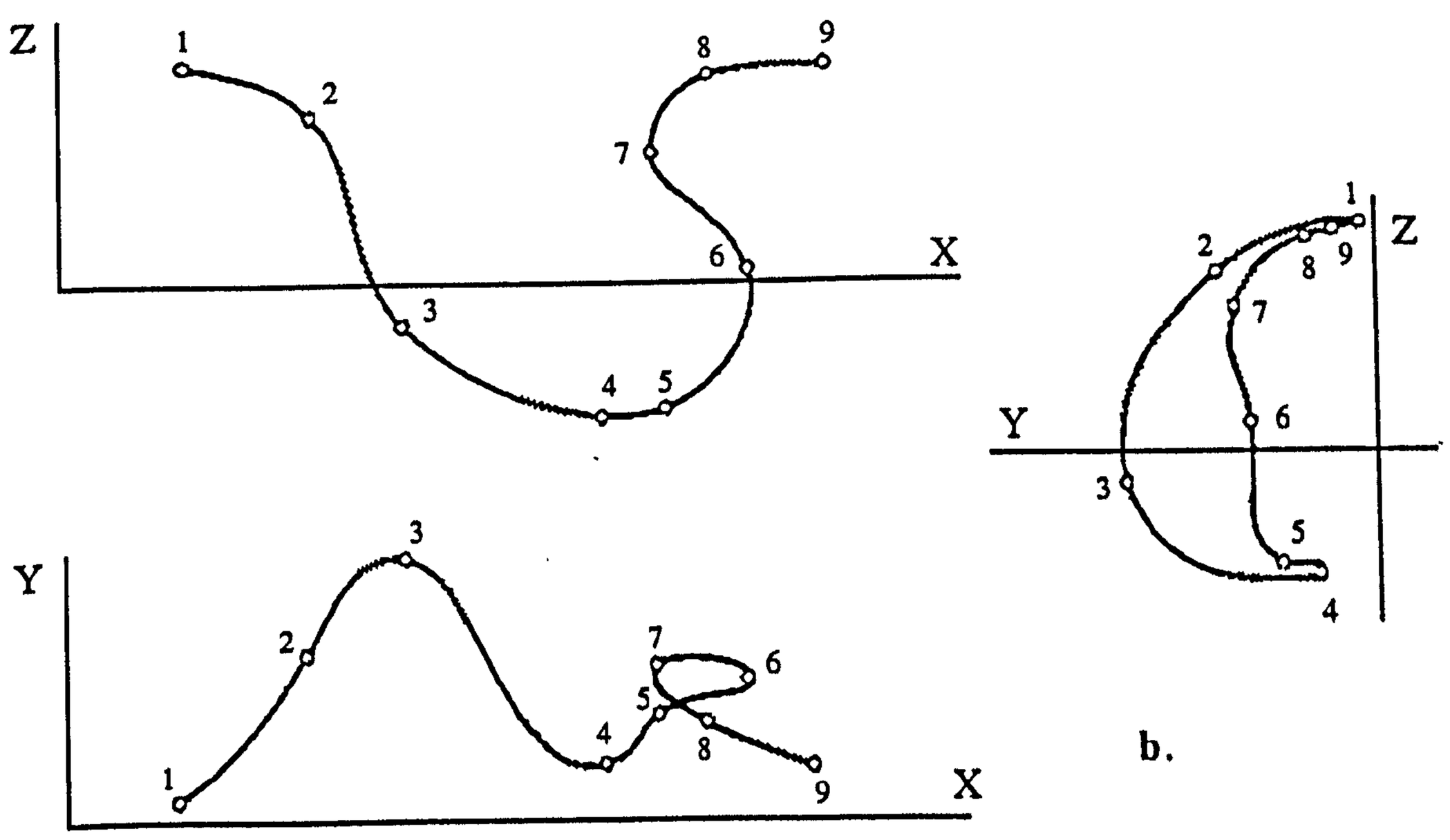


Figure 8.7 Motion paths of body of a common dolphin (Slijper 1961).



a.



b.

Figure 8.8 a. Flight cycle of a pigeon.
b. Motion paths of a flying pigeon (Gray 1968).

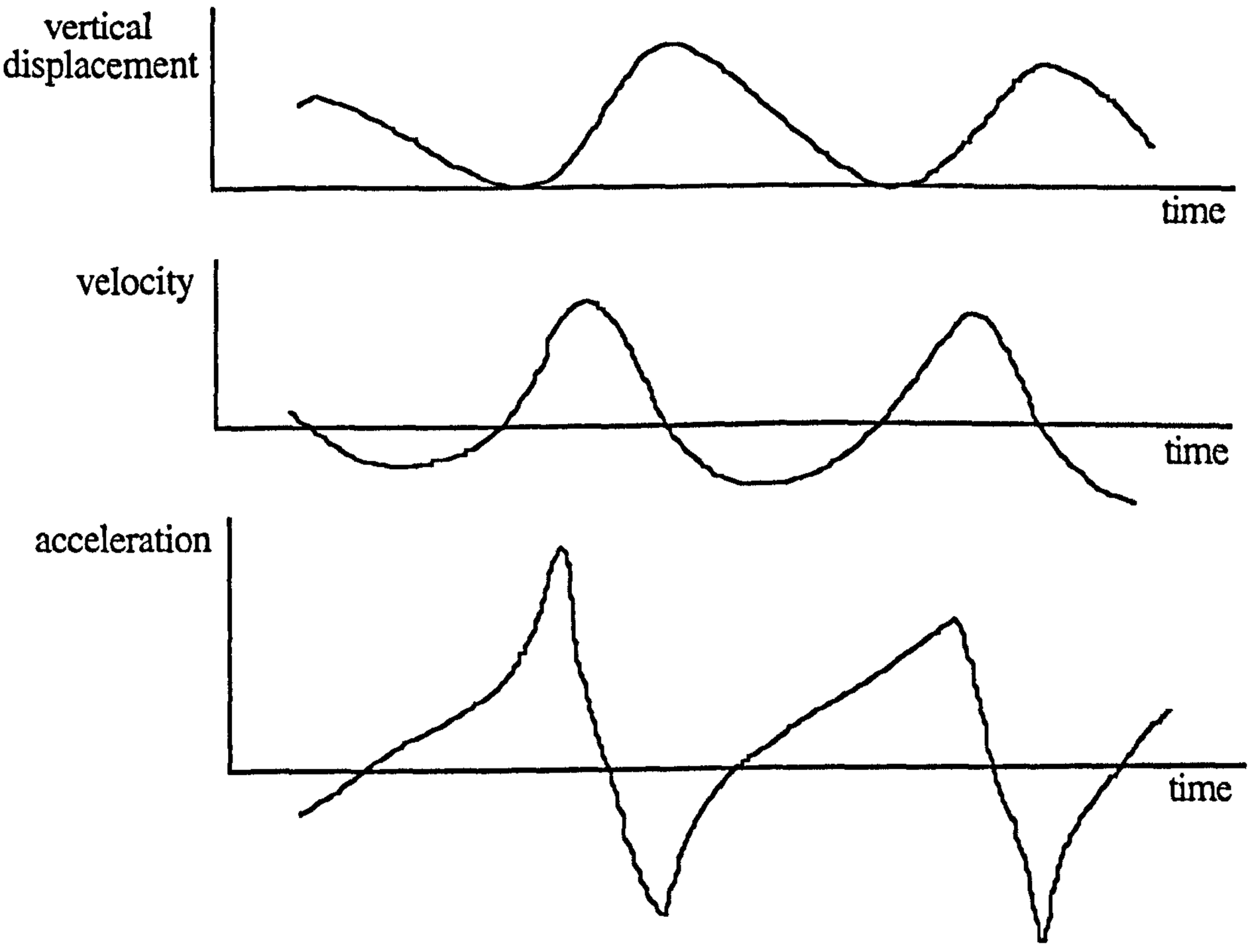
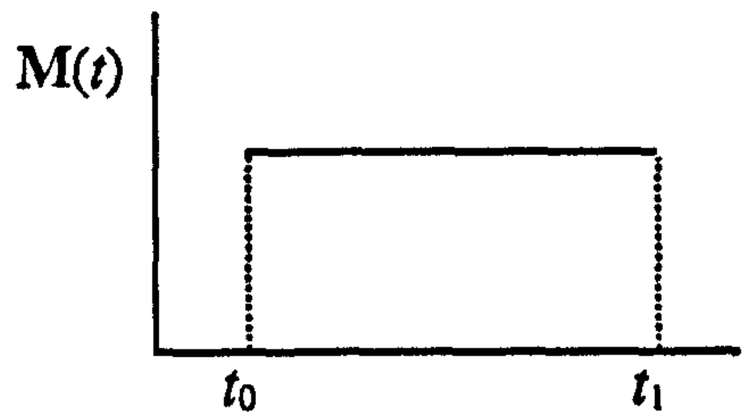
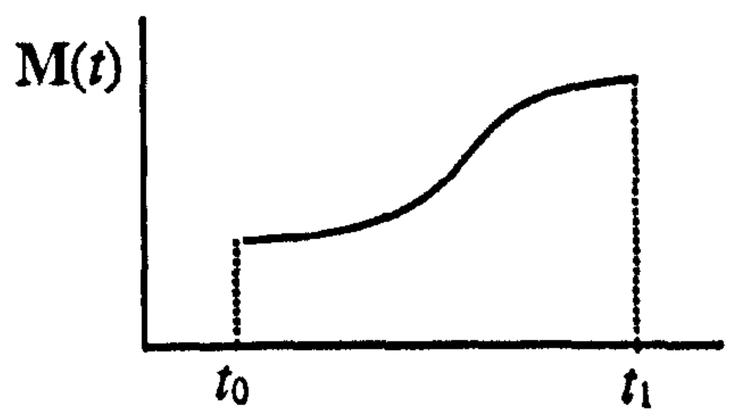


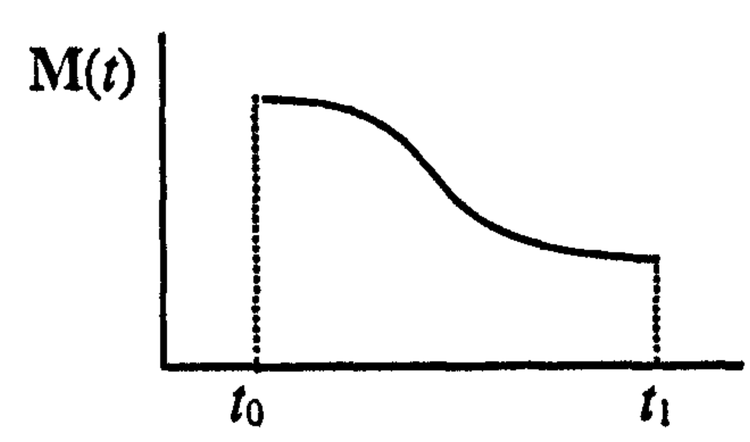
Figure 8.9 Motion captured displacement curve and derived velocity and acceleration curves of human heel during walking.



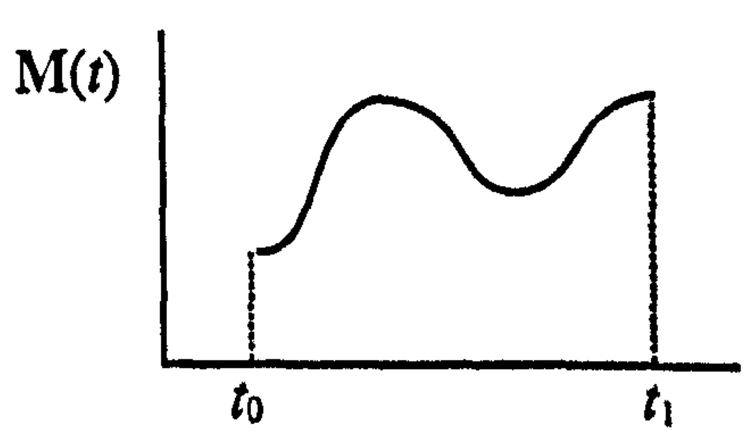
a. Constant velocity.



b. Smooth acceleration.



c. Smooth deceleration.



d. Momentum graph with accelerations and decelerations.

Figure 8.10 Various momentum / time graphs.

Momentum graphs may also be customised and fine tuned so they fit to the specific requirements of an animated sequence. Start and end times and acceleration profile may be adjusted by directly manipulating a momentum graph. Increasing the tangent of a momentum graph generates higher acceleration for the same mass (see figure 8.10b), while decreasing the tangent generates deceleration (see figure 8.10c). Velocity may be kept constant by setting the momentum graph tangent parallel to the time axis (see figure 8.10a). Motion paths allow finely constrained and efficient dynamic control suitable for animation. VOOs equipped with motion paths offer a combination of dynamic constraints with elastoplastic and viscoelastic material behaviour on deformable objects.

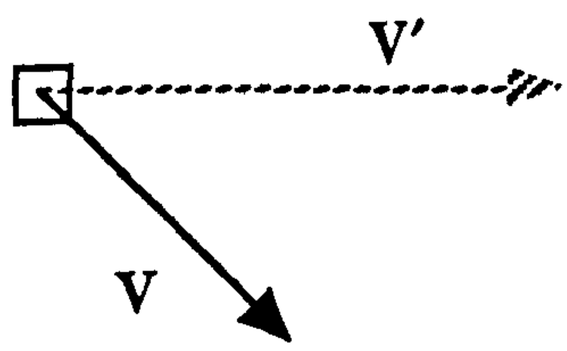
8.3 Constraining VOOs

In previous chapters, a VOO was introduced as a deformation tool equipped with mechanisms for elastoplastic and viscoelastic material behaviour. A VOO is defined as a fixed vector with position, magnitude and direction in 3D space. The field of influence of a VOO describes how the deformation is distributed in the space around a VOO. Additionally, in animated deformation, VOOs may also play a vital role as carriers of spatial and temporal constraints as well as of deformation parameters. To this end, VOOs will be equipped with spatial and temporal constraints.

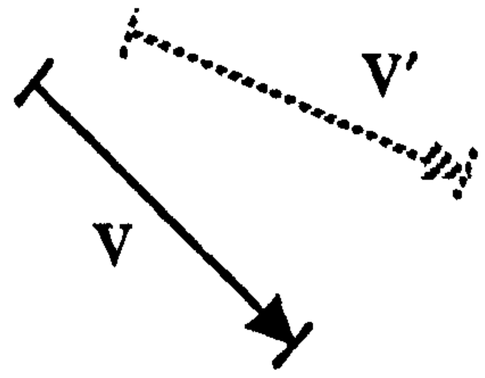
VOOs may be constrained in space and time in a variety of ways. Simple spatial constraints may be applied to the position, length, direction and aiming point of a VOO. A combination of spatial and temporal constraints may be applied by attaching the head or tip of a VOO on a motion path. Two or more VOOs may be connected using an arthron (joint). Sets of VOOs and arthra may be used to construct skeletons and plexi (plural of plexus).

8.3.1 Constraining a Single VOO

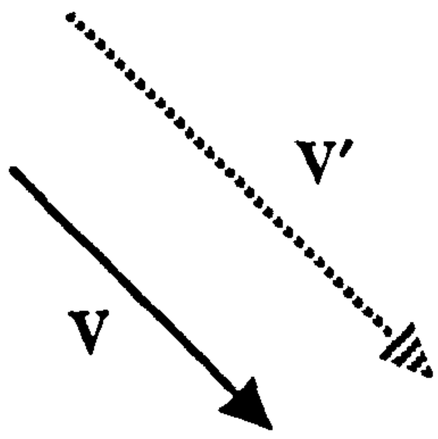
A VOO may assume a constant position in space by attaching its head and or its tip at a fixed point in space. In figure 8.11a, **V** represents the initial state of a VOO and **V′** represents the transformed state. The head of **V** is attached to a fixed position in space.



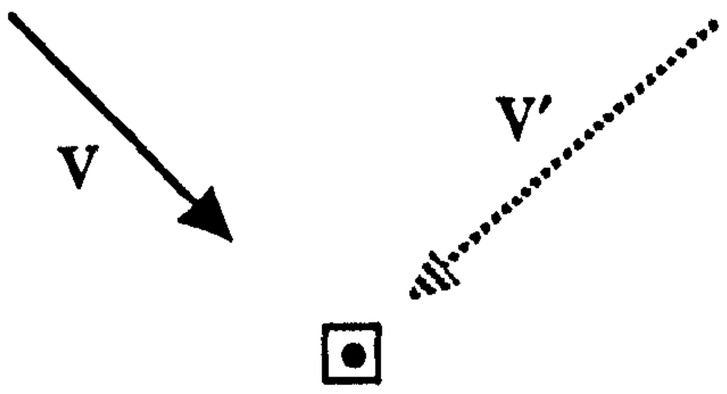
a. Constrained position of VOO head.



b. Constrained length of VOO.



c. Constrained direction of VOO.



d. Constrained aiming of VOO.

Figure 8.11 Four types of constraints for single VOOs.

A VOO may be assigned constant length. The head and or tip may be moved freely but the resulting VOO only receives a combination of translation and rotation thus keeping its length constant (see figure 8.11b). Alternatively, a VOO may be assigned constant direction. The head and or tip may be moved freely but the resulting VOO only receives a combination of translation and scaling thus keeping its direction constant (see figure 8.11c). The aiming of a VOO may be constrained to a fixed point in space. The head and or tip may be moved freely but the resulting VOO receives a combination of translation, rotation and scaling, such that the VOO continues to aim at the fixed point (see figure 8.11d).

8.3.2 Constraining VOOs on Motion Paths

The head, the tip or the aiming point of a VOO may each be attached to a motion path. By doing so, it is assumed that mass is concentrated on the head, the tip or the aiming point of a VOO. Therefore, instead of mass being distributed over the entire volume of a deformable object it only appears concentrated on the ends of the VOOs that control its deformation. By forcing the end mass points of a VOO to move on motion paths a VOO becomes a carrier of dynamic constraints. The effect of a force being applied to a mass point is then distributed to all the parts of a deformable object through the influence field of a VOO (see figure 8.12) (see also colour plate 8.23 in section 8.6).

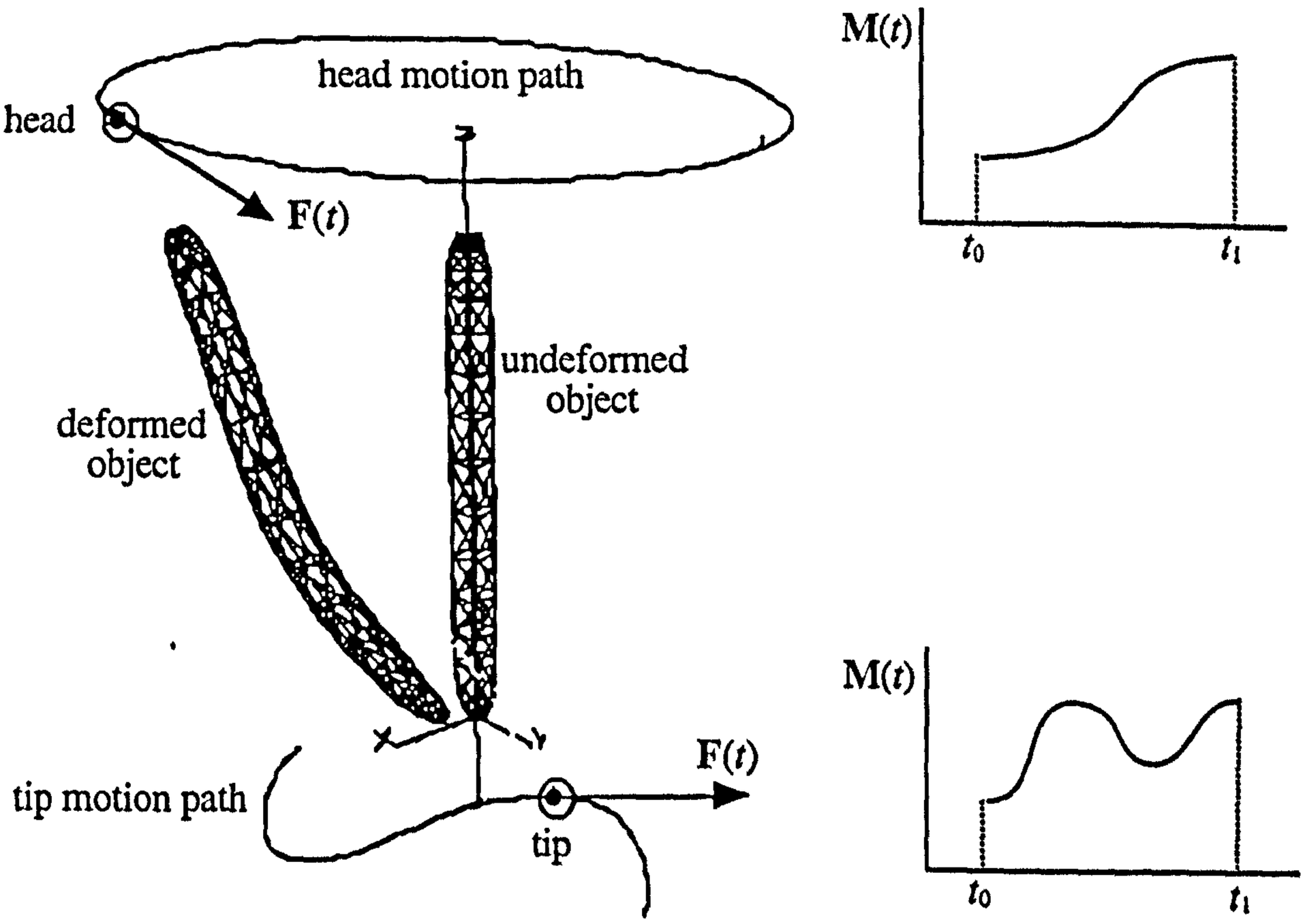


Figure 8.12 A VOO constrained on two motion paths.

Each motion path is equipped with a momentum graph, the start and end times and an initial velocity. The momentum graph yields the magnitude of the force applied on an end mass point. The direction of the force vector is given by the tangent on the path curve (see figure 8.12). Constraining VOOs on motion paths requires the solution of differential equations of motion (see section 8.2.4) only once per motion path per time step. This results to a very efficient

implementation of dynamic constraints. In conventional dynamic constraint methods for deformable objects it is usually required to solve large systems of simultaneous differential equations (one for every mass node / vertex) (Platt and Barr 1988), (Baraff and Witkin 1992), (Metaxas and Terzopoulos 1992).

8.3.3 Connecting Two or More VOOs

VOOs may be connected using free or fixed arthra (plural of arthron). Connected VOOs may be used for the creation of articulated structures and networks of VOOs. Such structures are capable of achieving a complex, yet controllable, deformation effect on 3D objects.

8.3.3.1 The Arthron

An arthron is created by constraining at least two VOO ends together. The tip of one VOO may be attached to the head of another VOO or a pair of VOO heads or tips may be attached together (see figure 8.13). Moving an arthron in space forces all attached VOO ends to move with it.

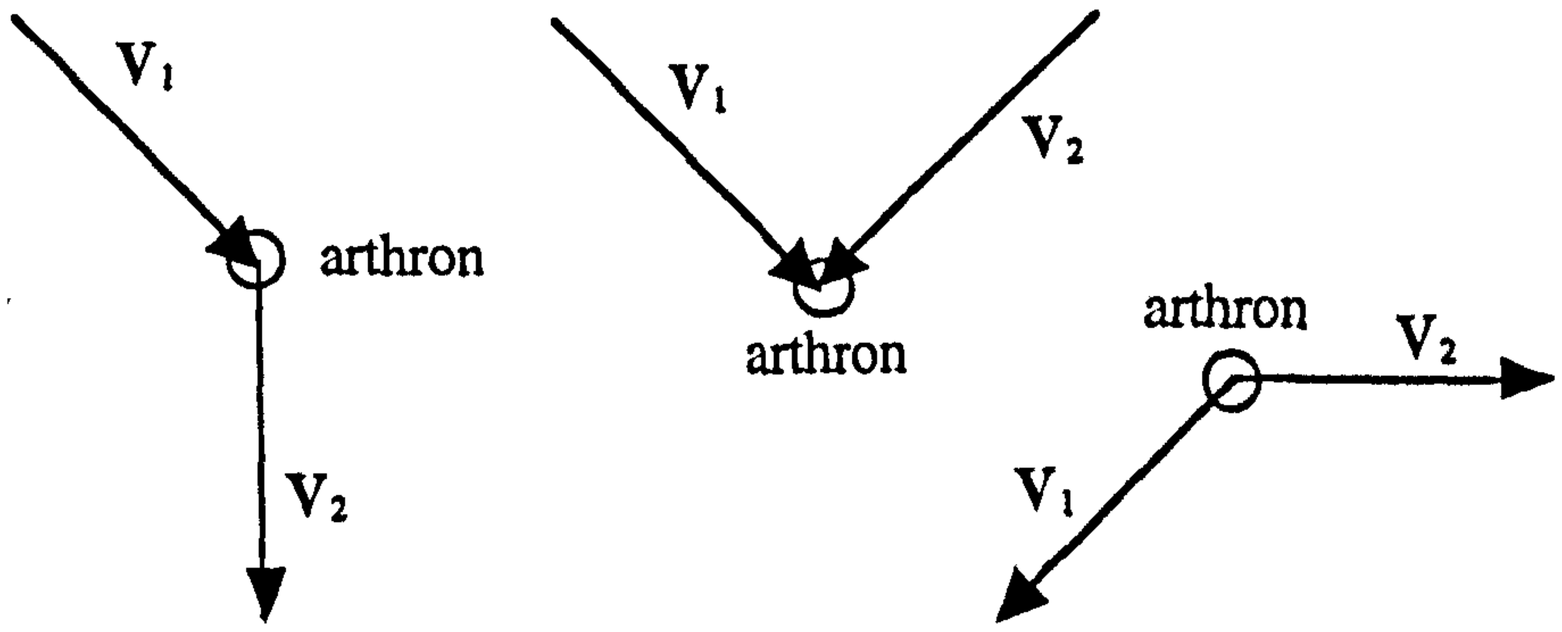


Figure 8.13 Two VOOs connected with an arthron.

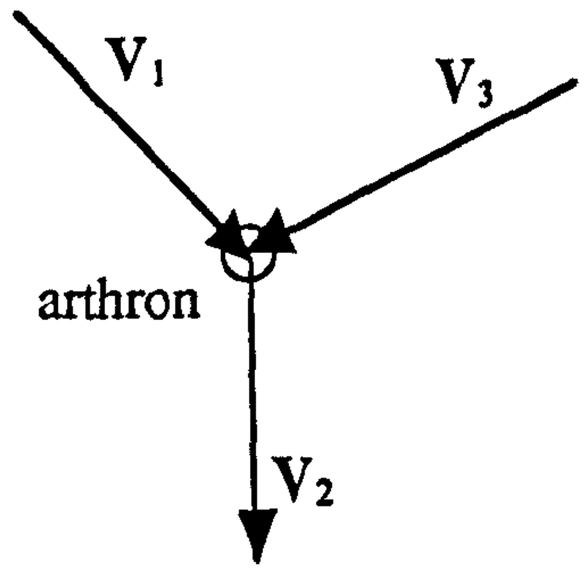


Figure 8.14 Three VOOs connected with an arthron.

An arthron may be created by constraining more than two VOO ends (see figure 8.14). Arthra may be used to connect VOOs and assist the creation of organised deformation structures (see colour plates 8.24 through to 8.29 in section 8.6).

8.3.3.2 Free and Fixed Arthra

A free arthron can be moved freely in 3D space. The new position of the arthron is inherited by all the attached VOO ends provided that there are no other spatial constraints applied to any of the participating VOOs (see figure 8.15). The position of a free arthron may be manipulated interactively or attached to a motion path.

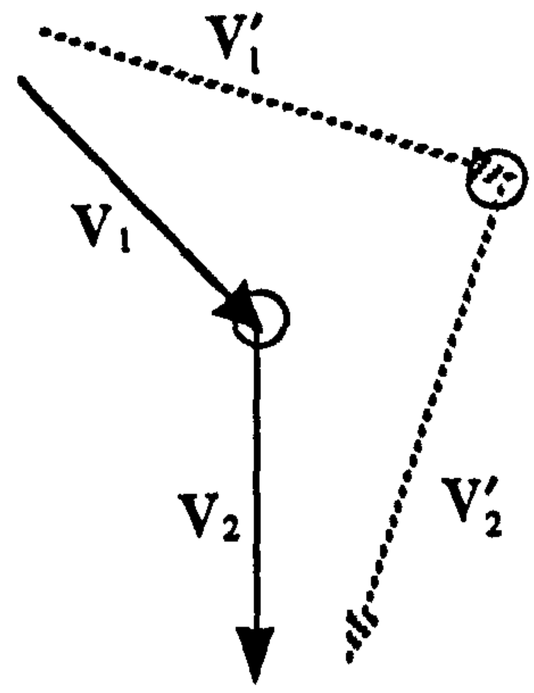


Figure 8.15 Two VOOs connected with a free arthron (marked with a circle).

A fixed arthron occupies a constant position in space and serves as a position constraint for all the attached VOO ends (see figure 8.16).

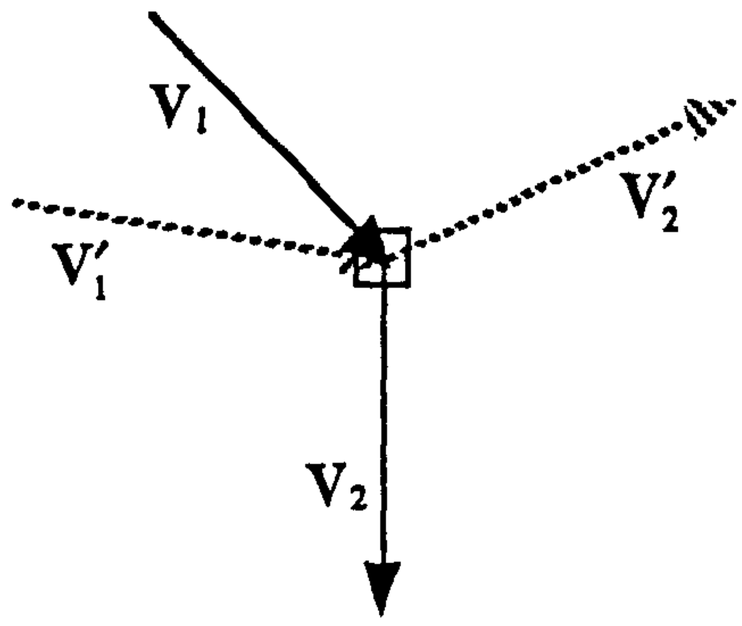


Figure 8.16 Two VOOs connected with a fixed arthron (marked with a square).

In the case where connected VOOs have also received other spatial constraints such as direction, aiming, position or length (see section 8.3.1) these constraints take precedence. In figure 8.17, for example, the movement of the free arthron between V_1 and V_2 is restricted by the fixed head and constant length constraints of V_1.

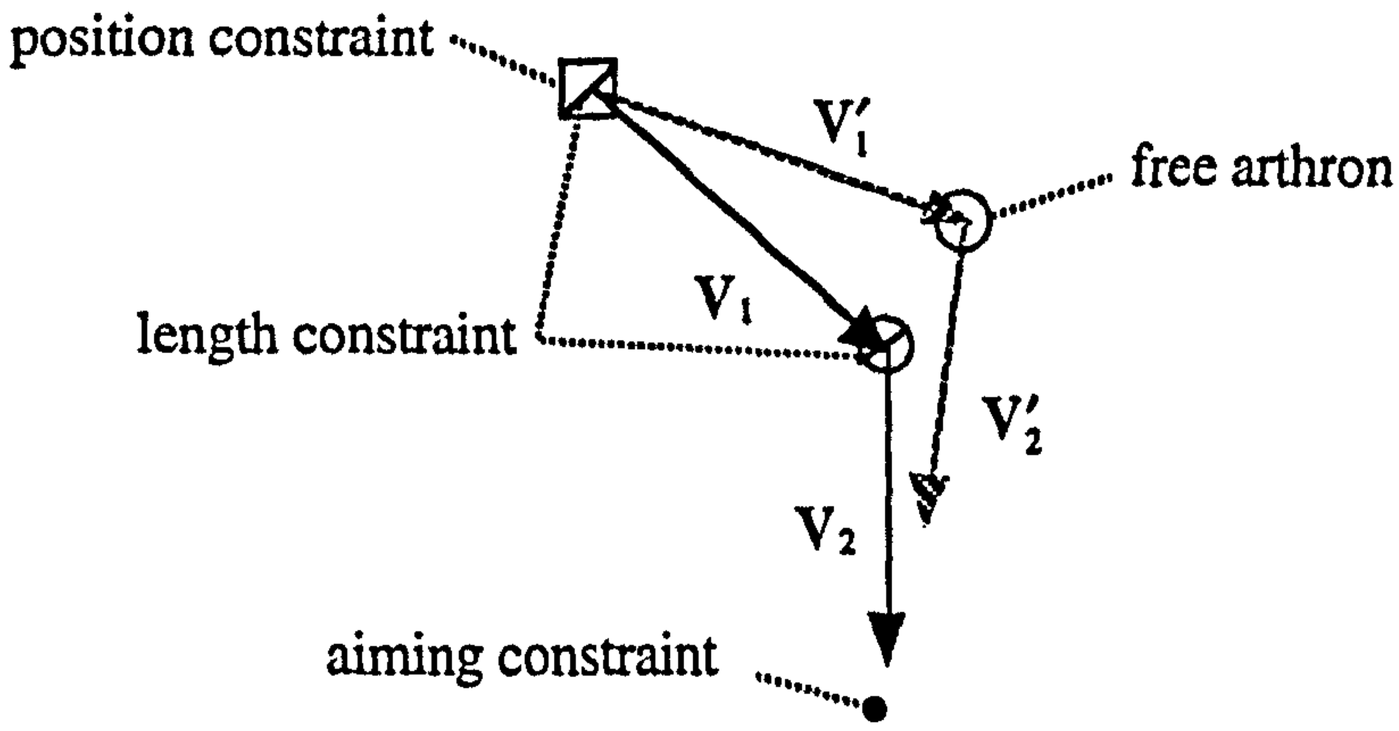


Figure 8.17 The head of V_1 has a fixed position, V_1 has a length constraint and the tip of V_2 has an aiming constraint. Tip of V_1 and head of V_2 are connected to a free arthron.

8.3.4 Influence Fields Revisited

The influence field of a VOO may be of type polar or length (see section 5.5). Connected VOOs may be assigned polar or length fields of influence. However, polar fields are more suitable for connected VOOs because they offer control over the cap coefficient κ (see section 5.5.2.3).

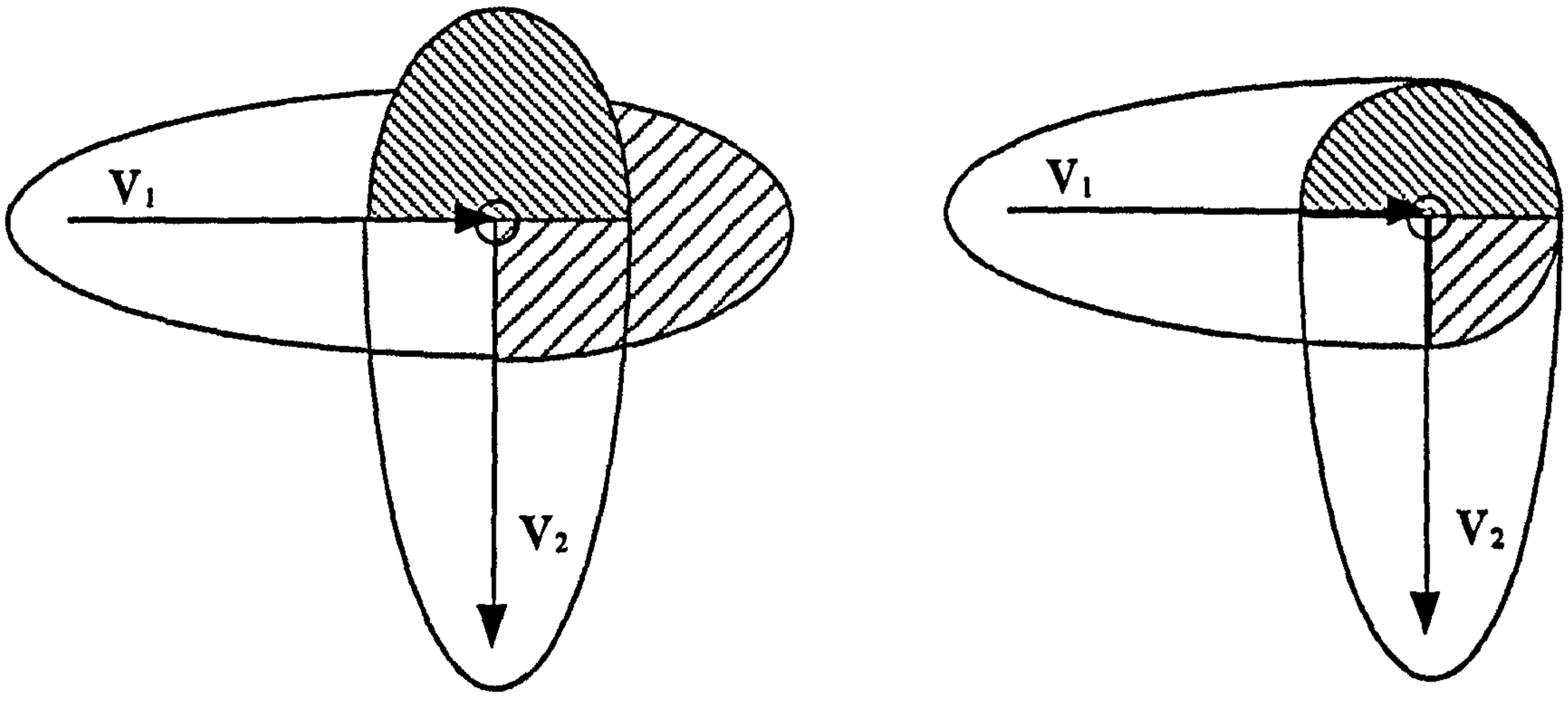


Figure 8.18 Adjusting the cap coefficient of the fields of influence around an arthron.

Manipulation of the cap coefficient has the effect of scaling the ellipsoid cap which allows the customisation of the shape of the fields of influence around an arthron (see figure 8.18). This enables control of the smoothness of deformation of the 3D objects that are influenced by the arthron. Object vertices that happen to lie inside both of the cap fields of the connected VOOs receive a weighed strain vector by pre-multiplying their respective damping elements.

8.3.5 Skeletons of VOOs

Sets of VOOs may be connected with arthra and organised into open chains and skeletons. Skeletons of VOOs may be used as underlying structures for soft bodied deformable characters (see figure 8.19).

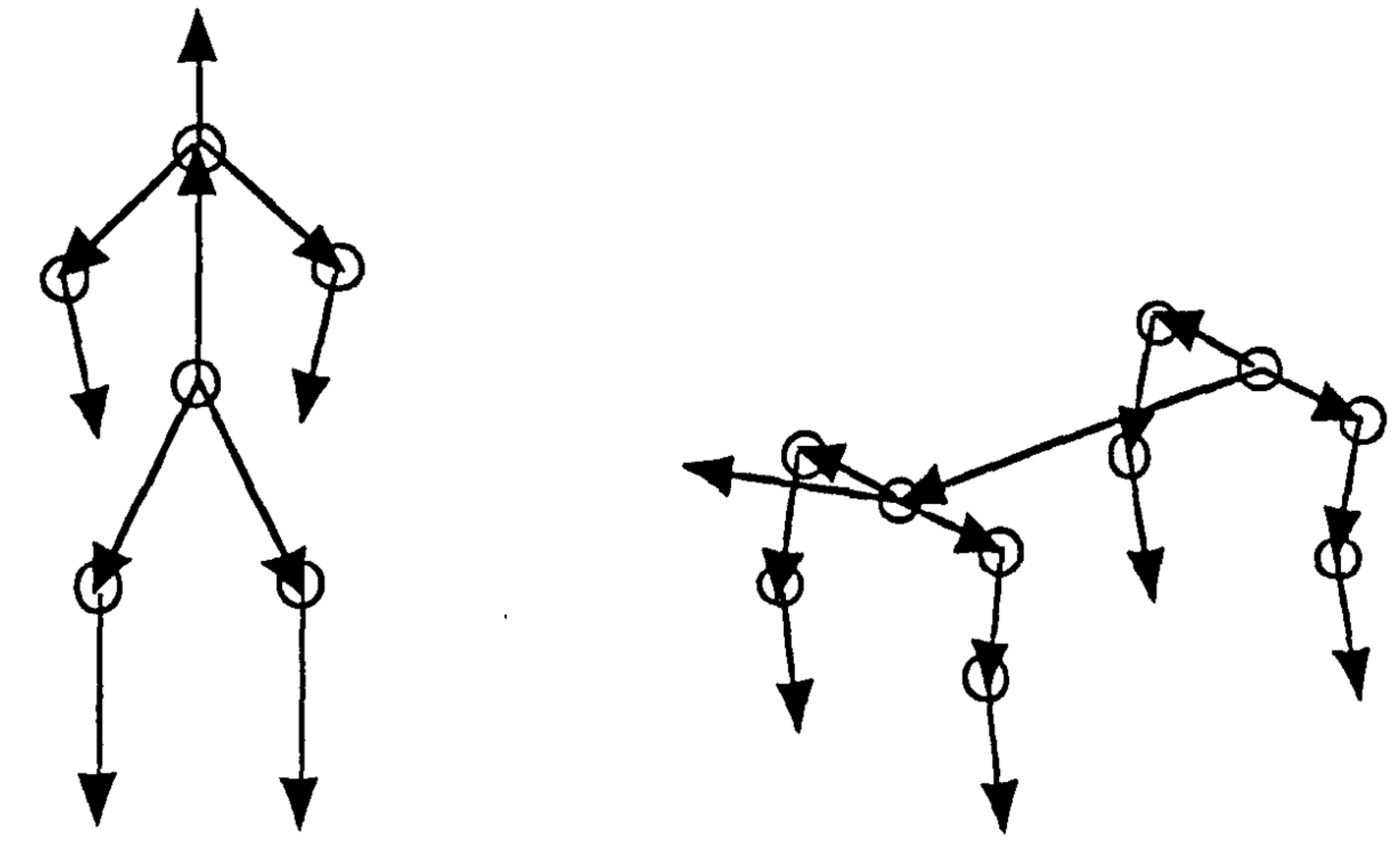


Figure 8.19 Simple bipedal and quadrupedal skeletons.

Arthra of a VOO skeleton may be attached to motion paths in order to create walk cycles, flight cycles, etc. The combination of VOO skeletons and motion paths offers a powerful and versatile mechanism for the animation of deformable organic characters.

8.3.6 A Plexus of VOOs

Three VOOs may be connected with three arthra as shown in figure 8.20. The three VOOs form a closed triangular cell.

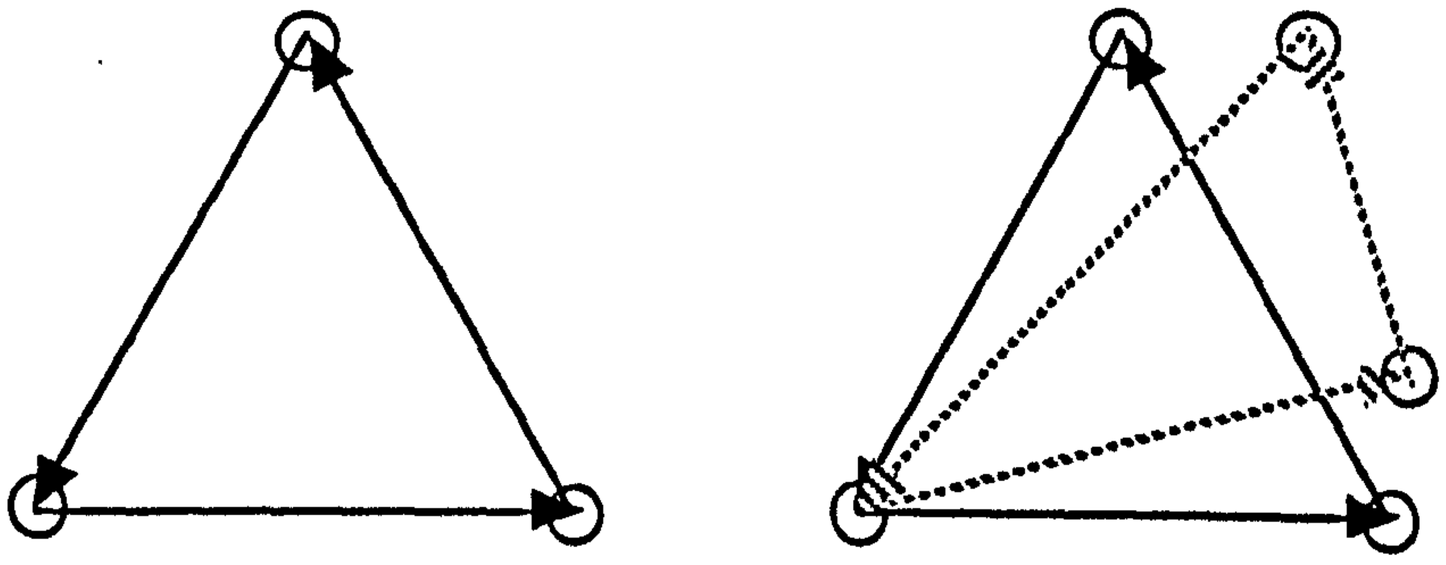


Figure 8.20 A triangular cell of VOOs.

Moving any of the three arthra distorts the deformation cell which in turn effects a deformation

upon the objects under its influence.

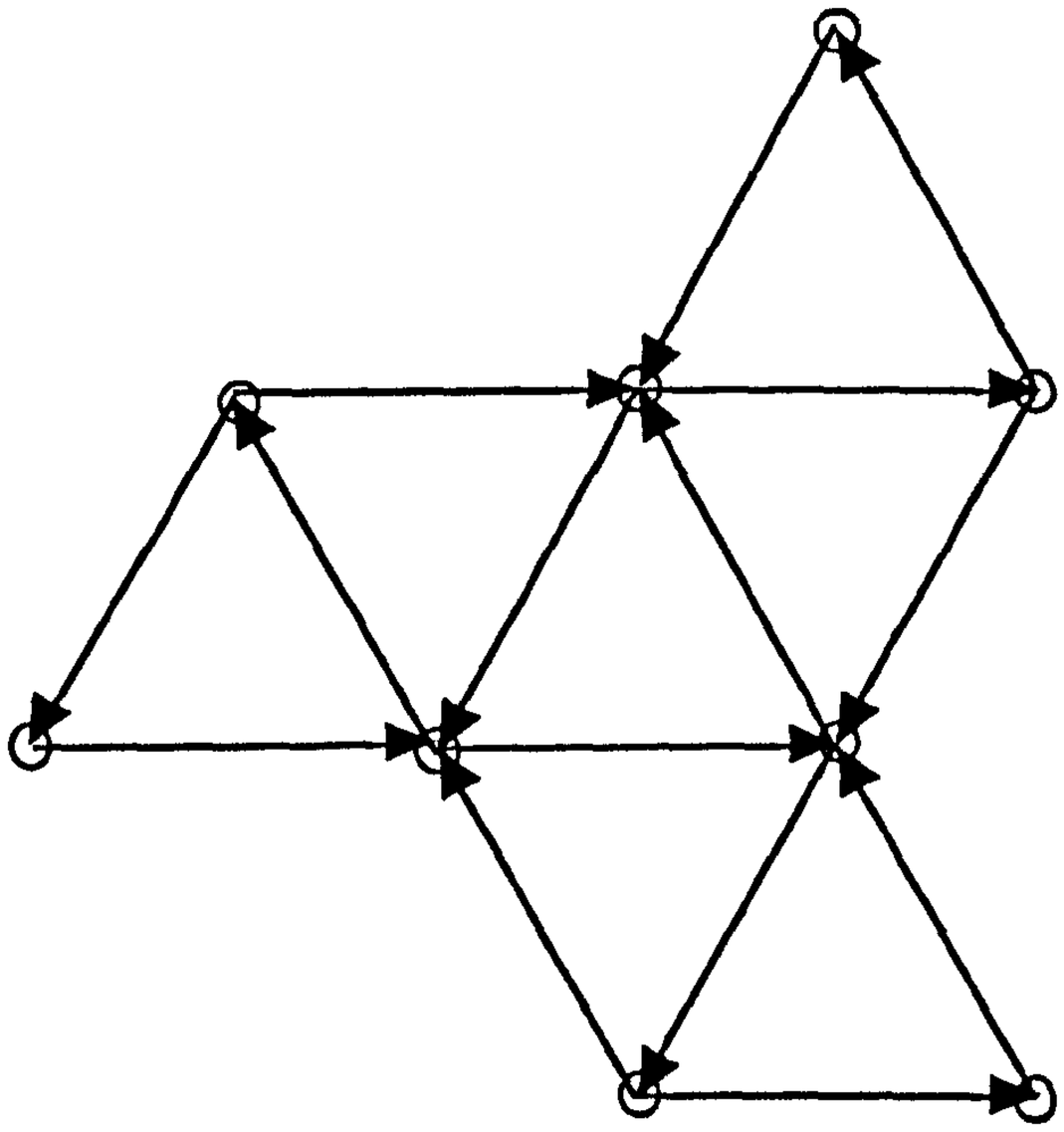
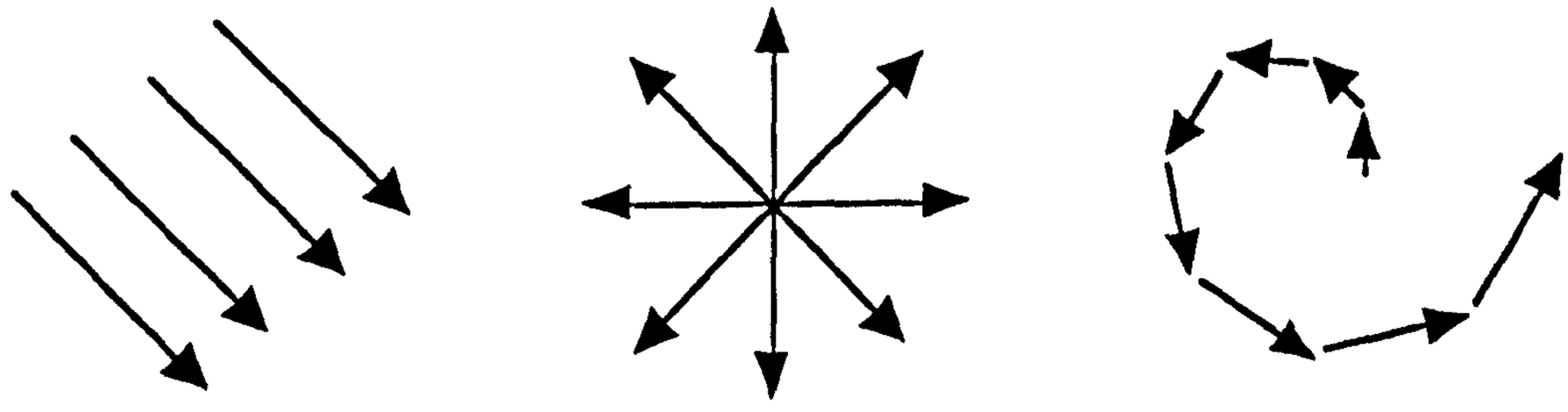


Figure 8.21 A plexus of VOOs.

Several triangulation cells may be connected to form a plexus of VOOs (see figure 8.21). A plexus of VOOs forms a complex deformation tool suitable for the animation of dynamic natural phenomena involving fluids, etc. (see colour plates 8.29 and 8.30 in section 8.6).

8.3.7 Vector Fields of VOOs

Organised sets of VOOs may be defined algorithmically using vector fields. For example, a directional vector field may consist of an array of VOOs all sharing the same direction (see figure 8.22a). A directional field of VOOs may be used to achieve the deformation effect of wind. A radial vector field consists of an array of VOOs all starting at a given point and having directions which are distributed radially (see figure 8.22b). A radial field of VOOs may be used to achieve the deformation effect of water springs. A vortex vector field consists of an array of VOOs which are arranged on a spiral (see figure 8.22c). A vortex field of VOOs may be used to achieve the deformation effect of tornadoes or other weather phenomena.



a. Directional field. b. Radial field. c. Vortex field.

Figure 8.22 Vector fields (Hilton and Egbert 1994).

Dynamic vector fields of VOOs may best be created and animated using an animation scripting language, such as CGAL (Comninos 1986) or MEL (Alias / Wavefront 1997). The state of each VOO may be defined using a time dependent vector field function.

8.4 Results

Several researchers have focused on the development of specialised constraint models that forcibly confine the outcome of physical behaviour. Such models deal with issues such as describing predefined trajectories, initial and final value problems, collision detection and avoidance, structural attachments, etc. (see Witkin and Kass 1988, Platt and Barr 1988, Cohen 1992, Baraff and Witkin 1992, Snibbe 1995).

In the suggested model, constraints are not applied directly on deformable objects. The functionality of the VOO deformation tool is being extended to become an interactive constraint application tool. So, a VOO serves as a carrier of spatiotemporal dynamic constraints as well as of deformation parameters. This two-level approach helps separate the deformation problem from the constraint application problem.

Spatial constraints, such as those of position, length, direction and aiming may be directly applied to single VOOs. Spatial behaviour of VOOs is restricted by these constraints whether the user attempts to modify the state of a VOO interactively or through an animation script.

Sets of VOOs may be connected with arthra to form open chains and skeletons. Skeletons of VOOs are suitable for constructing animated characters based on vertebrate living creatures (e.g. mammals). Sets of VOOs may also be connected with arthra to form triangular deformation cells. Many such cells may be combined to create plexi of VOOs. A plexus may be used as an underlying deformation mechanism for objects with fluid motion (e.g. water) or animated characters based on invertebrate living creatures (e.g. jelly fish).

Motion paths are spatial paths equipped with momentum / time graphs and initial and final time constraints. A motion path serves as a combined spatial and temporal constraint and it can be applied to the head, tip or aiming point of a VOO. A motion path may also be applied to an arthron of a skeleton or of a plexus of VOOs. Motion paths offer a versatile animation tool and may be used to solve a great variety of animation problems.

The combination of skeletal structures with motion paths and the elastoplastic and viscoelastic properties of VOOs enable the construction of animal locomotion mechanisms suitable for animation of organic characters. Motion paths, however, require numerical solution of ordinary differential equations which introduces a considerable computational overhead. Since mass is considered to be concentrated on the ends of VOOs these calculations will only be performed once per motion path per time step. This improves the performance of the dynamic constraints model.

Another way of constraining VOOs in space and time is through vector fields. The spatial and temporal behaviour of sets of VOOs may be defined using vector field functions. Vector fields of VOOs may be used to simulate the effect of wind, tornadoes, running water and other natural phenomena.

Colour plates relevant to this chapter can be found in section 8.6. Also, see animation tests 12 through to 18 in video tape, time code: from 10:06:45:00 to 10:08:38:00.

8.5 Summary and Conclusion

In this chapter, several ways of constraining the deforming effect of VOOs in space and time were introduced. The focus was concentrated on fast, effective and intuitive ways of dynamic constraint specification that are well suited to the animation practice. VOOs may receive constraints in space and in time either directly or indirectly. Directly specified constraints are spatial constraints which restrict the position, length, direction or the aiming point of a VOO. Indirectly specified constraints are arthra and motion paths. Arthra specify connection points between several VOOs. Motion paths help restrict the dynamic motion of a VOO in space and time. Skeletons and plexi of VOOs can be constructed by connecting several VOOs with arthra. Finally, vector fields may be used in order to define the state of VOOs algorithmically.

Applying constraints to the deformation tool rather than to the deformation itself creates an extra layer of control. In this way, the constraint model becomes simpler to implement because it is applied directly to a VOO and not to a deformable object. Improved efficiency is more obvious in the case of dynamic motion paths where numerical solution is required once per motion path and not per object vertex. The constraint model becomes more extendible and open to future extensions because it is independent from the geometric representation and the specific deformation algorithm.

The original concepts introduced in this chapter are:

- the combination of motion paths with momentum / time graphs to specify dynamic constraints,
- the application of spatial and temporal constraints directly onto VOOs.

The following chapter will overview applications and results of vector offset operators for deformable organic objects.

8.6 Colour Plates

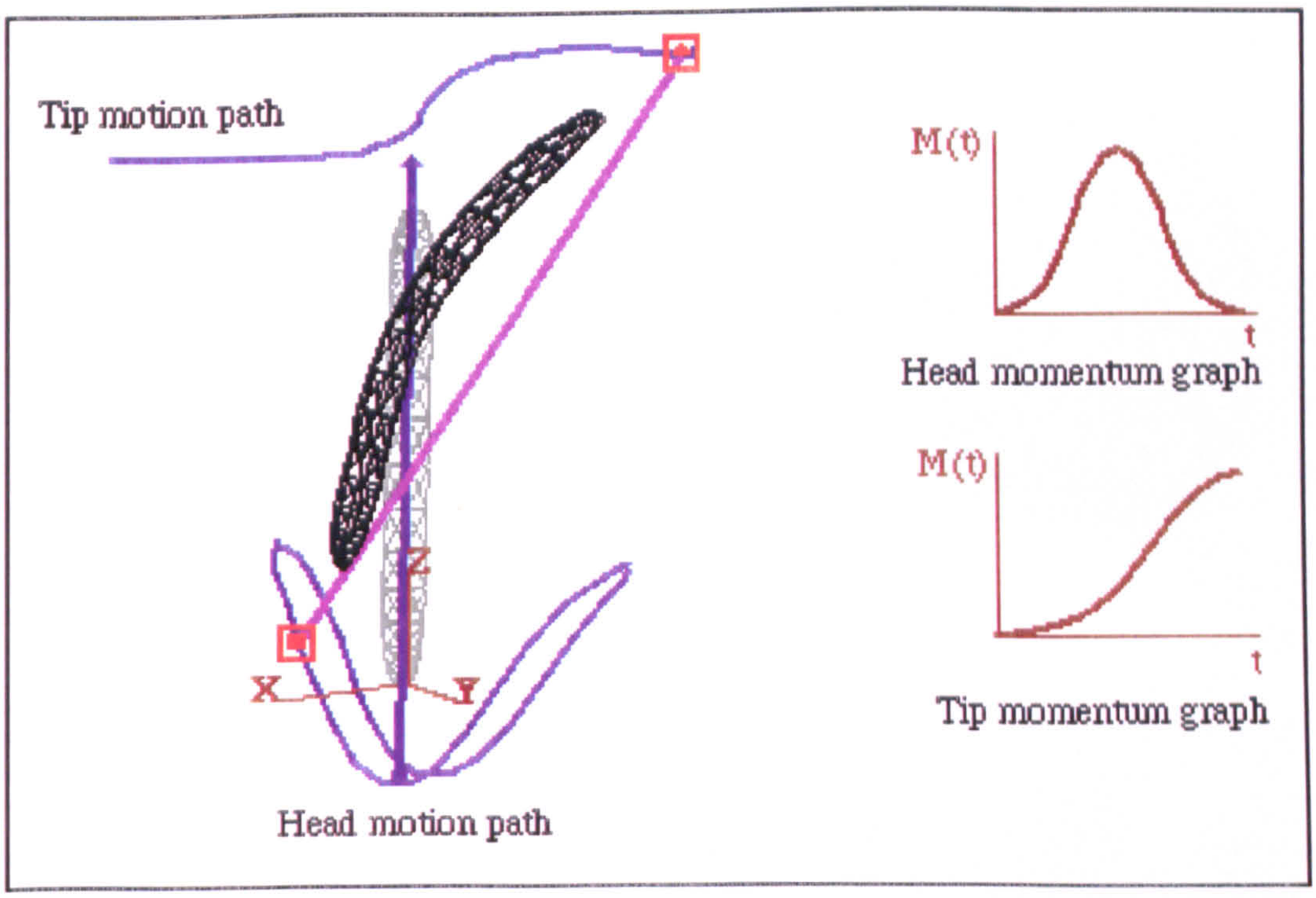


Figure 8.23 A VOO constrained with two motion paths.

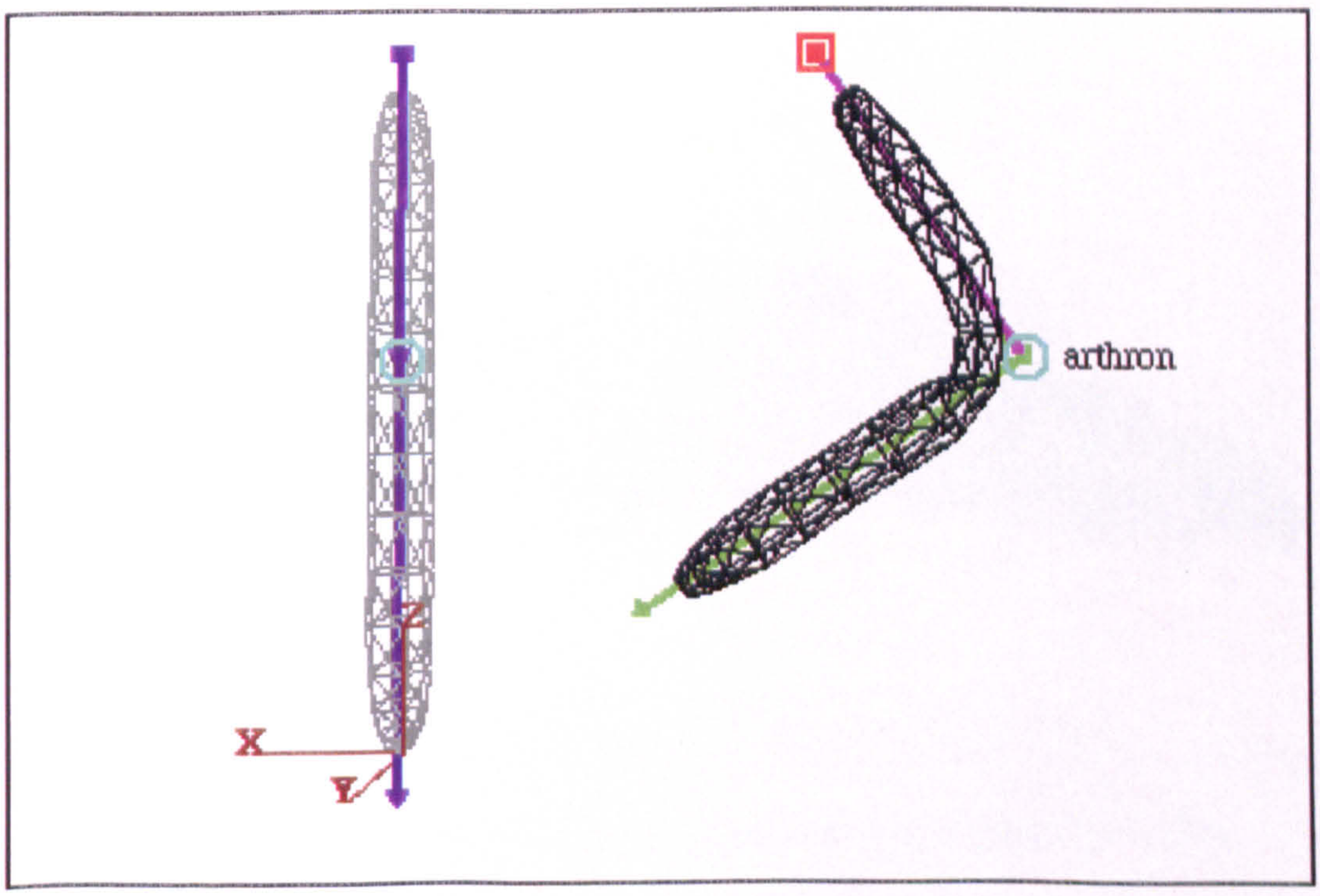


Figure 8.24 Two VOOs connected with one arthron.

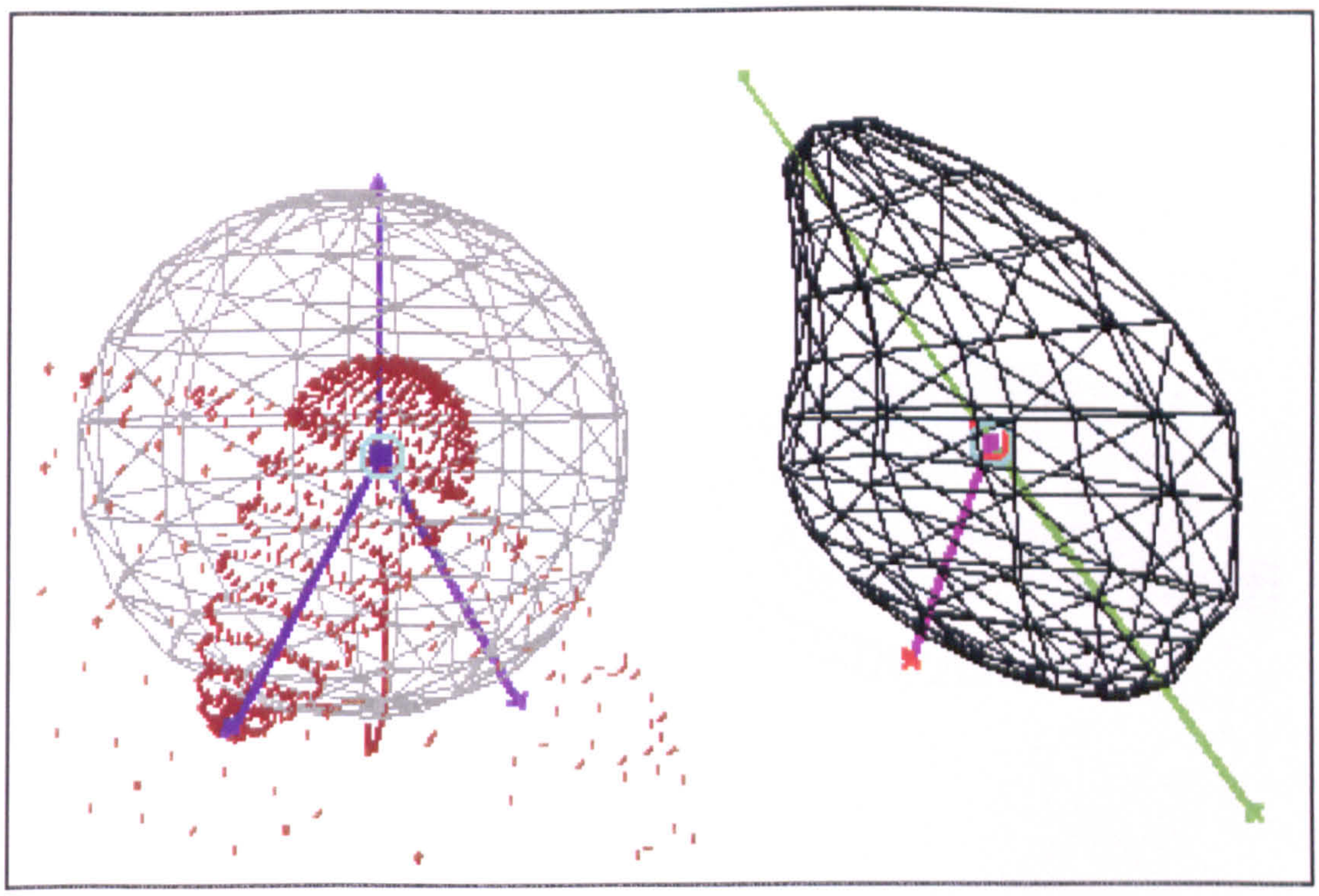


Figure 8.25 Three VOOs connected with one arthron.

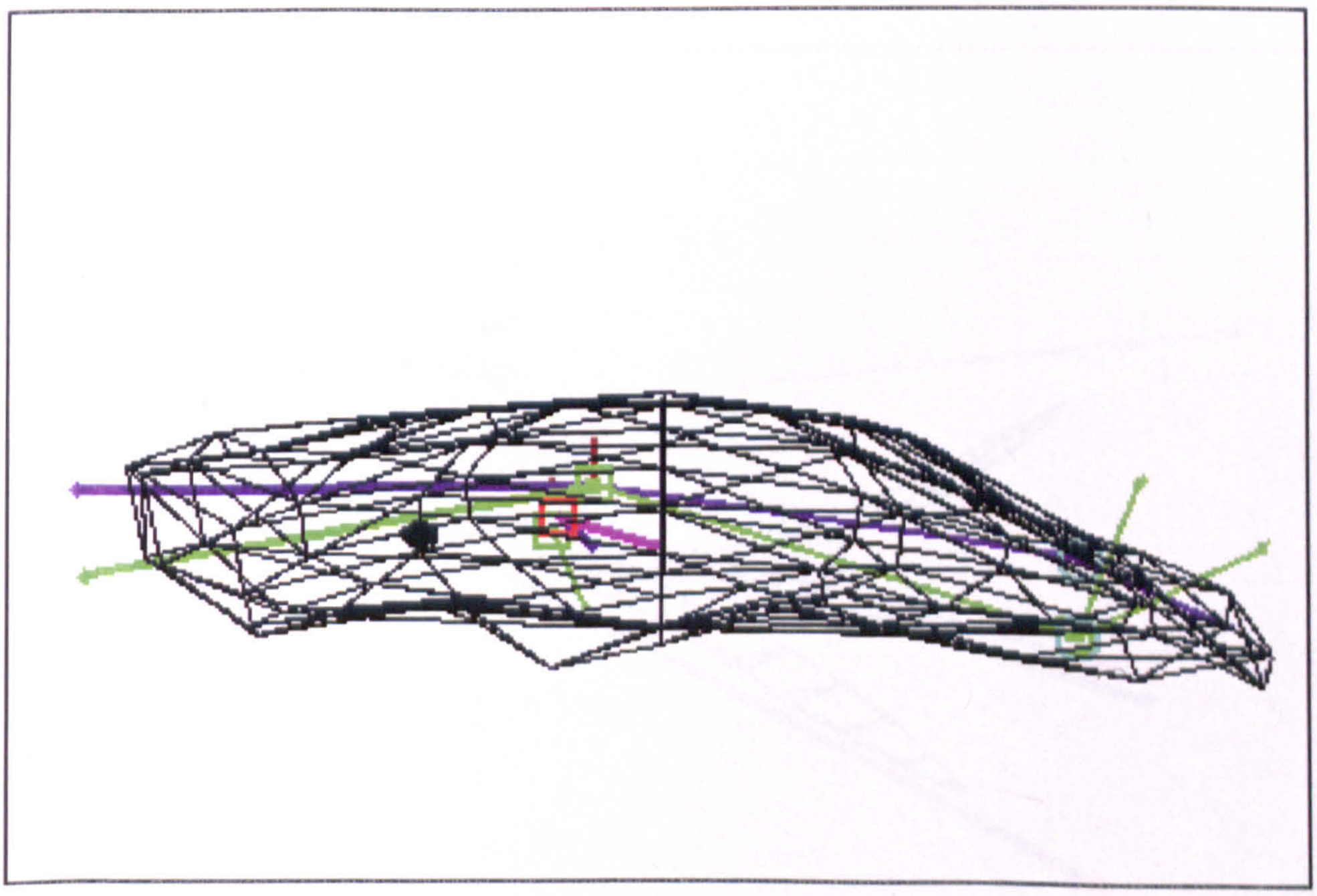


Figure 8.26 A reduced polygon whale model with a set of six VOOs.

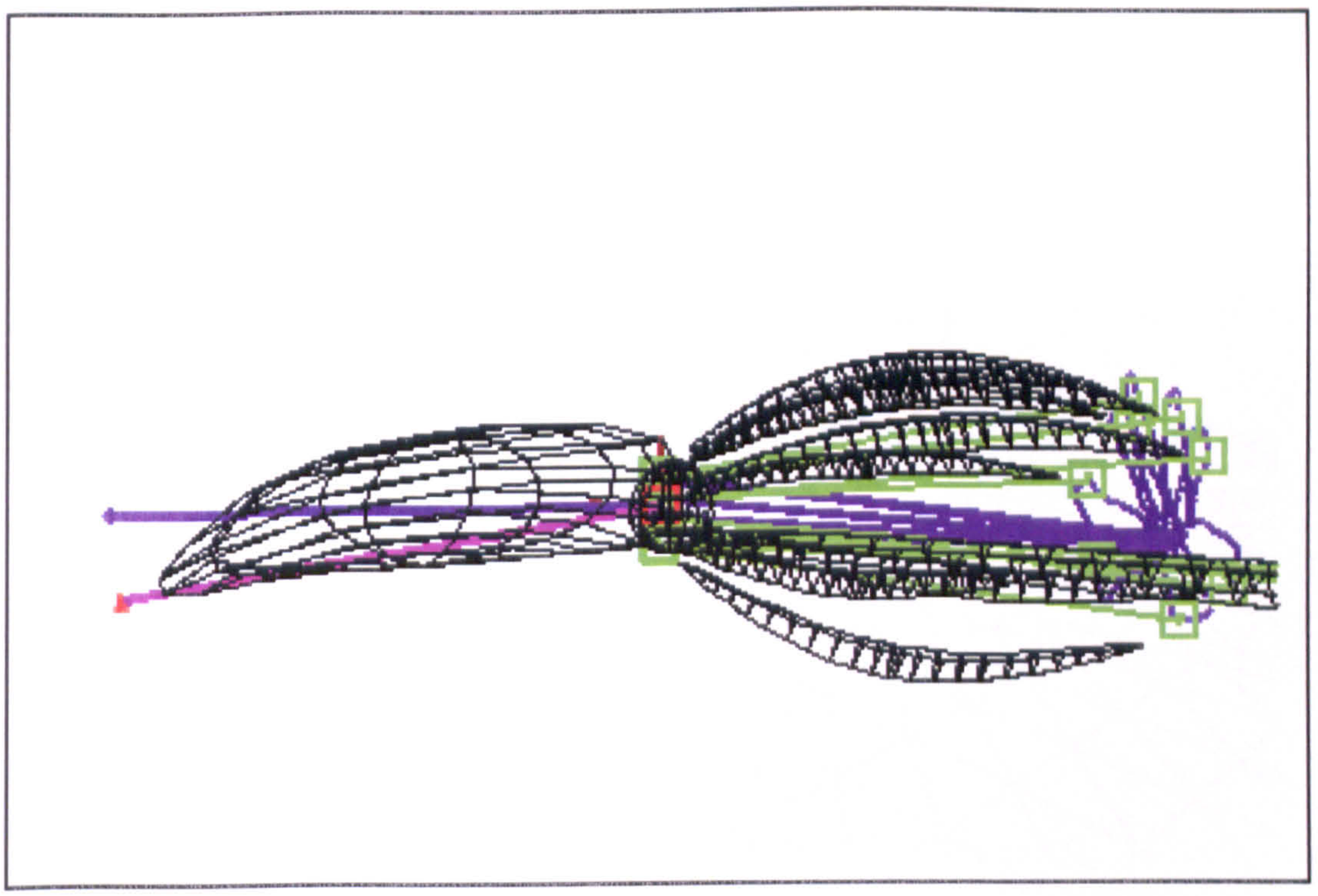


Figure 8.27 A reduced polygon squid model with a set of eleven VOOs. Eight of the VOOs have their tips constrained to motion paths.

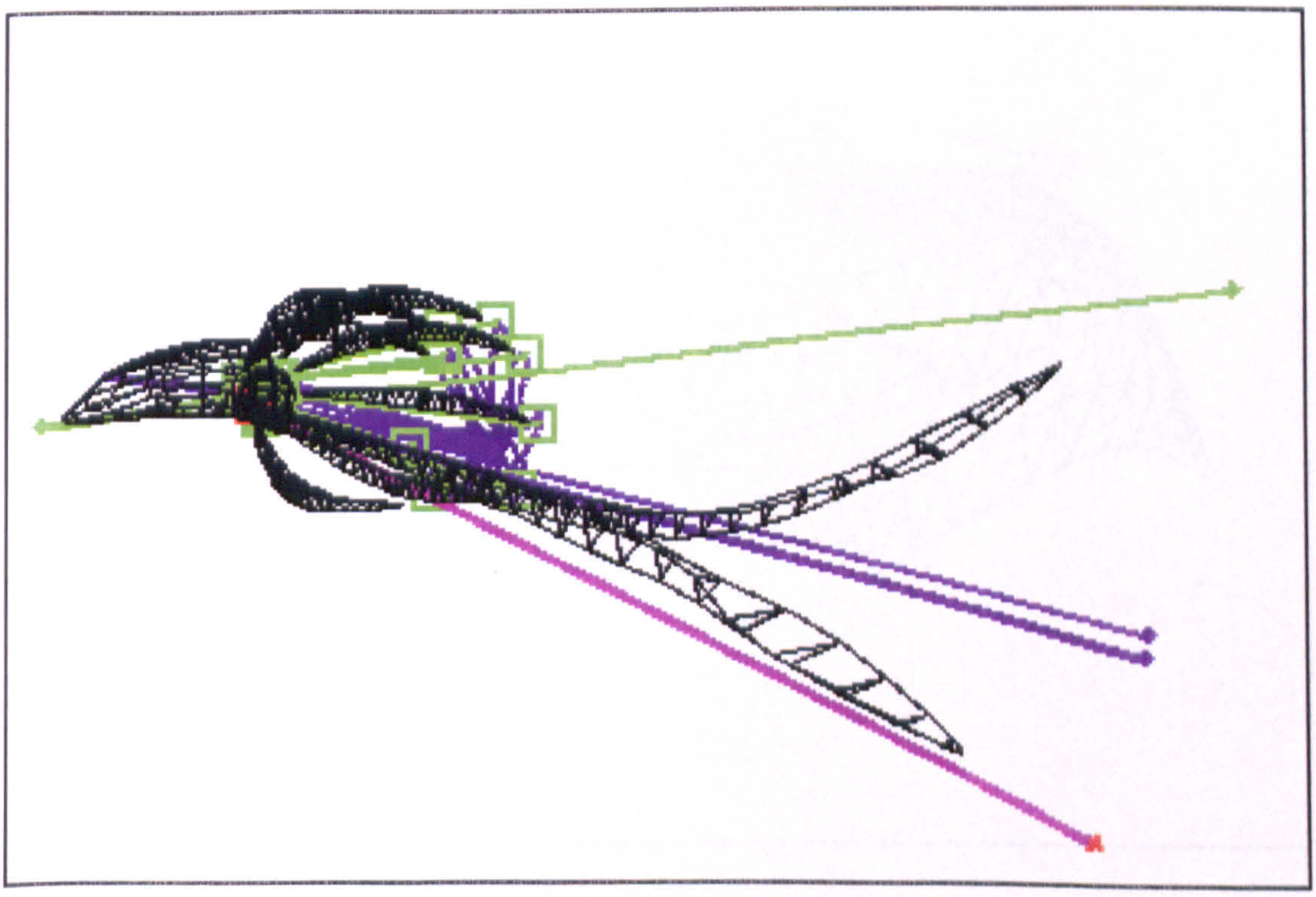


Figure 8.28 Same as in figure 8.27.

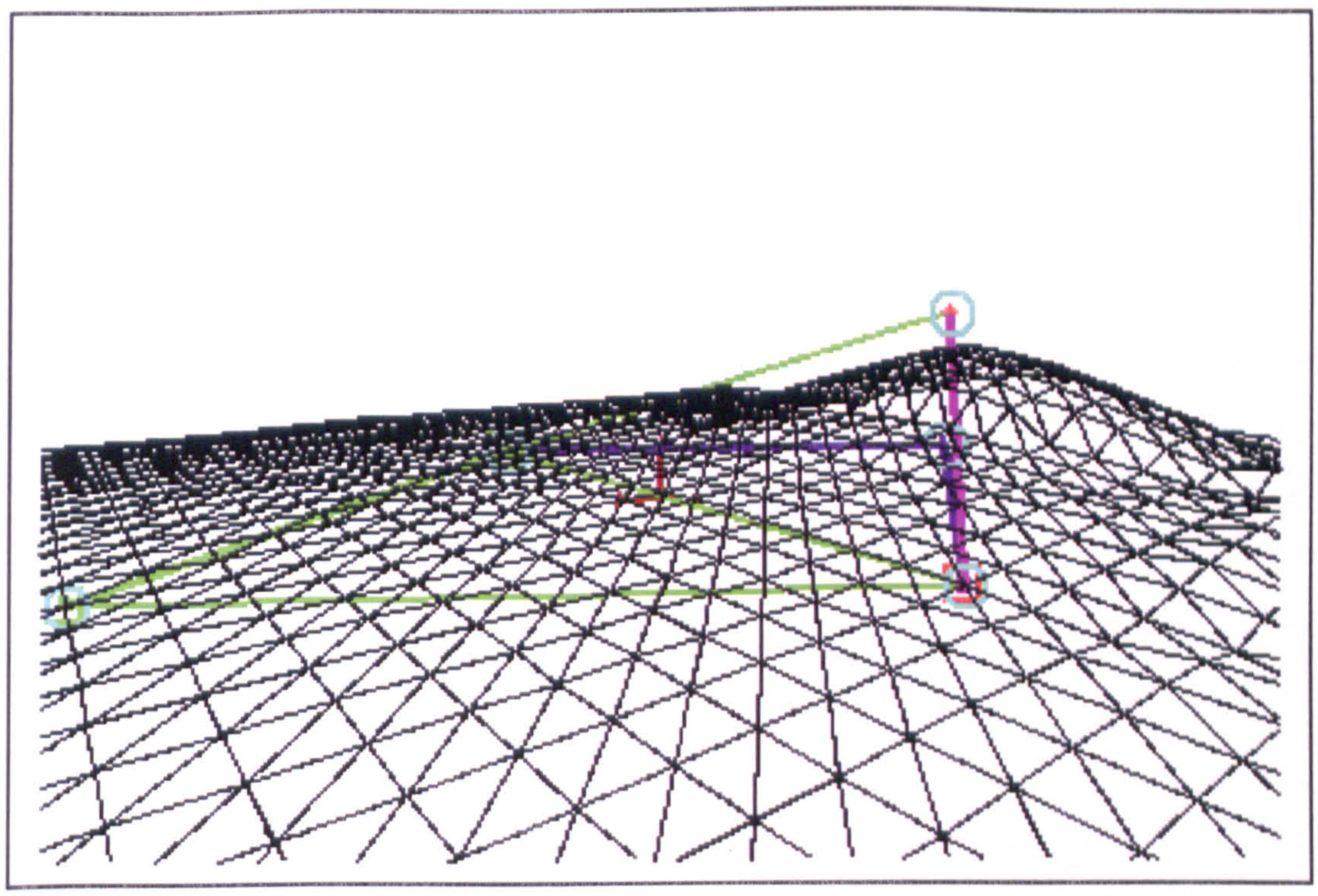


Figure 8.29 A terrain object influenced by a plexus of five VOOs.

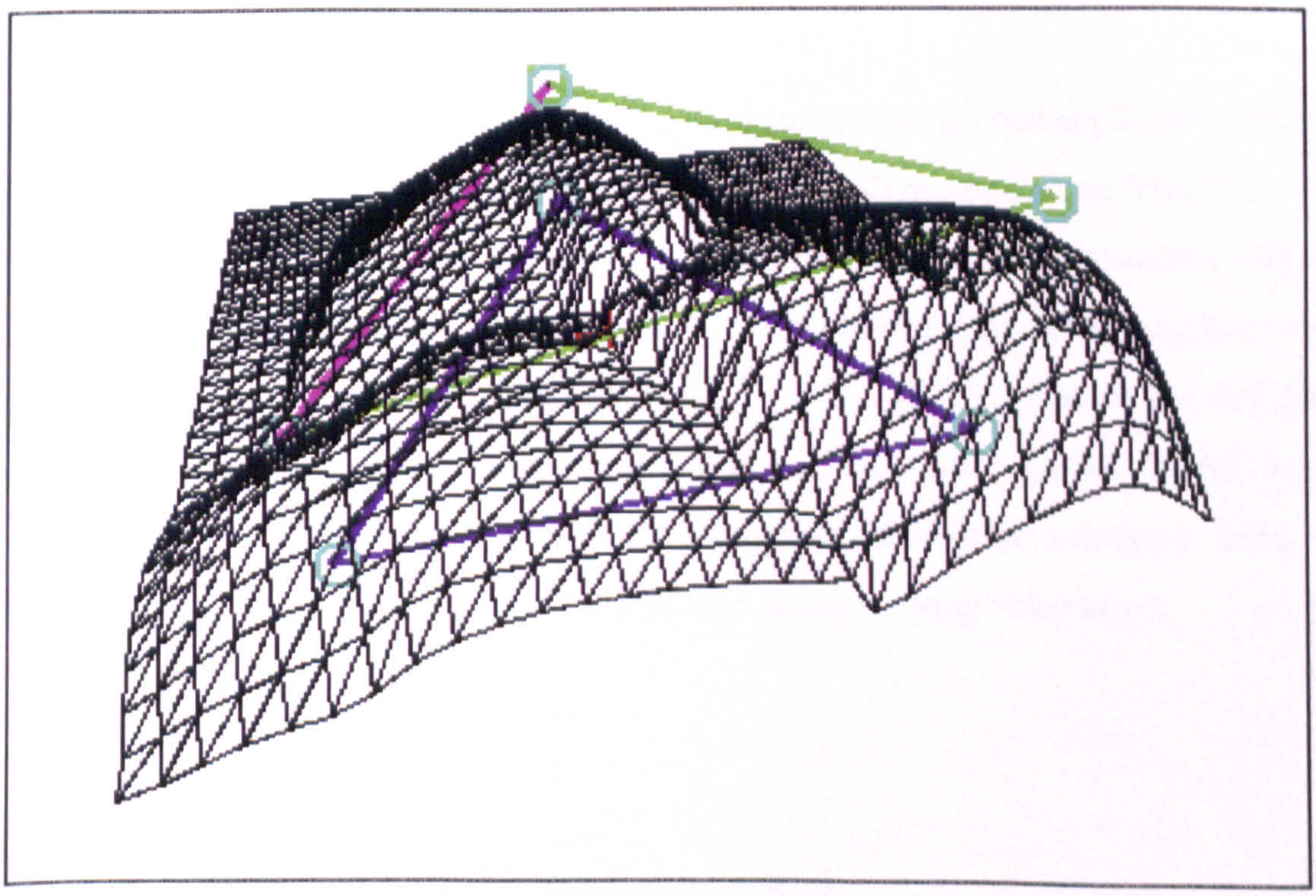


Figure 8.30 A terrain object influenced by a triangular cell of three VOOs.

9

Application and Results of Vector Offset Operators for Deformable Organic Objects

9.1 Introduction

The suggested deformation model has been implemented in the test bed application VOODOO (Vector Offset Operators for Deformable Organic Objects). The application VOODOO has an interactive interface which allows a user to interact with most of the deformation parameters described in the previous chapters. The application also has an animation interface which is linked to the animation system CGAL (Comninos 1980, 1986). Most of the deformation parameters may be controlled through animation scripts written in the CGAL scripting language. This application was used for the creation of several animated films which demonstrate the features of the suggested model (see accompanying video tape).

The objectives of this chapter are:

- to present the VOODOO interaction framework,
- to present the VOODOO animation framework
- to present the research results.

9.2 The Structure of VOODOO

The application VOODOO has been developed in a modular fashion (see figure 9.1). The core deformation model consists of four distinct models: the plastic model (see chapter 5), the elastoplastic model (see chapter 6), the viscoelastic model (see chapter 7) and the constraint model (see chapter 8). There are two further models which deal with the interactive and animation interfaces respectively. All four deformation models communicate with each other and are linked with the interactive and animation interface models via global data structures.

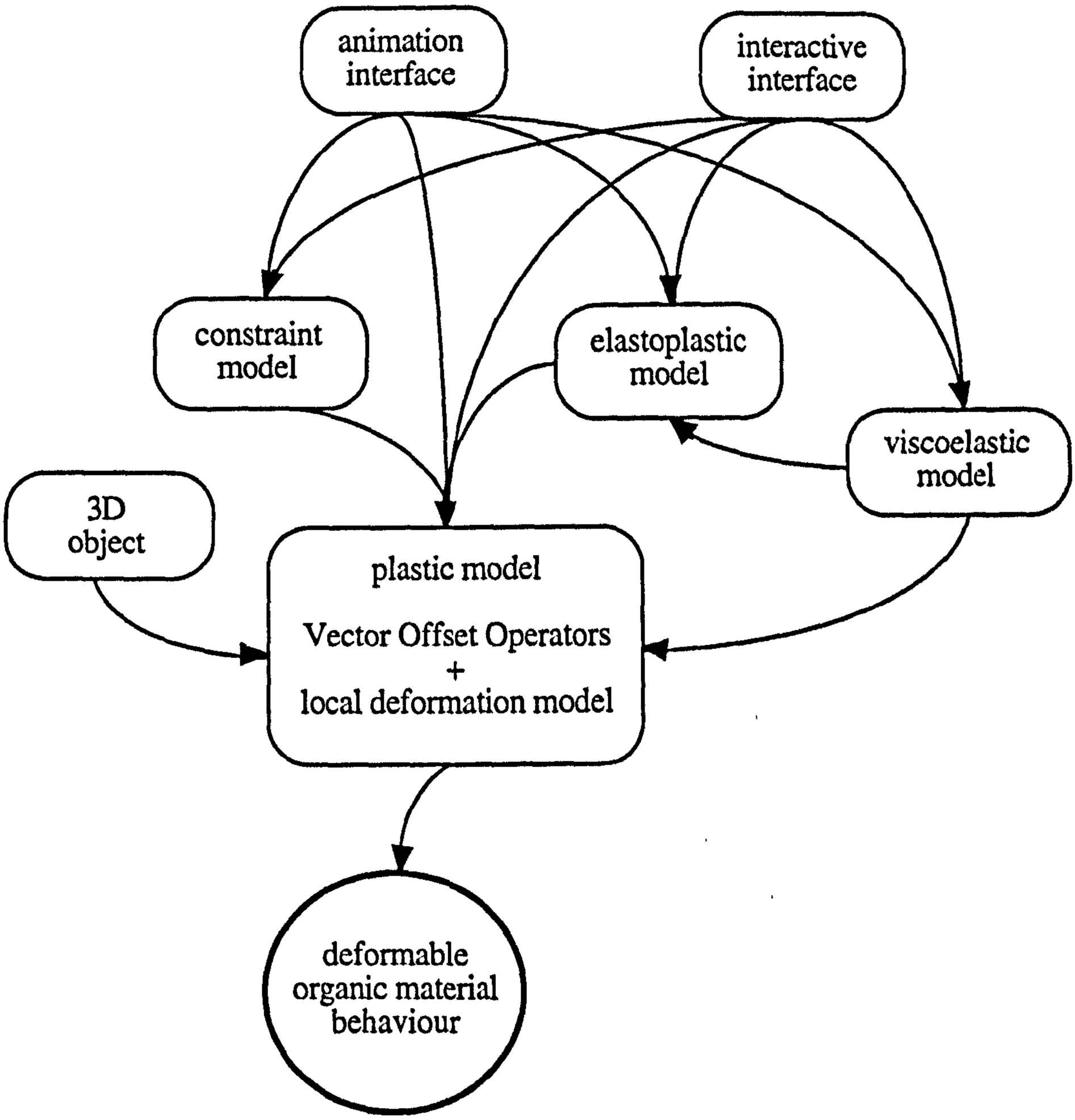


Figure 9.1 The structure of the application.

The design philosophy has been based on a combination of top down and bottom up techniques. The overall model is independent of geometric representation and it may be adapted to deform parametric surface control vertices, volumetric elements or polygon vertices alike.

9.3 The Interaction Framework

Object deformation in 3D space can be used to generate complicated shape and movement which are best tested interactively. Direct interactive manipulation of 3D deformable objects allows the freedom to test shape and movement visually and in real time (or in near real time) as opposed to animation time. Interaction can be achieved with a system which is capable of screen updates between 10 to 25 frames per second (Hz). Sometimes, interactive deformation may still be usable with slower update rates. Nevertheless, a deformation model must be capable of achieving effective interactive performance.

9.3.1 Using Interactive VOODOO

The suggested deformation model is capable of generating time based deformations which are self activated. Physically based models such as elastic restoration of VOOs (see chapter 6), viscoelastic strain creep (see chapter 7) and dynamic motion path constraints (see chapter 8) may be used to generate automated animation. Interactive manipulation of a set of parameters enables direct visual experimentation and fine tuning of self activated deformation behaviour.

Interaction may also be used to manipulate a whole range of spatial deformation parameters, such as VOO manipulation, influence field, strain control and strain constraint parameters (see chapter 5). This enables direct evaluation of the visual effect created by each spatial deformation parameter.

An interactive interface may also serve as a test bed for testing the performance of various features of the deformation model. Modular development of the suggested deformation model

allows the user to switch features on and off and thus isolate certain features for more accurate performance testing (see sections 5.9.2, 6.4.2 and 7.5.2).

Another use of the interactive framework is as a 3D modelling tool. The shape of an object may be deformed using spatial deformation parameters. The resulting object may then be used as a static model or it may participate as a key-shape in interpolated animation using a key-framing technique.

9.3.2 Interactive VOODOO Parameters

A brief description of the set of user defined parameters is presented in the following sections.

9.3.2.1 The Spatial Deformation Parameters

A set of spatial deformation parameters can be used to apply local strain to objects, to control the shape of deformation and to constrain the spatial behaviour of deformation inside the influence field of a VOO.

A VOO may be interactively manipulated by offsetting its head and tip along each of the world axes in 3D space (see section 5.3.2). This may result in global translation and local bending and elongation or contraction of the influenced object. A VOO may also be rotated about its local main axis thus causing torsion strain on the object. The local strain vector is applied to every vertex inside the field of influence of a VOO using the strain damping function (see section 5.4.2). The user may define a strain damping function by drawing a 2D graph in the CGAL polygon editor. The graph may then be imported to VOODOO and be assigned to an influence field of a VOO. The size and shape of the influence field ellipsoid may be interactively controlled by scaling the ellipsoid radii and the ellipsoid cap coefficient (see section 5.5). The user also has interactive control over the three strain control coefficients of flexibility, locality and weight and the three strain constraint coefficients of elongation, volume preservation and

cross-section (see sections 5.4.3, 5.4.5 and 5.4.6).

The colour plates 5.45 through to 5.56 in section 5.11 illustrate the effect of most of the spatial deformation parameters.

9.3.2.2 The Organic Material Parameters

A set of organic material parameters can be used to assign elastoplastic and viscoelastic material properties and to control the dynamic behaviour of deformation inside the influence field of a VOO.

The elastoplastic model is based on an elastic energy damping function and a stress / strain function which convey the spatial and temporal behaviour of an elastic VOO (see section 6.3). The user may define these two functions by drawing 2D graphs in the CGAL polygon editor. The graphs may then be imported to VOODOO and be assigned to a VOO. The user also has interactive control over the parameter of mass, the elastic strain limit, the fracture strain limit, the start time and the duration of the elastic restoration.

The viscoelastic model is based on a spatial gradient of viscous rate function, a strain creep function, a stress relaxation function and a stress / strain hysteresis function which convey the spatial and temporal behaviour of a viscoelastic VOO (see sections 7.2, 7.3 and 7.4). The user may define these four functions by drawing 2D graphs in the CGAL polygon editor. The graphs may then be imported to VOODOO and be assigned to a VOO. The user also has interactive control over the coefficients of viscosity, strain creep, the duration of strain creep process and the duration of stress relaxation process.

The colour plates in section 6.6 and in section 7.7 illustrate the effect of some of the organic material properties.

9.3.2.3 The VOO Constraint Parameters

A set of spatial and temporal constraint parameters can be used to control the dynamic or kinematic behaviour of a VOO.

Motion paths are based on pairs of spatial paths and momentum functions (see section 8.2). The user can define a spatial path by drawing a polygon using the CGAL polygon editor. Momentum functions may also be based on polygonal graphs. Other motion path related parameters are the start and end times, the initial velocity and the mass. The user can also choose to "hook" or "unhook" a VOO from a motion path. Finally, the user may interact with the position of an arthron connection and thus control skeletons and plexi of VOOs (see section 8.3).

The colour plates in section 8.6 illustrate the effect of some of the VOO constraint parameters in interactive VOODOO.

9.4 The Animation Framework

In the previous sections it was shown that interactive VOODOO may be used for visual experimentation and evaluation of deformation parameters, for performance testing and for 3D modelling.

Animated deformation may be achieved simply by using interactive VOODOO as a modelling tool and by combining it with a key-framing technique. However, a much more powerful and flexible approach to animation may be based on a combination of VOODOO and animation scripting. VOODOO deformation parameters may be controlled through animation scripts written in the CGAL scripting language. This creates an animation framework suitable for testing time related parameters as well as for animation production.

9.4.1 Linking VOODOO to CGAL

CGAL is a 3D computer animation system which supports an interactive interface and a scripting language (Comninos 1980, 1986). The CGAL scripting language is based on Pascal-like code. Through CGAL scripting an animator may access an extensive list of sophisticated modelling, animation and rendering routines. CGAL scripting supports sparse arrays which can be used to simulate data structures and linked lists. VOODOO has been interfaced to CGAL through such sparse arrays. All VOO related parameters may be assigned to fields of a CGAL sparse array. All the related deformed objects may be created using the CGAL modelling facilities or be imported from other sources. These objects may also be assigned to the elements of a related sparse array. This enables access via scripting to any of the VOO parameter or influenced object through specific array elements.

In the following CGAL script example, one "character" object and one VOO are defined by a set of parameters which are assigned to the elements of a sparse array "ina". The VOO then is animated for 100 frames by translating its head and tip using CGAL time variant functions.

```
var ina, outa, noc, nov, noa, id, v, i;

procedure set_VOO_parameters;
begin
(* VOODOO input array *)                    ina = array(3);
(*.....*)
(* VOODOO Characters *)
(* Number of VOODOO characters *)  noc = 1;        ina[1] = array(noc);
(* List of VOODOO characters *)
(**)
(* 1st character id *)                      id = 1;
(* Character name *)                        ina[1, id] = [object_name, 0, 0, 0];
(*.....*)
(* VOOs *)
(* Number of VOOs *)                        nov = 1;
                                            ina[2] = array(nov);
(* List of VOOs *)
(* 1st VOO id *)                            v = 1;
                                            ina[2, v] = array(27);
(* VOO Id *)                                ina[2, v, 1] = v;
(* VOO type *)                              ina[2, v, 2] = 1;
(* Id list of influenced character ids *)   ina[2, v, 3] = [1];
(* Initial VOO [[head],[tip]] *)            ina[2, v, 4] = [[0, 0, -0.5], [0, 0, 4.5]];
(* VOO Head position *)                     ina[2, v, 5] = [0, 0, -0.5];
(* Head parameters *)                       ina[2, v, 6] = [0, 0, 25, 0, 1, 1];
```

```
(* VOO Tip position *)                    ina[2, v, 7] = [motion_path, 0];
(* Tip parameters *)                      ina[2, v, 8] = [momentum_graph, 0, 50, 0, 1, 1];
(* Influence field scaling factor *)      ina[2, v, 9] = 1;
(* Locality coefficient *)                ina[2, v, 10] = 0;
(* Weight coefficient *)                  ina[2, v, 11] = 1;
(* Torsion strain *)                      ina[2, v, 12] = 0;
(* Torsion coefficient *)                 ina[2, v, 13] = 1;
(* Elongation coefficient *)              ina[2, v, 14] = 1;
(* Volume preservation coefficient *)     ina[2, v, 15] = 1;
(* Flexibility coefficient *)             ina[2, v, 16] = 1;
(* Spatial damping of strain graph *)     ina[2, v, 17] = strain_damping_graph;
(* Mass *)                                ina[2, v, 18] = 50;
(* Elasticity coefficient *)              ina[2, v, 19] = 50;
(* Elastic limit *)                       ina[2, v, 20] = 1;
(* Fracture limit *)                      ina[2, v, 21] = 10;
(* Elastic restoration max time *)        ina[2, v, 22] = 25;
(* Stress / strain graph *)               ina[2, v, 23] = stress_strain_graph;
(* Viscosity coefficient *)               ina[2, v, 24] = 1;
(* Viscoelastic creep max time *)         ina[2, v, 25] = 25;
(* Viscoelastic creep graph *)            ina[2, v, 26] = creep_time_graph;
(* Integration step *)                    ina[2, v, 27] = 1;
(*.....*)
(* VOODOO arthra *)
(* Number of arthra *)                    noa = 0; ina[3] = array(noa);
(*.....*)
(* Prepare character copies *)            for i = 1 to noc do ina[1,i,2] = copy(ina[1,i,1]);
(* Initialise VOODOO *)                   outa = eahf_1( [0, ina] );
(*.....*)
end;

(* main script *)
begin
 cvoff;
 dso;
 set_VOO_parameters;
 stfon;
 (* animation script *)
 script 1 to 100 do
  begin
   write frame;
   (* animate VOO Head *)
   (* position X *)                        ina[2, 1, 5, 1] = ctvf(0, 6.5, 1, 100);
   (* position Y *)                        ina[2, 1, 5, 2] = ctvf(0, 5.5, 1, 100);
   (* position Z *)                        ina[2, 1, 5, 3] = ctvf(-0.5, 1.5, 1, 100);
   (* VOO Tip is animated via motion path *)
   (* Call VOODOO *)                       outa = eahf_1( [22, ina, frame] );
   (* Select deformed characters *)        dso; for i = 1 to noc do so ina[1, i, 2];
  end;
end.
```

The above CGAL script is a simple example of a single VOO operating on one object. Longer scripts and CGAL customised macros may be written in order to define arrays of objects operated upon by arrays of VOOs and VOOs may be connected with arthra to form skeletons or plexi. Any of the VOO parameters may be animated by assigning it to a frame related expression. VOOs may also be animated by constraining their head and or tip to dynamic motion paths. Elastic restoration of VOOs, viscoelastic strain creep and stress relaxation processes may be triggered at specific frames during an animation and their duration may be constrained between start and end frames. CGAL macro scripts may be used to invoke an interactive VOODOO window through which the user has direct control over most of the parameters (see section 9.3).

9.4.2 Animations with VOODOO

During the course of this research project the candidate has been involved in the production of several animated films using the interactive and animation VOODOO frameworks. Some of these animations are short clips and were produced purely for demonstration purposes. Others, came as the result of a combined scientific, creative and commercial effort.

9.4.2.1 Test Animations

Several short animation tests demonstrate the visual effect of the deformation features of VOODOO. Here follows a list of descriptions of these tests (see colour plates 9.11 through to 9.20 in section 9.7).

Animation Test 1 Plastic VOODOO

(see video tape, time code: 10:04:05:00)

One plastic VOO operates on one object by causing it to stretch, squash, bend and twist. The VOO is animated via a CGAL script. Non-default parameters: elongation $\varepsilon = 1.5$, flexibility $\gamma =$

1.3, influence field width (A, B) = 0.5, influence field cap κ = 0.1, VOO type = Polar, spatial strain damping graph (see figure 9.2).

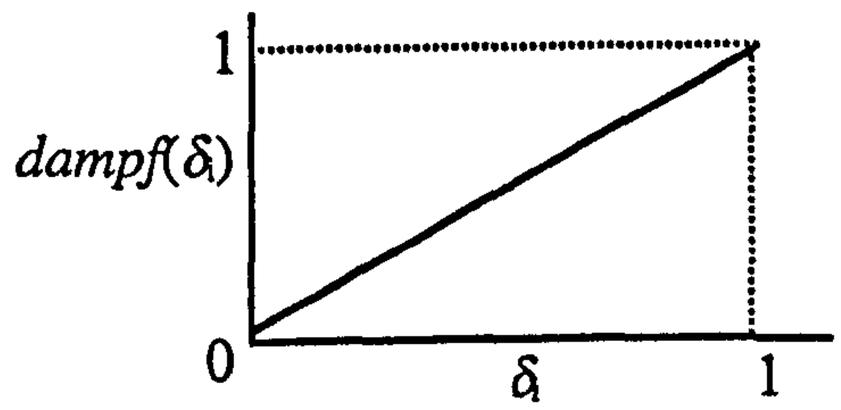


Figure 9.2 Spatial strain damping graph for animation tests 1, 2 and 3.

Animation Test 2 Plastic VOODOO

(see video tape, time code: 10:04:19:00)

Two plastic VOOs with partially overlapping influence fields operate on one object. The two VOOs are animated via motion path constraints attached to the tips of the VOOs. Non-default parameters: elongation ε = 2, flexibility γ = 1.5, influence field width (A, B) = 0.4, influence field cap κ = 0.1, VOO_1 type = Polar, VOO_2 type = Polar, spatial strain damping graph (see figure 9.2).

Animation Test 3 Plastic VOODOO

(see video tape, time code: 10:04:32:00)

Two plastic VOOs operate on one object. The two VOOs are animated via a CGAL script and their influence fields appear as semi-transparent objects. Non-default parameters: influence field width (A, B) = 0.75, VOO_1 type = Polar, VOO_2 type = Length, spatial strain damping graph (see figure 9.2).

Animation Test 4 Elastic VOODOO

(see video tape, time code: 10:04:46:00)

One elastoplastic VOO operates on one object. The stretching of the VOO is animated via a CGAL script. The elastic restoration process, generated by the elastic model of VOODOO results in a decayed oscillation. Non-default parameters: flexibility γ = 1.3, mass = 50, elastic limit e_e = 1 x object length, fracture limit e_f = 10 x object length, duration of elastic restoration (t_{max} - t_0) = 190 frame steps, VOO type = Polar, spatial strain damping and stress / strain graphs

(see figure 9.3).

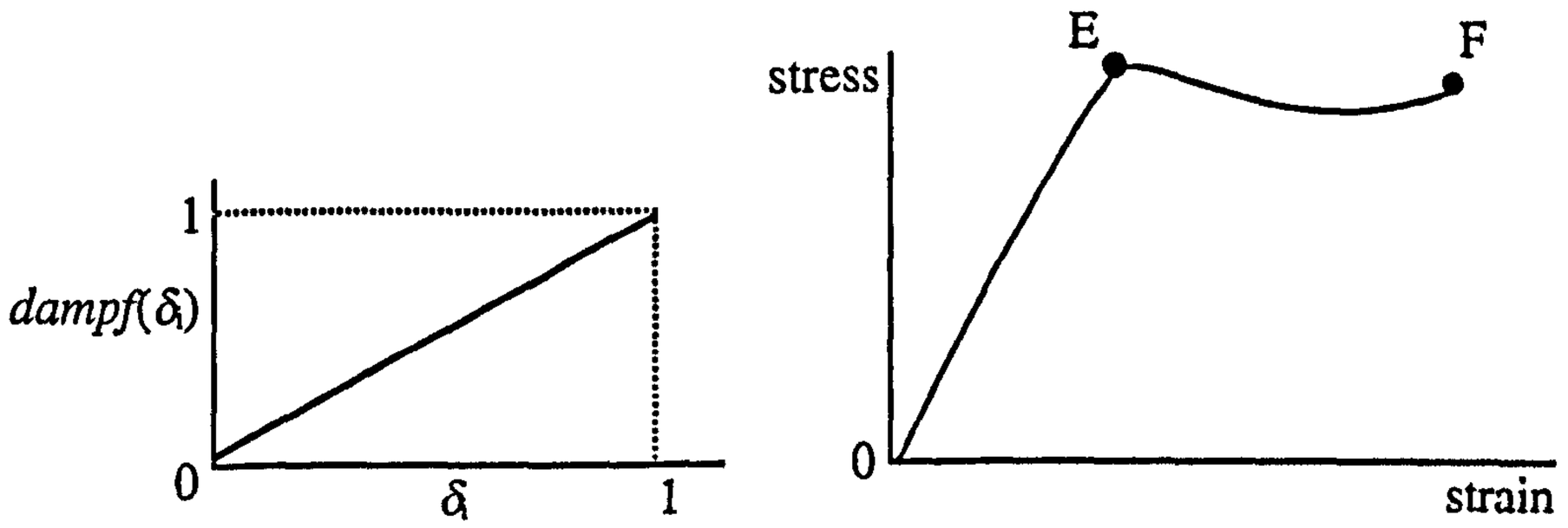


Figure 9.3 Spatial strain damping and stress / strain graphs for animation tests 4 and 7.

Animation Test 5 Elastoplastic VOODOO

(see video tape, time code: 10:04:59:00)

One elastoplastic VOO operates on one object. The stretching of the VOO is animated via a CGAL script. The elastic restoration process, generated by the elastic model of VOODOO results in a smooth return to the original shape. Non-default parameters: flexibility $\gamma = 1.3$, mass = 100, elastic limit e_e = 1 x object length, fracture limit e_f = 10 x object length, duration of elastic restoration ($t_{max} - t_0$) = 190 frame steps, VOO type = Polar, spatial strain damping and stress / strain graphs (see figure 9.4).

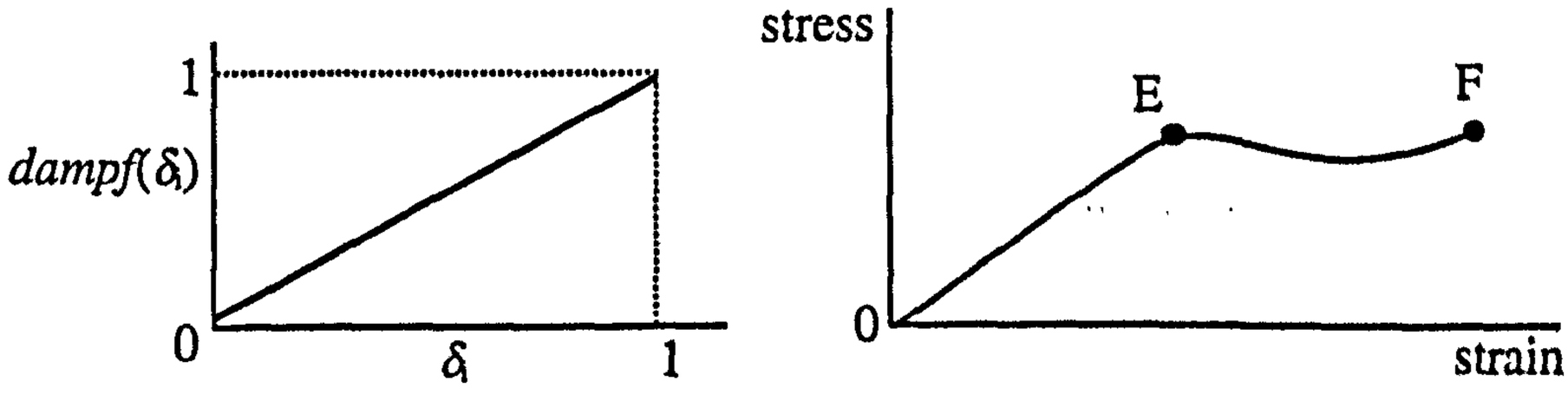


Figure 9.4 Spatial strain damping and stress / strain graphs for animation tests 5 and 6.

Animation Test 6 Elastoplastic VOODOO

(see video tape, time code: 10:05:12:00)

Two elastoplastic VOOs operate on one object. The stretching of the two VOOs is animated via a CGAL script. The elastic restoration process is generated by the elastic model of VOODOO. Non-default parameters: VOO_1: flexibility $\gamma = 1.3$, mass = 100, elastic limit e_e = 1 x object length, fracture limit e_f = 10 x object length, duration of elastic restoration ($t_{max} - t_0$) = 125 frame steps, VOO type = Polar, VOO_2: flexibility $\gamma = 1.3$, mass = 100, elastic limit e_e = 1 x object length, fracture limit e_f = 10 x object length, duration of elastic restoration ($t_{max} - t_0$) = 125 frame steps, VOO type = Polar, spatial strain damping and stress / strain graphs (see figure 9.4).

Animation Test 7 Elastoplastic VOODOO

(see video tape, time code: 0:05:25:00)

Two elastoplastic VOOs operate on one object. The stretching of the two VOOs is animated via a CGAL script. The elastic restoration process is generated by the elastic model of VOODOO. Non-default parameters: VOO_1: flexibility $\gamma = 1.3$, mass = 50, elastic limit e_e = 1 x object length, fracture limit e_f = 10 x object length, duration of elastic restoration ($t_{max} - t_0$) = 125 frame steps, VOO type = Polar, VOO_2: flexibility $\gamma = 1.3$, mass = 50, elastic limit e_e = 1 x object length, fracture limit e_f = 10 x object length, duration of elastic restoration ($t_{max} - t_0$) = 125 frame steps, VOO type = Polar, spatial strain damping and stress / strain graphs (see figure 9.3).

Animation Test 8 Viscoelastic VOODOO

(see video tape, time code: 10:05:38:00)

One viscoelastic VOO operates on one object. The VOO is animated via a CGAL script and the strain creep effect is generated by the viscoelastic model of VOODOO. Non-default parameters: viscosity $\nu = 1.1$, strain creep $\mu = 0.5$, duration of strain creep ($t_{max} - t_0$) = 250 frame steps, VOO type = Polar, spatial strain damping, viscous strain rate and strain creep graphs (see figure 9.5).

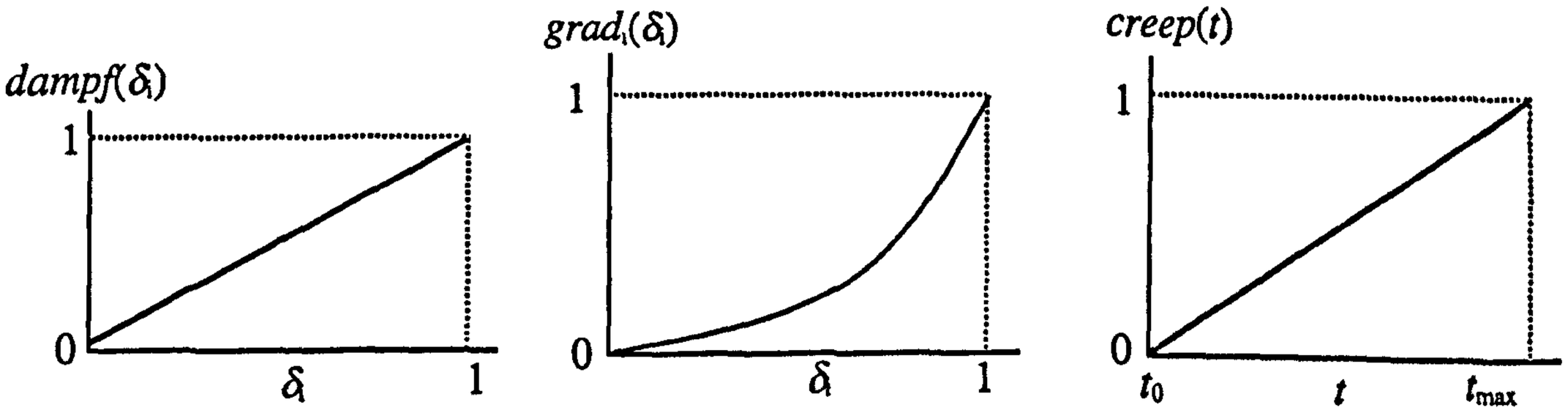


Figure 9.5 Spatial strain damping, viscous strain rate and strain creep graphs for animation test 8.

Animation Test 9 Viscoelastic VOODOO

(see video tape, time code: 10:06:01:00)

One viscoelastic VOO operates on one object. The VOO is animated via a CGAL script and the strain creep effect is generated by the viscoelastic model of VOODOO. Non-default parameters: flexibility $\gamma = 1$, viscosity $\nu = 1.1$, strain creep $\mu = 0.1$, duration of strain creep $(t_{max} - t_0) = 300$ frame steps, VOO type = Polar, Tip field active, spatial strain damping, viscous strain rate and strain creep graphs (see figure 9.6).

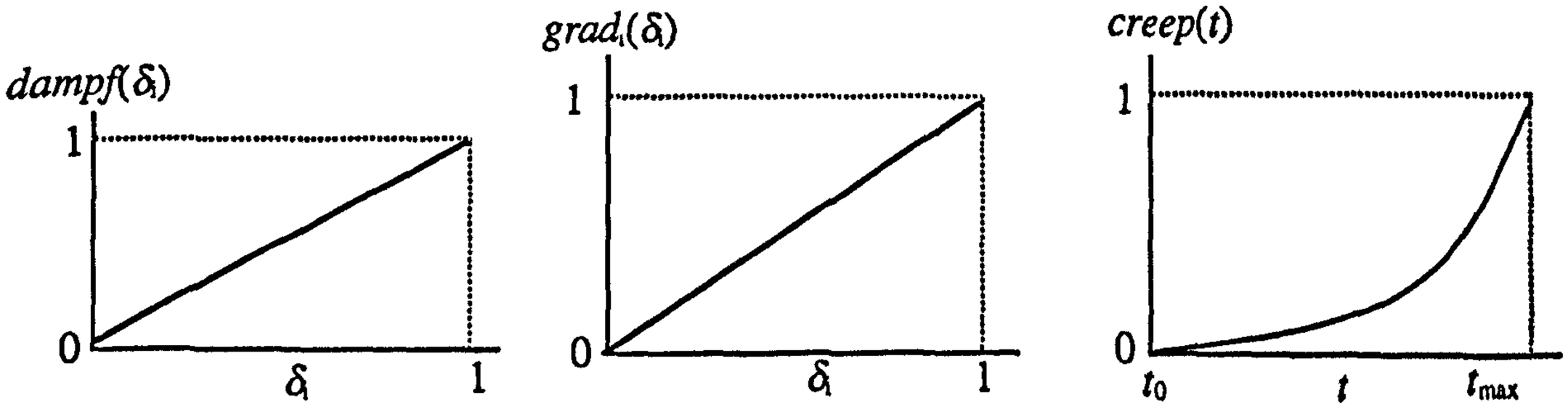


Figure 9.6 Spatial strain damping, viscous strain rate and strain creep graphs for animation tests 9 and 10.

Animation Test 10 Viscoelastic VOODOO

(see video tape, time code: 10:06:14:00)

One viscoelastic VOO operates on one object. The VOO is animated via a CGAL script and the strain creep effect is generated by the viscoelastic model of VOODOO. Non-default parameters: flexibility $\gamma = 1$, viscosity $\nu = 1$, strain creep $\mu = 0.65$, duration of strain creep $(t_{max} - t_0) = 300$

frame steps, VOO type = Polar, Head field active, spatial strain damping, viscous strain rate and strain creep graphs (see figure 9.6).

Animation Test 11 Viscoelastic VOODOO

(see video tape time code: 10:06:27:00)

One viscoelastic VOO operates on one object. The VOO is animated via a CGAL script, the strain creep effect is generated by the viscoelastic model and the elastic restoration is generated by the elastic model of VOODOO. Non-default parameters: mass = 100, viscosity $\nu = 1.9$, strain creep $\mu = 0.1$, duration of strain creep ($t_{max} - t_0$) = 400 frame steps, VOO type = Polar, spatial strain damping, viscous strain rate, strain creep and stress / strain graphs (see figure 9.7).

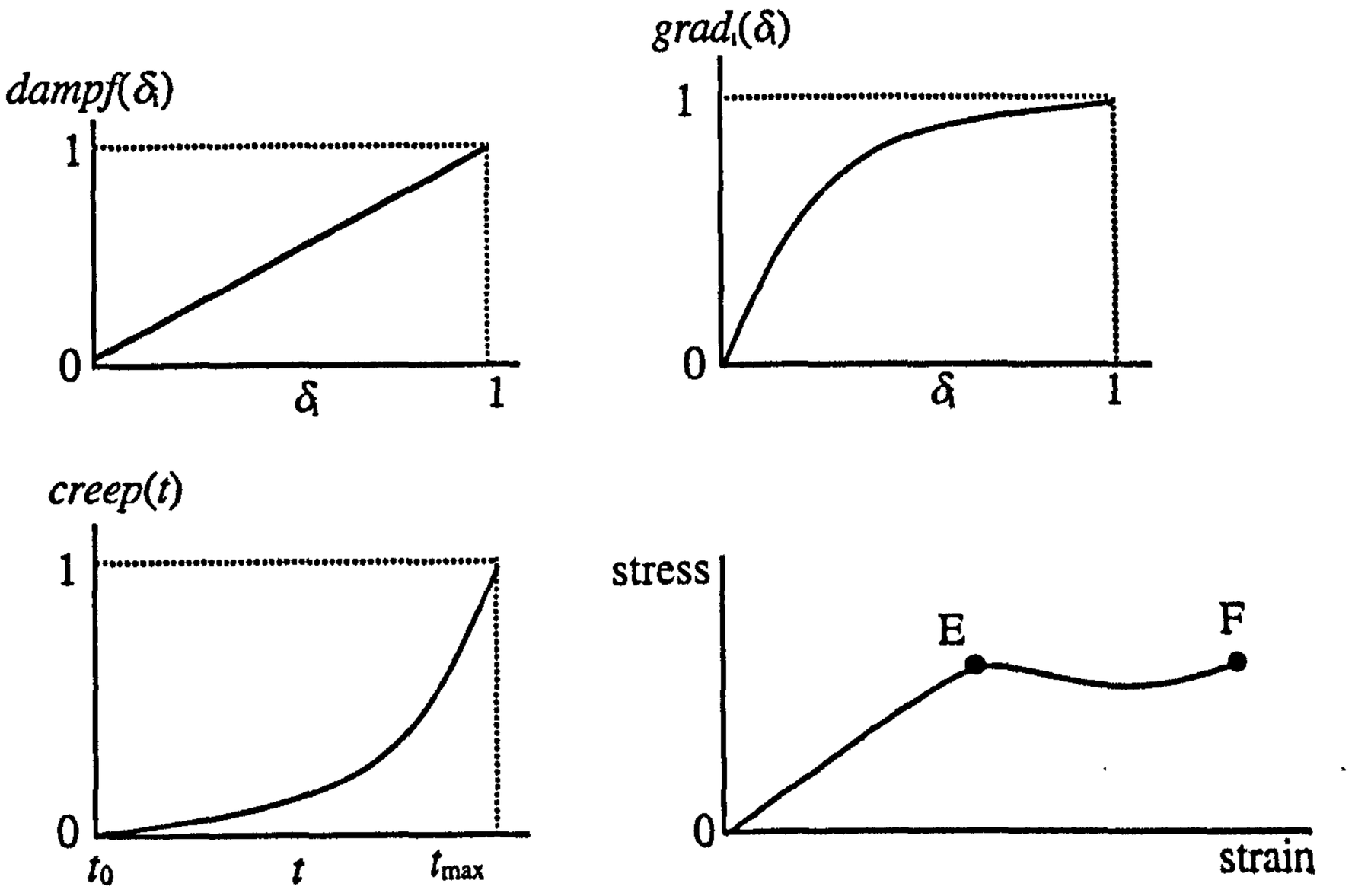


Figure 9.7 Spatial strain damping, viscous strain rate, strain creep and stress / strain graphs for animation test 11.

Animation Test 12 Plastic VOODOO with Motion Path Constraints

(see video tape, time code: 10:06:45:00)

One plastic VOO operates on one object. The VOO is animated using two motion path constraints attached to its head and tip. Non-default parameters: flexibility $\gamma = 1.6$, weight $\omega = 1.5$, VOO type = Polar, spatial strain damping and momentum graphs (see figure 9.8).

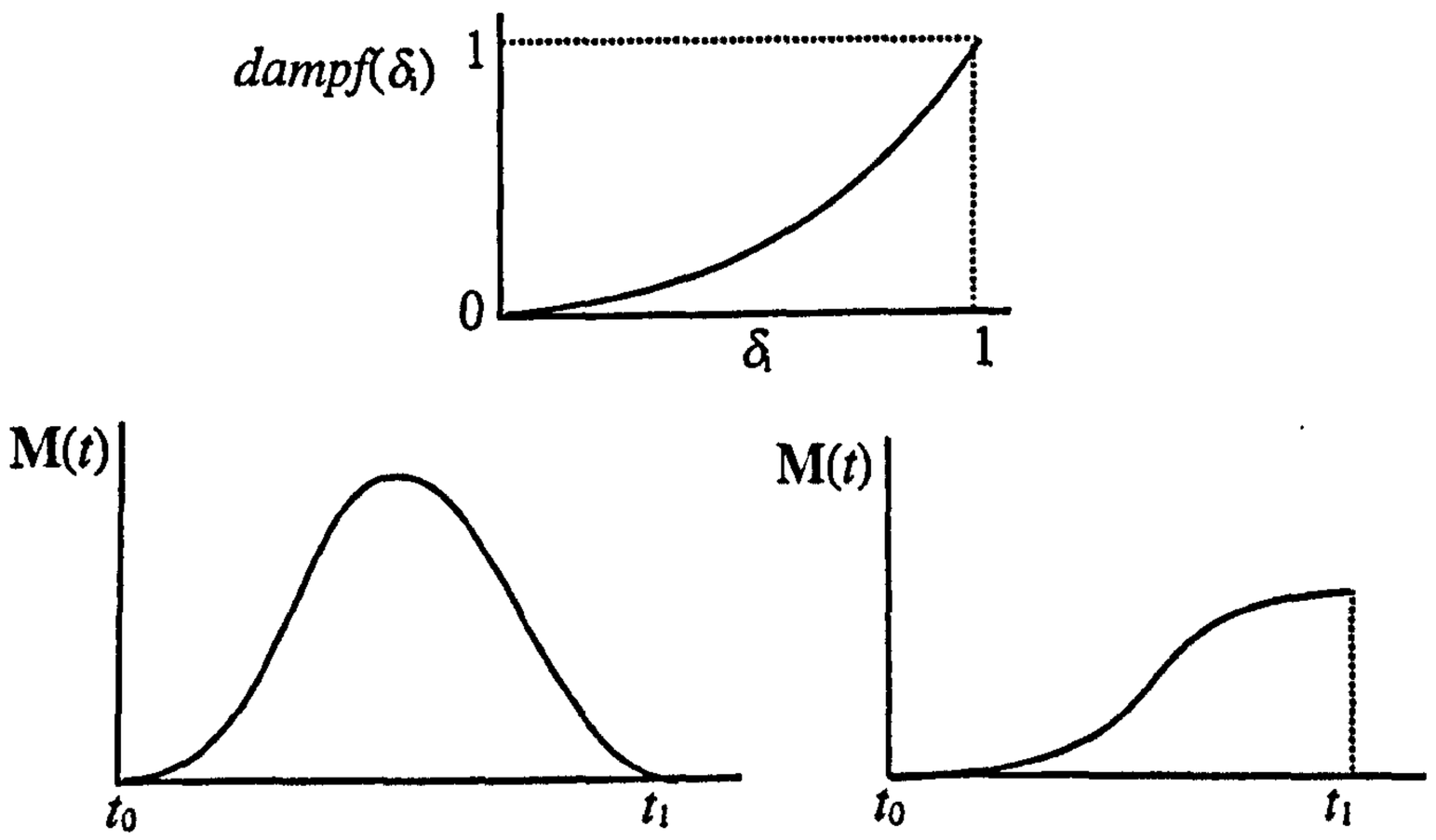


Figure 9.8 Spatial strain damping and momentum graphs for animation test 12.

Animation Test 13 Plastic VOODOO with Articulated VOOs

(see video tape, time code: 10:06:58:00)

Three plastic VOOs operate on one object. The three VOOs are connected with two arthra. Non-default parameters: flexibility $\gamma = 0$, field cap $\kappa = 0.75$, field width $(A, B) = 0.3$, VOO type = Polar, strain damping graph (see figure 9.2).

Animation Test 14 Viscoelastic VOODOO with Motion Path Constraints and Articulated VOOs

(see video tape, time code: 10:07:11:00)

Two viscoelastic VOOs operate on one object. The two VOOs are connected with one arthron which is attached to a motion path constraint. Non-default parameters: elongation $\varepsilon = 0.7$, field cap $\kappa = 0.75$, field width $(A, B) = 0.3$, strain creep $\mu = 0.9$, viscosity $\nu = 0.6$, duration of strain creep = 250, VOO type = Polar, spatial strain damping, viscous strain rate and strain creep graphs (see figure 9.9).

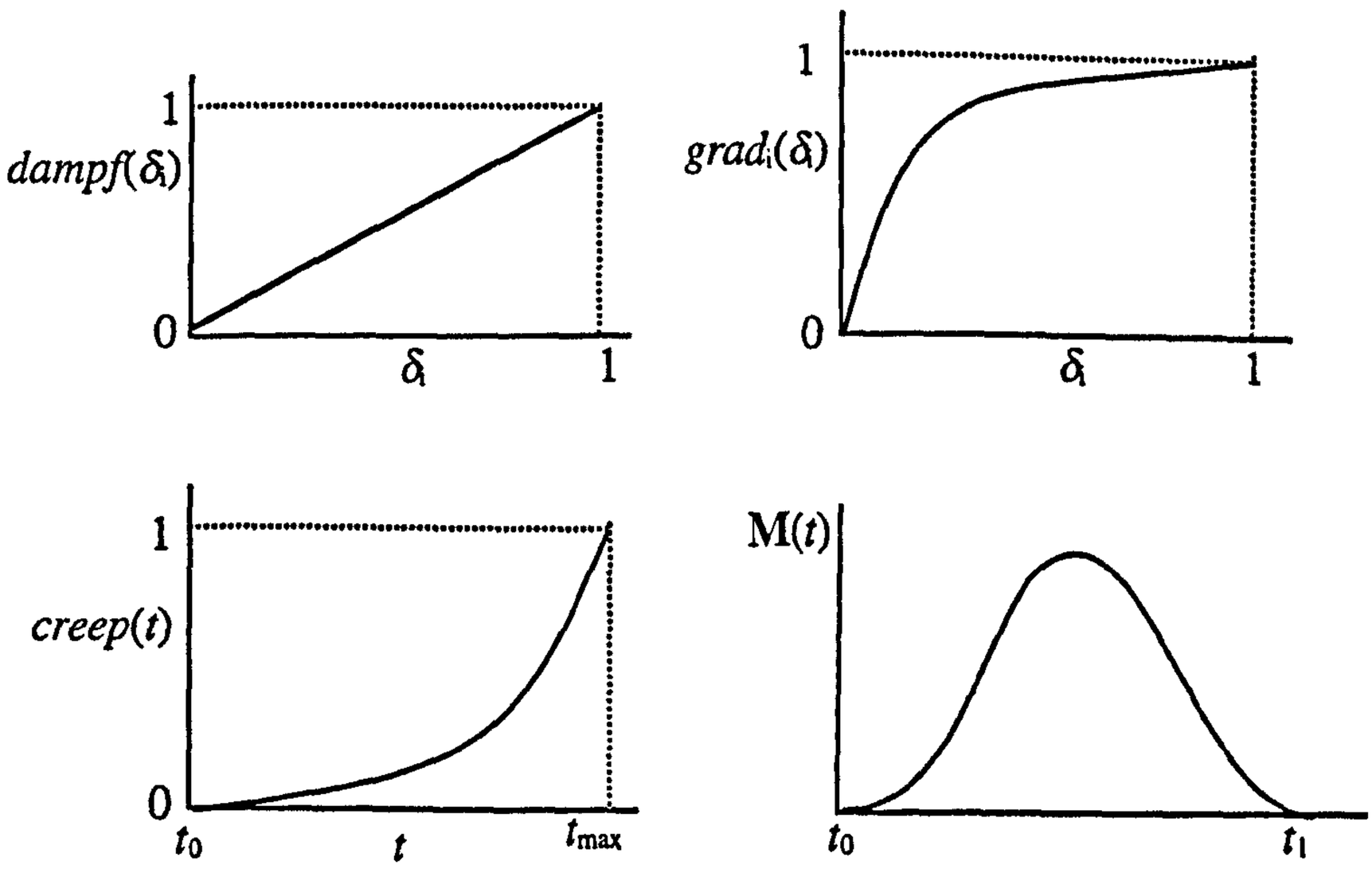


Figure 9.9 Spatial strain damping, viscous strain rate, strain creep and momentum graphs for animation test 14.

Animation Test 15 Viscoelastic VOODOO with Plexus of VOOs

(see video tape, time code: 10:07:24:00)

Three viscoelastic VOOs operate on one object. The three VOOs are connected with three arthra to form a plexus cell. Non-default parameters: strain creep $\mu = 0.5$, viscosity $\nu = 0.7$, duration of strain creep = 250, VOO type = Polar, spatial strain damping, viscous strain rate and strain creep graphs (see figure 9.10).

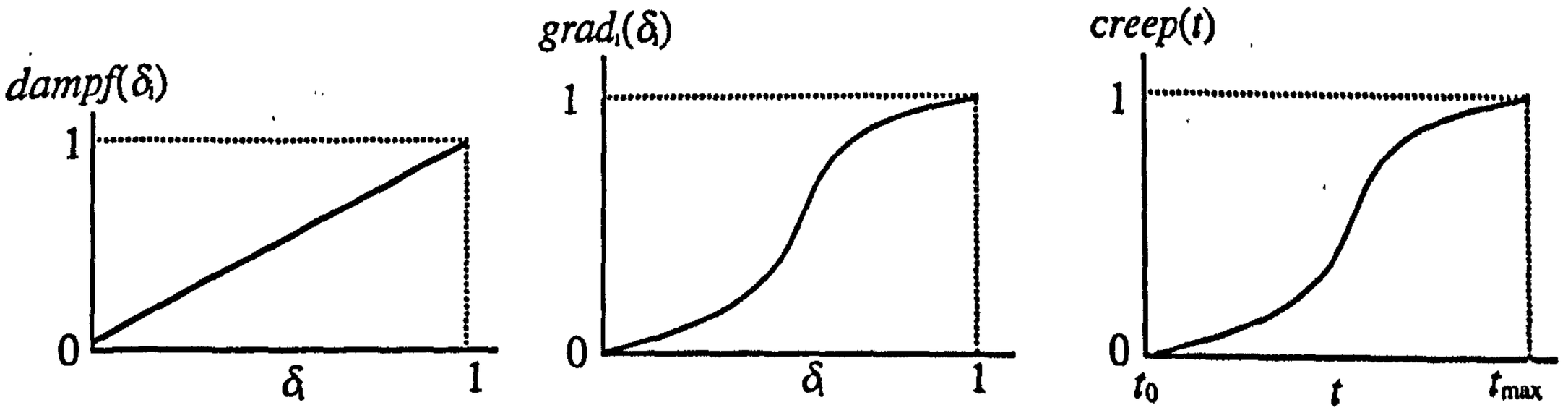


Figure 9.10 Spatial strain damping, viscous strain rate and strain creep graphs for animation test 15.

Animation Test 16 Walking Figure

(see video tape, time code: 10:07:37:00)

A set of five viscoelastic VOOs operate on a walking figure.

Animation Test 17 Beach Ball

(see video tape, time code: 10:07:53:00)

A set of five viscoelastic VOOs operate on a beach ball.

Animation Test 18 Giant Squid

(see video tape, time code: 10:08:24:00)

A set of eleven viscoelastic VOOs operate on a giant squid model.

9.4.2.2 Incredible Suckers

(see video tape, time code: 10:08:38:00)

(see colour plates 9.20 through to 9.26 in section 9.7 and 8.26 through to 8.28 in section 8.6)

Incredible Suckers is a documentary about cephalopods (squid, octopodes, cuttlefish, etc.) produced by Oxford Scientific Films Ltd and commissioned by the BBC. The candidate participated as a team leader in a production team which produced an animated sequence featuring a giant squid and a sperm whale.

The giant squid model was created using CGAL modelling routines and a triangulation polygon converter based on an algorithm developed by the candidate (see appendix A). A squid head was connected to ten different tentacles to form one continuous solid. This was then attached to a squid body object. All of the squid parts were created using the a polygon skinning technique. The sperm whale model was obtained using a 3D laser scanner CYBERWARE. The scale model of the whale was provided by the Natural History Museum in London.

Animating the fight between these two creatures was the most difficult scenes the animation team had to develop during the production. Both characters were animated using VOODOO organic deformation tools. A set of ten VOOs was used for applying animated deformation to

each of the ten tentacles of the squid independently and one VOO was used to control the deformations on the squid body and head together. Deformation of the sperm whale body was also based on a set of organic VOOs. One VOO was used to control the front half of the body and the head and another VOO controlled the rear half of the body and the tail. Two pairs of VOOs controlled the flippers and the tail flukes and a single VOO controlled the lower jaw of the whale's mouth. Both squid and whale models were treated as viscoelastic masses and were subjected to strain creep every time a deformation was applied. This gave an organic living form feel to the entire animation.

The documentary Incredible Suckers was at first broadcast in January 1996 by BBC2 as part of the "Natural World" series. It was then broadcast in 1997 by 13-WNET in the USA. The film has won an award at the Wildscreen Film Festival in Bristol (1996). The animation "Incredible Suckers" has since been broadcast by Channel 4 as part of the "Absolutely Animals" series, by RTL in Germany as part of the "Welt der Wunder" series and by Canal+ in France and it has participated in the Bit.Movie '96 animation festival in Italy.

9.4.2.3 after birth (excerpts)

(see video tape, time code: 10:10:24:00)

(see colour plates 9.27 through to 9.32 in section 9.7)

After birth is an animation exercise on life and the evolution of life, juxtaposing myth, reality and dream. The myth of the child bearer stork is interrupted by a fictitious underwater evolution chain. In after birth there is a blue whale and a stork character which have been animated using an early version of the VOODOO organic modelling routines. The set of VOOs used to control the blue whale was very similar to the ones used for the sperm whale (see section 9.4.2.2). Deforming the stork character was achieved just with a pair of VOOs operating on the wings and body of the stork. VOOs were also used to achieve the wave effect on the water surface. A sinusoidal wave was applied as an expression to the ends of several VOOs which applied their deformation effect on a seascape polygon mesh. Air bubbles emanating from the whale's nostrils were also animated by applying one VOO on each element of an array of deformable particles.

nostrils were also animated by applying one VOO on each element of an array of deformable particles.

After birth has gained awards at the following computer graphics festivals: Jazz.Bit, Pori, Finland 1994, Pixel Art EXPO, Rome, Italy 1994 and EUROGRAPHICS, Cambridge, UK 1992. After birth was also screened at: Machida City Museum of Graphic Arts, Machida City, Japan 1993, Image du Futur, Montreal, Canada 1993, Montage - International Festival of the Image, Rochester NY, USA 1993, European Students Computer Graphics Festival, Montreuill, France 1993, Bit.Movie, Riccione, Italy 1993, SIGGRAPH screening room, Chicago IL, USA 1992 and London Film Festival, London, UK 1992.

9.4.2.4 Gullaesthetics

(see video tape, time code: 10:11:16:00)

In Gullaesthetics a flock of seagulls appears in flight above a seascape. Each seagull was animated using a pair of VOOs for the wings. Waves on the seascape were also generated by applying sinusoidal expressions to the ends of VOOs. Gullaesthetics was screened at the London Film Festival in 1991.

9.5 Results

9.5.1 The Effectiveness of VOODOO

A vector offset operator is an interactive strain application tool which serves as an interface to deformation. VOOs create an intermediate layer between user and deformation process, which makes the VOODOO generic tool independent from the underlying deformation technique and the object representation model. In the context of this study VOOs where linked to a local deformation model and operated upon polygonal data. This resulted in a simpler and more

efficient approach to deformation. However, the suggested tool may easily be linked to a free form deformation technique and a volumetric or parametric surface representation. So the true contribution of the VOODOO tool is concentrated on the specification of organic material properties and spatiotemporal constraints.

VOOs may be used in conjunction with conventional key framing animation techniques. By defining key states for a VOO at specified times of an animation and by interpolating between these key states of a VOO one can generate smooth animated deformation. Another approach is to apply procedural animation to the time related state of a VOO using a scripting language.

The elastoplastic model is based on a user defined stress / strain graph. This enables the customisation of the material behaviour of an object according to data derived from real-life experiments. Alternatively, the user may experiment by modifying the shape of a stress / strain graph and visually customise the elastoplastic behaviour of an object. The user may also apply time constraints to the elastic restoration process by defining an energy damping graph and by directly controlling the duration of the process. Other user controlled parameters are the property of mass and the elastic and fracture strain limits of the material. Manipulation of these graphs and parameters provides a mechanism which is capable of generating a whole range of natural and artificial elastoplastic material behaviours. An advantage of the elastoplastic model is that it offers the freedom to design the elastoplastic material behaviour. Another advantage of the model is that VOOs are being used as an intermediate layer between the elastoplastic model and the deformation model. This allows the application of elastoplastic parameters directly on VOOs. An elastically restored VOO transfers the deformation inside its field of influence and onto the influenced objects. This two layer approach helps separate the deformation problem from the elastoplastic behaviour problem and it greatly improves the efficiency of the overall model.

The viscoelastic model provides a mechanism for the implementation of the three characteristic viscoelastic features of strain creep, stress relaxation and stress / strain hysteresis. A spatial gradient of strain rate is used to generate a viscous effect in a deformable object. The user is offered the freedom to customise the viscous behaviour of an object by directly specifying the

strain rate gradient and by manipulating the value of the viscosity coefficient. An object may be made to respond like a thin, runny fluid or a thick, sticky fluid or at the other extreme as an elastoplastic material with no viscosity at all. A strain creep graph provides a temporal constraint for the development of viscoelastic creeping strain inside the volume of an object. The strain creep mechanism is used to transform the initially viscous behaviour of an object into common elastoplastic behaviour over a user specified period of time. With this mechanism the user may customise the material behaviour of an object, from perfectly viscous to viscoelastic and finally to elastoplastic. A stress relaxation graph is used to diminish the internal elastic stresses over a period of time. An hysteresis graph is superimposed on top of the stress / strain graph of a material in order to describe the hysteresis loop during the loading and unloading process. The viscoelastic model is capable of reproducing an approximation of real organic material behaviour in deformable objects. This may be achieved by incorporating strain creep, stress relaxation and hysteresis graphs derived from experiments on real organic materials. Such graphs may be obtained from existing literature and may be calibrated and modified to conform to specific animation constraints.

Compared to conventional animation techniques, such as key-framing, the viscoelastic model offers a greater degree of freedom. The animator is freed from the burden of having to specify key frames for every minute movement in an organically deforming object. By defining or modifying existing creep, relaxation and hysteresis graphs, the user has a fine and high level control over the shape and the temporal behaviour of the deformation. Thus, the viscoelastic model offers extensive control of complex organic material behaviour and it may be used to automate the animation process of organically deformable objects. Compared to physically based simulation models of viscoelasticity, the suggested viscoelastic model has the advantage of efficiency. The introduced model results to a fast implementation which is general, reusable, extendible, independent of geometric representation and capable of generating realistic looking, organic material behaviour.

The functionality of the VOO generic deformation tool can be extended to a constraint application tool. So, a VOO may also serve as a carrier of spatiotemporal dynamic constraints in conjunction with geometric and physical deformation parameters. This two layer approach, yet

suitable for constructing animated characters based on vertebrate living creatures. Sets of VOOs may also be connected with arthra to form closed triangular deformation cells. Many such cells may be combined to create a plexus of VOOs. A plexus may be used as an underlying deformation mechanism for objects with fluid motion or animated characters based on invertebrate living creatures.

Motion paths are spatial paths equipped with momentum / time graphs and initial and final time constraints. A motion path serves as a combined spatial and temporal constraint and it can be applied to the head, the tip or the aiming point of a VOO. A motion path may also be applied to an arthron of a skeleton or of a plexus of VOOs. Motion paths offer a versatile animation medium and may be used to solve a great variety of animation problems. The combination of skeletal structures with motion paths and the elastoplastic and viscoelastic properties of VOOs can lead to the construction of animal locomotion mechanisms suitable for animation of organic characters. Motion paths, however, require numerical solution of ordinary differential equations which introduces a considerable computational overhead. Since mass is considered to be concentrated on the ends of VOOs these calculations need only be performed once per motion path per time step. This improves the performance of the dynamic constraints model. Another way of constraining VOOs in space and time is through vector fields. The spatial and temporal behaviour of sets of VOOs may be defined using vector field functions. Vector fields of VOOs may be used to simulate the effect of wind, tornadoes, running water and other natural phenomena.

9.5.2 The Performance of VOODOO

The efficiency of the plastic model is primarily based on the implementation of a simple local deformation model. The main objective of the model is to calculate a plastic strain vector per active vertex, per active VOO per time step. Moreover, the involvement of ellipsoids as fields of influence offers a very efficient mechanism for identifying the set of active vertices. Precalculation of a list of damping elements per active vertex per active VOO further improves the efficiency of the plastic model.

The efficiency of the elastoplastic model is based on the fact that the elastic restoration process is only applied to the time related state of a VOO. So, the process does not attempt to restore individual vertices. A further element which improves the efficiency of the model is the use of a simplified method for deriving the elasticity coefficient and the permanent plastic strain from a stress / strain graph. Also, a simple integration is used for the calculation of the restored state of a VOO. The local deformation model is then activated to restore the deformed objects.

The efficiency of the viscoelastic model is based on the precalculation of a spatial gradient for the viscous strain rate per active vertex of an object. A further improvement on efficiency is achieved by calculating the strain creep and stress relaxation factors once per time step. This adds only a minor computation overhead to the model. The integration time step may be increased to two or more frames in order to further speed up the integration process. This may affect the smoothness of a deforming object but it enables interaction with viscoelastic objects of considerable size. Reducing the time step to a fraction of a frame improves the smoothness of the flow of deformation during the viscoelastic creep of strain. However, setting the time step to values less then half a frame will dramatically decrease the performance of the model. This may be acceptable at rendering time but not at interactive time.

The efficiency of the dynamic constraints model is based on the fact that mass is considered to be concentrated only on the ends of VOOs. So, numerical solution of ordinary differential equations is only performed once per motion path per time step.

9.5.3 The Scope of VOODOO

The VOODOO model is capable of applying spatial deformation to 3D polygonal, parametric surface and volumetric objects but not to implicit surfaces. The presented implementation of the model has only included a polygonal object representation for the sake of brevity and for clearly demonstrating the deformation effects.

The deformation generated by a VOO tool may be directly applied in global or local space. The application may easily be extended, though, to deform free form deformation lattice space. A VOO can cause stretch, squash, bend and twist strain and the volume of a deformed object may be preserved. Shearing strain has not been implemented in the presented application.

External forces may be applied to deformable objects using dynamic motion paths equipped with momentum / time graphs. However, there are no environmental forces like gravity or wind, no angular momentum, no self-penetration avoidance and no collision detection currently incorporated in the model. These issues will be dealt with in future work.

9.5.4 User Experience

The interactive and animation framework of VOODOO linked to the animation software CGAL was used by the candidate along with several other students at the NCCA, for the production of animation tests and a number of student projects. A certain amount of feedback has been received concerning user experience, effectiveness and performance.

Case 1

A student who participated in the project "Incredible Suckers" (see section 9.4.2.2) has used the VOODOO framework for a period of more than three months. The students's involvement in the project included animating the fins of the giant squid character and the ink cloud emerging from the mouth of the squid towards the end of the animation. The student found the use of the animation framework through CGAL scripting non-intuitive but highly effective and efficient.

This particular student, however, had been well accustomed to working with animation scripting.

Case 2

A student was asked to test the interactive framework and the plastic properties of VOODOO by deforming simple and complex 3D objects. The student recognised a certain ease of use of the interactive deformation tool VOO and the creation of local strain on objects. However, the student commented that the facilities for creating new VOO tools, setting the initial parameter values and linking VOOs with joints were very primitive and awkward to use.

Case 3

A student was asked to test the animation framework and the organic properties of VOODOO by creating short animations of viscoelastic objects. The student found the control of organic deformation behaviour through parameters and graphs non-intuitive. The student also realised that to design new stress / strain or strain creep graphs and to effectively control viscoelasticity parameters required a deeper understanding of the theory of deformable materials. However, the student suggested that modifying existing graphs and sets of parameters and instantly evaluating the visual result could lead to a more practical and efficient way of using the system.

In the candidate's own experience, the VOODOO application although effective and efficient, presents serious drawbacks in the interaction interface particularly with the organic model. Intuitive interaction with complex organic material behaviour is a difficult problem which extends beyond the scope of the present research. This issue will be dealt with further in future work.

9.6 Summary

This chapter presented the structure of the test bed application VOODOO. VOODOO consists of the following set of structured models:

- the plastic deformation model,
- the elastoplastic model,
- the viscoelastic model and
- the constraint model.

The functionality of the interactive framework of VOODOO was introduced. Interactive VOODOO may be used for visual experimentation of deformation, for testing the performance of various features of the deformation model and for 3D modelling. The animation framework of VOODOO was also introduced. A sample script illustrated a method of linking VOODOO to the animation software CGAL. CGAL scripts may be used to animate VOOs which in turn deform objects that enter their influence fields. This animation framework was used for the production of several animated sequences, which were, also, reviewed in this chapter. Finally, certain points that stretch the effectiveness and efficiency of the VOODOO model were examined. The next chapter will conclude this thesis.

9.7 Colour Plates

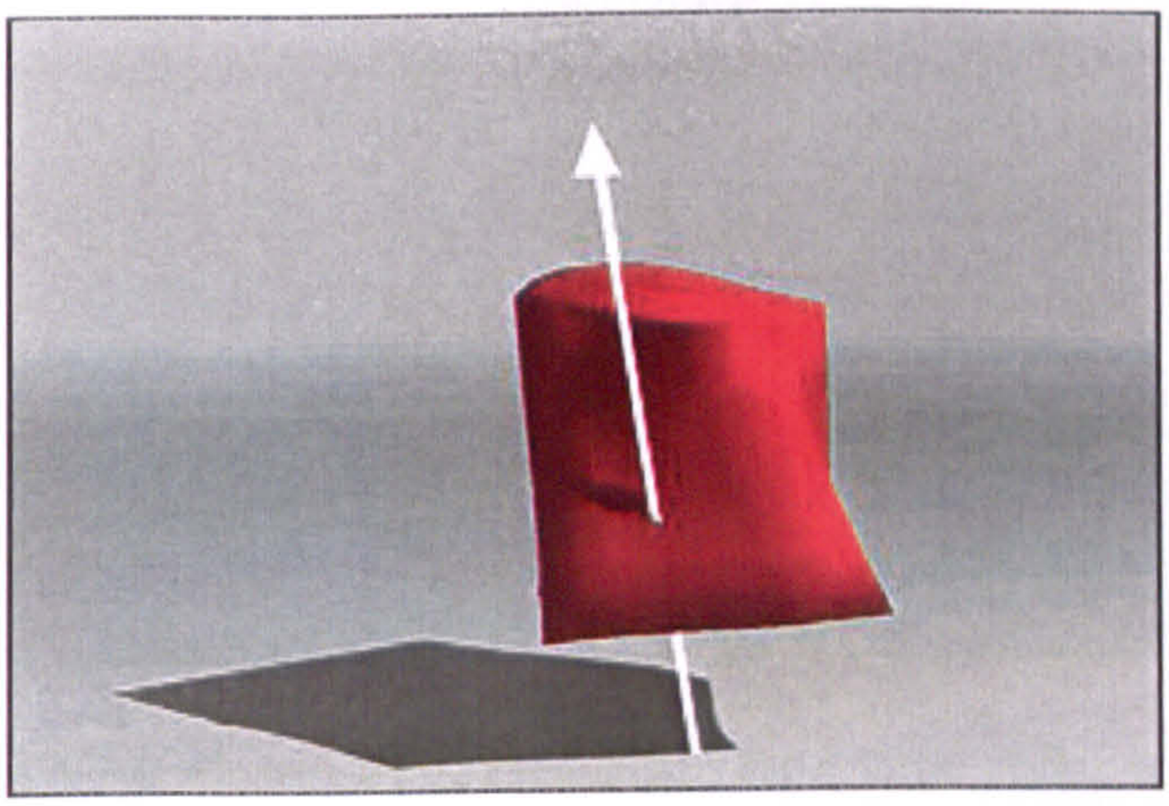


Figure 9.11 A plastic VOO with one object (see video tape, time code: 10:04:05:00).

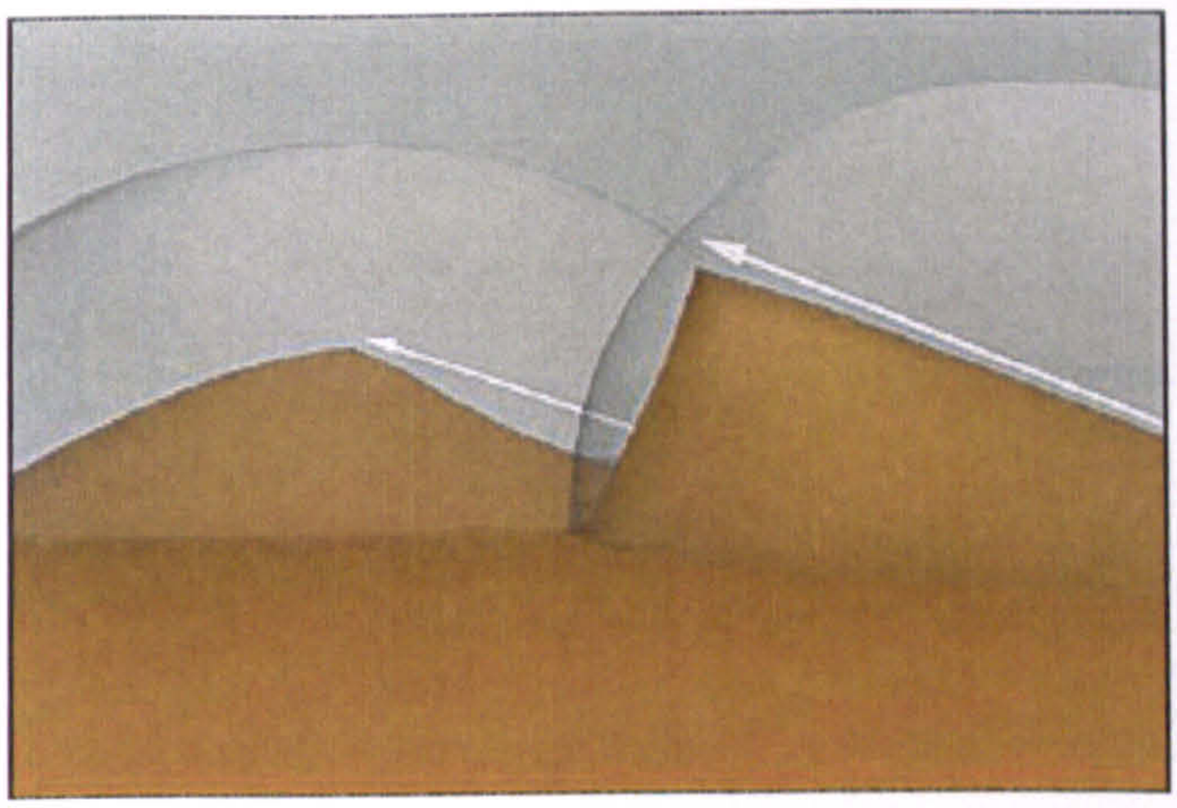


Figure 9.12 A polar VOO and a length VOO (see video tape, time code: 10:04:32:00).

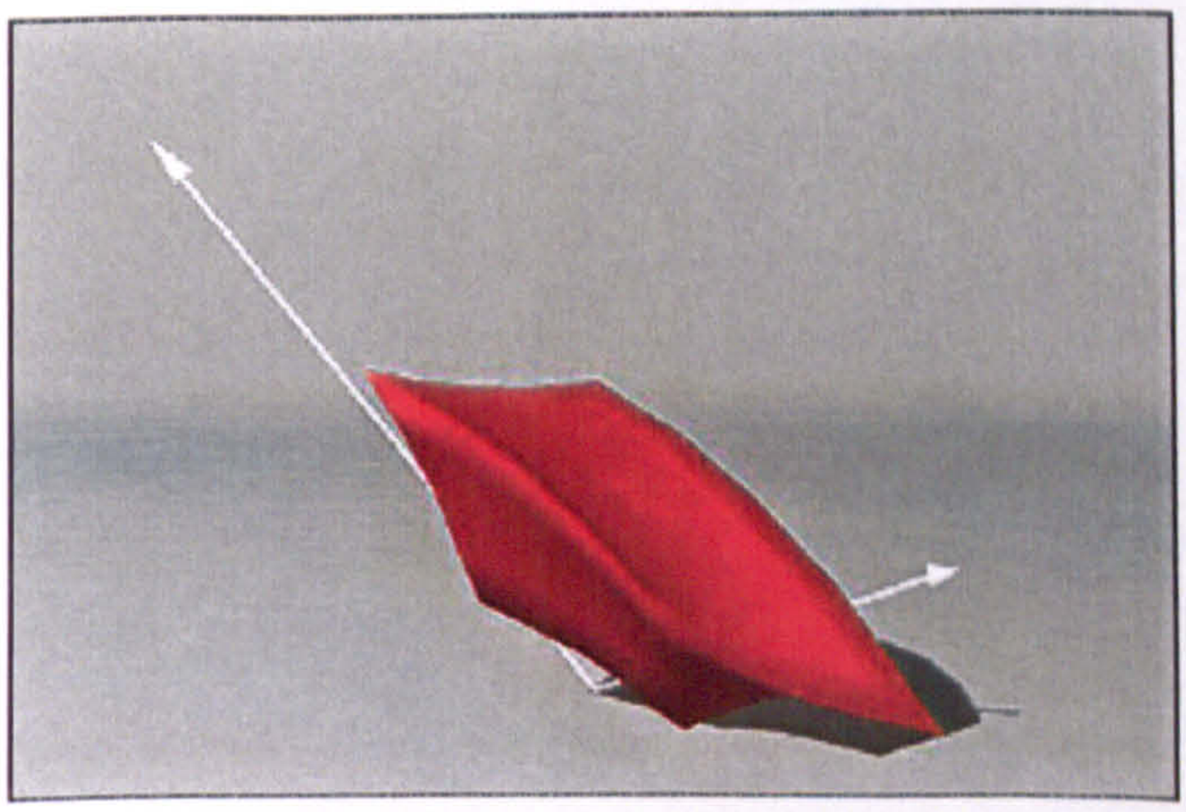


Figure 9.13 Two elastoplastic VOOs with one object (see video tape, time code: 10:05:12:00).

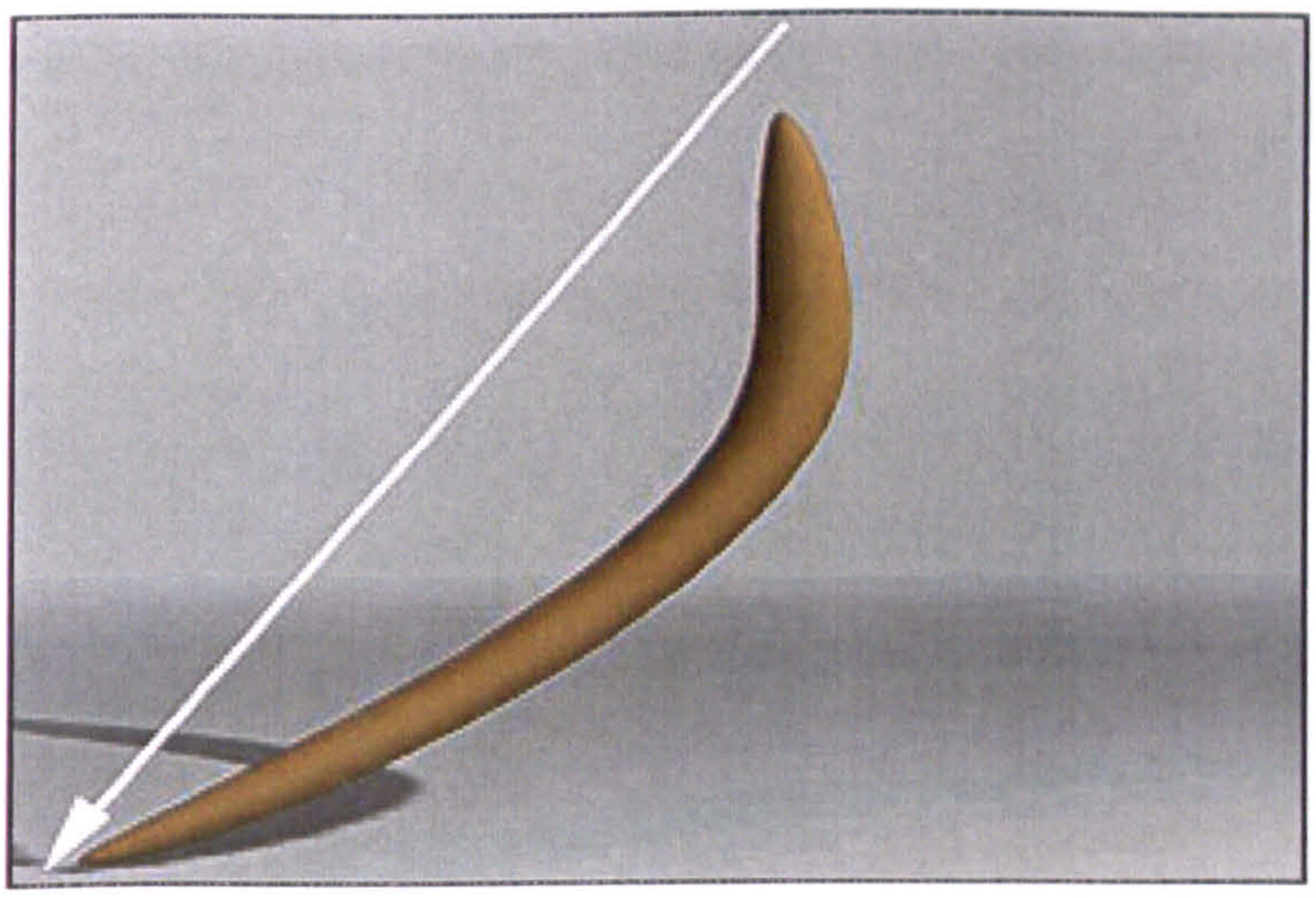


Figure 9.14 Viscoelastic VOO (see video tape, time code: 10:06:01:00).

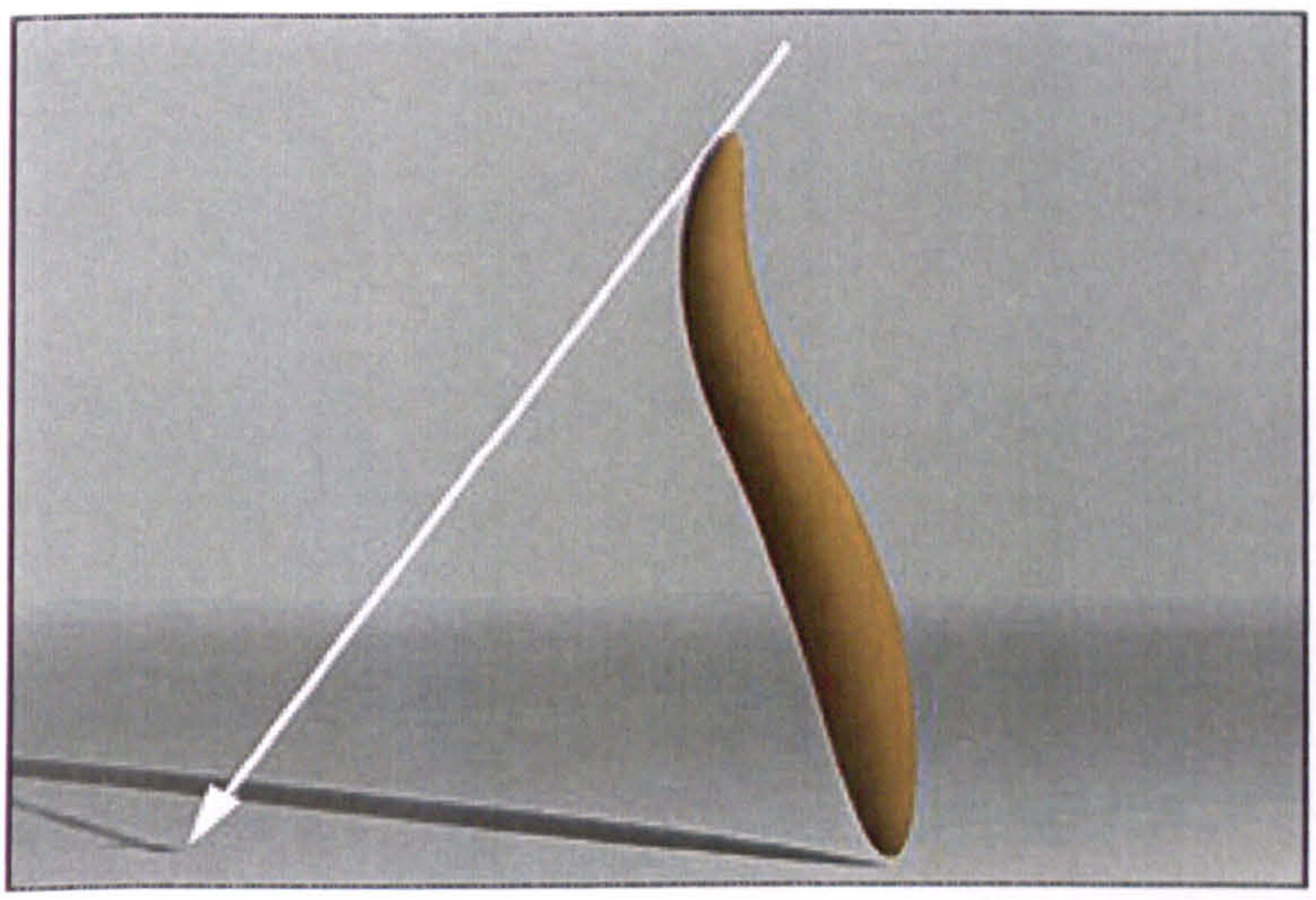


Figure 9.15 Viscoelastic VOO (see video tape, time code: 10:06:14:00).

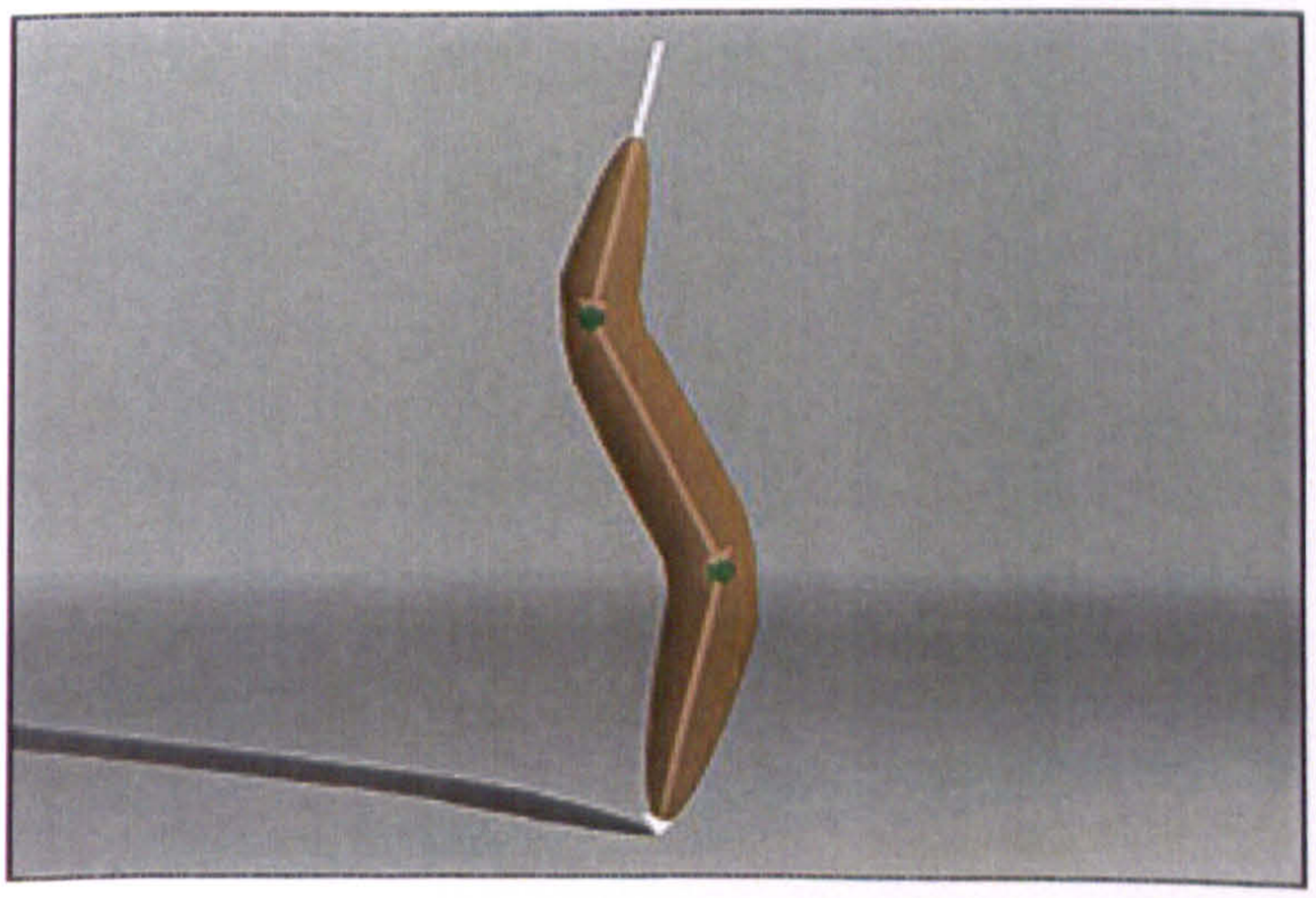


Figure 9.16 Plastic VOOs with arthra (see video tape, time code: 10:06:58:00).

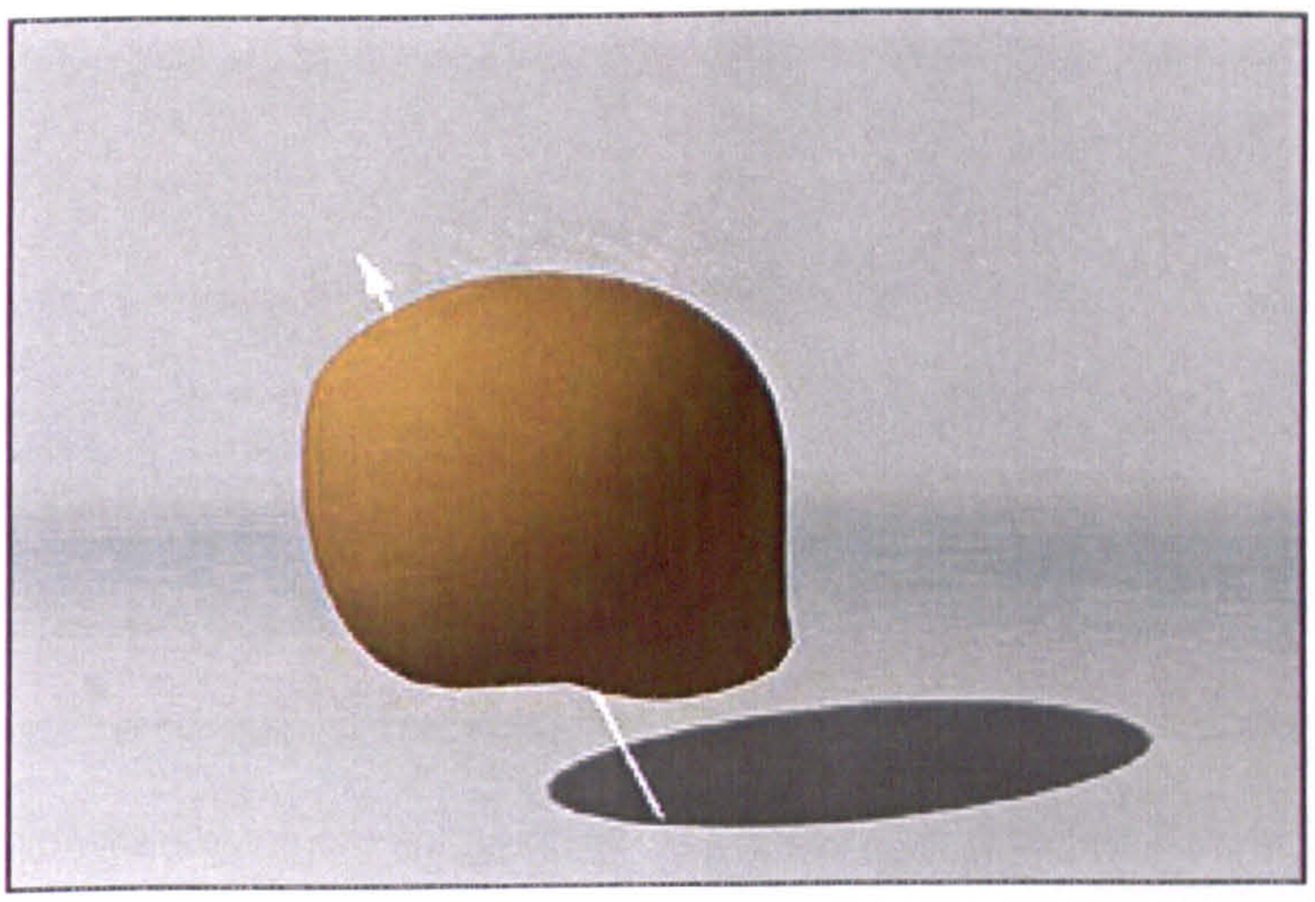


Figure 9.17 Viscoelastic VOO with a sphere (see video tape, time code: 10:05:38:00).

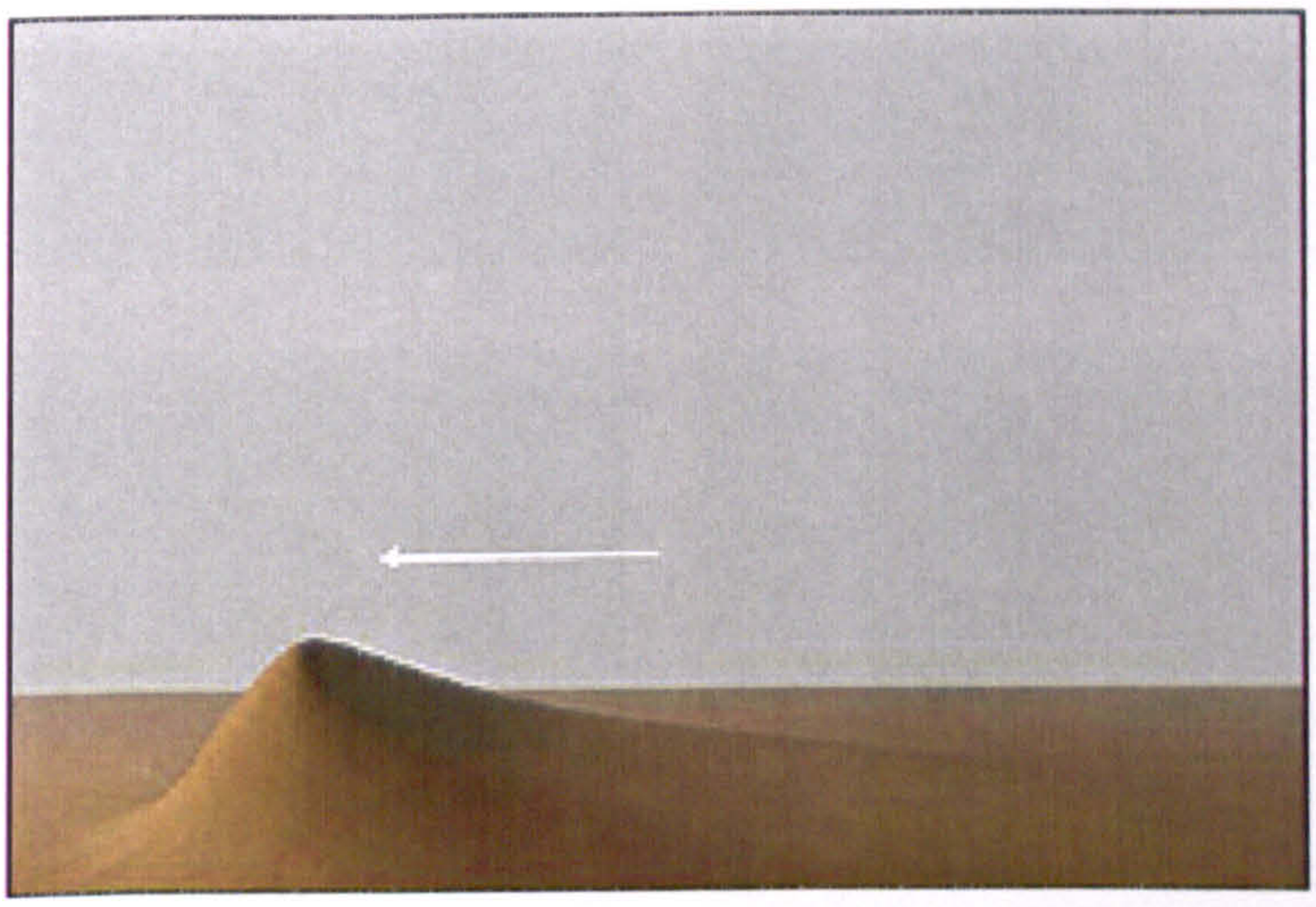


Figure 9.18 Viscoelastic VOO with terrain (see video tape, time code: 10:06:27:00).

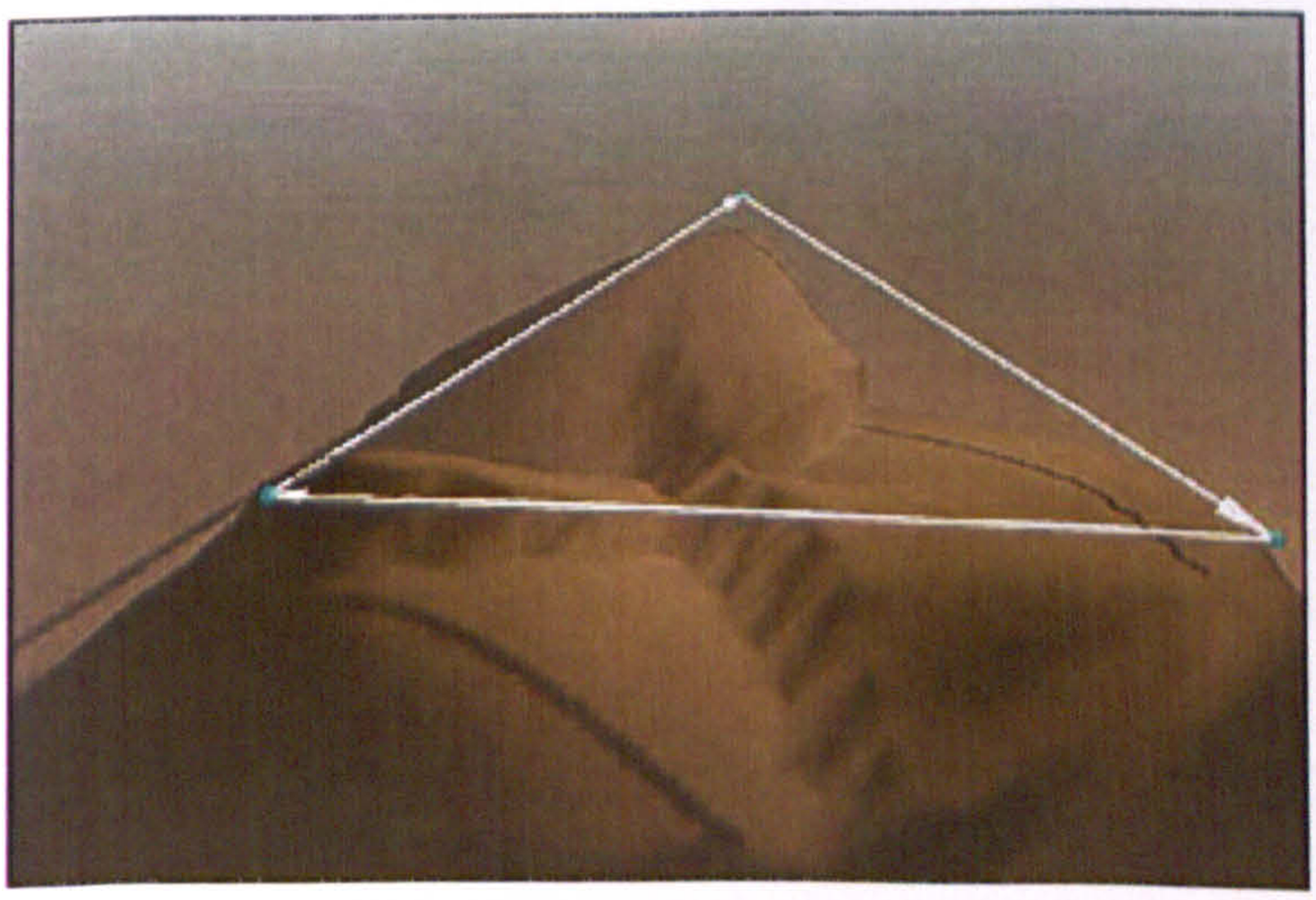


Figure 9.19 Plexus of VOOs with terrain (see video tape, time code: 10:07:24:00).

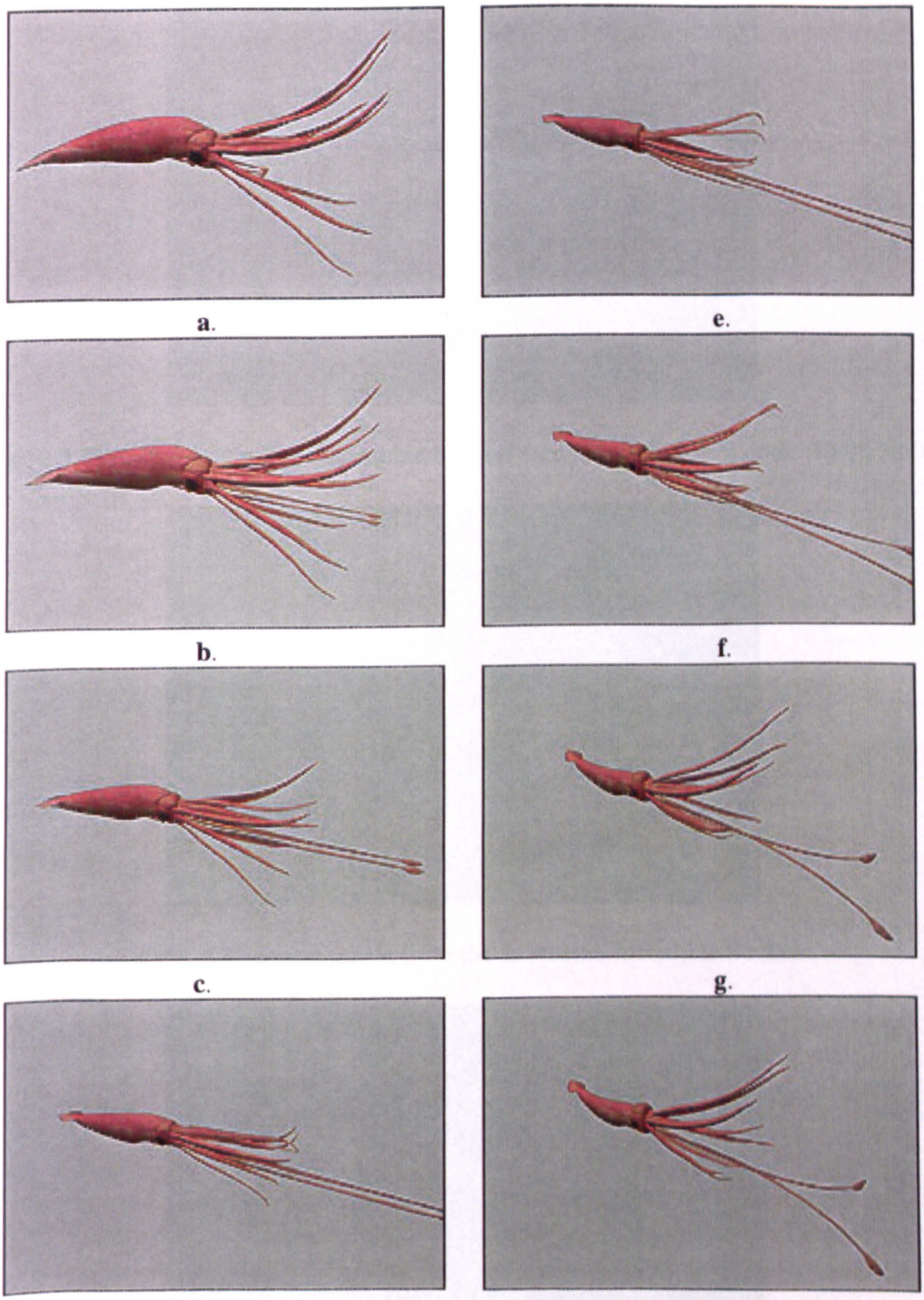


Figure 9.20 Giant squid animated using viscoelastic VOODOO (see video tape, time code: 10:08:24:00).

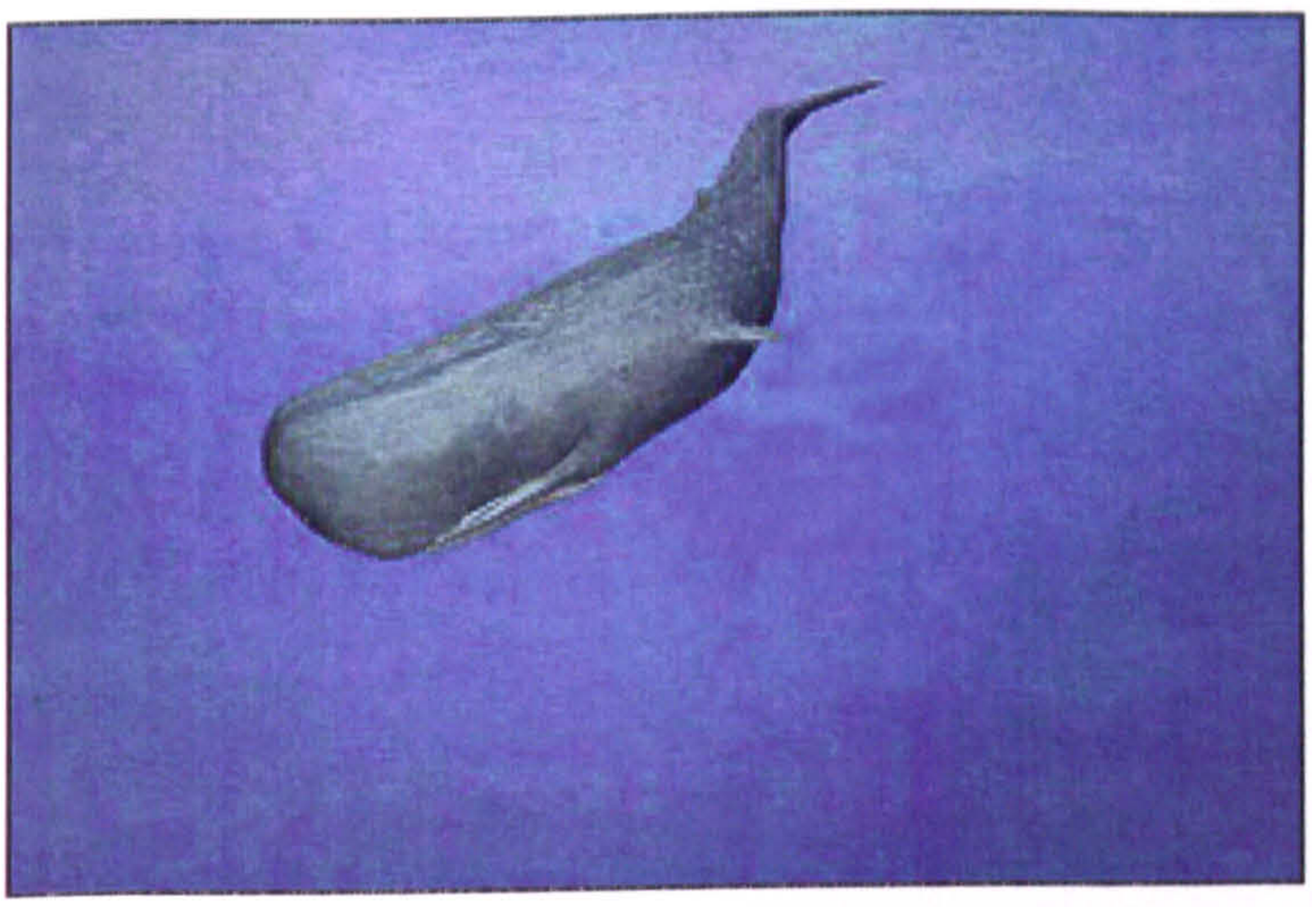


Figure 9.21 Sperm whale from Incredible Suckers (see video tape, time code: 10:08:38:00).

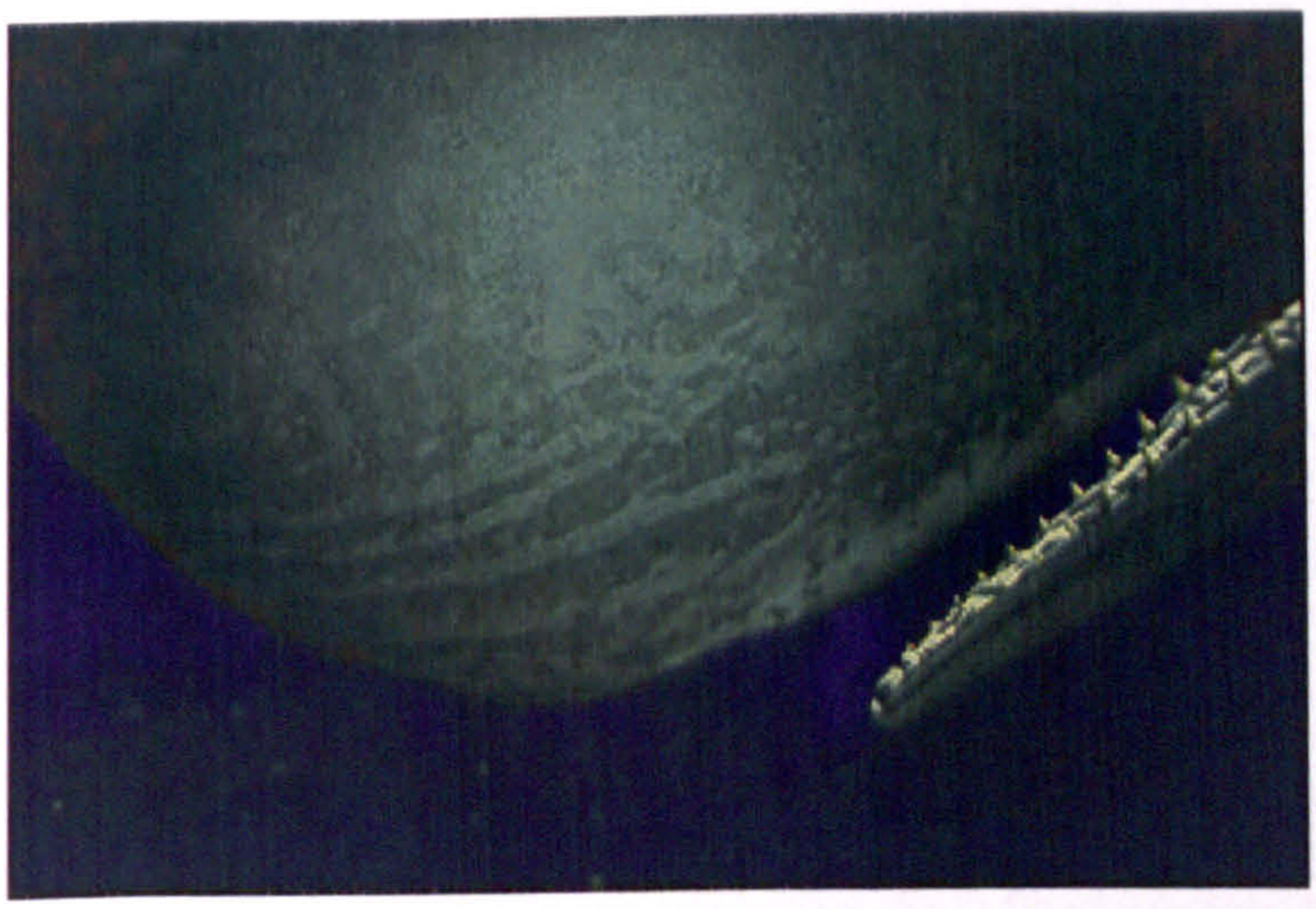


Figure 9.22 Sperm whale from the animation Incredible Suckers.

Figure 9.23 Sperm whale from the animation Incredible Suckers.

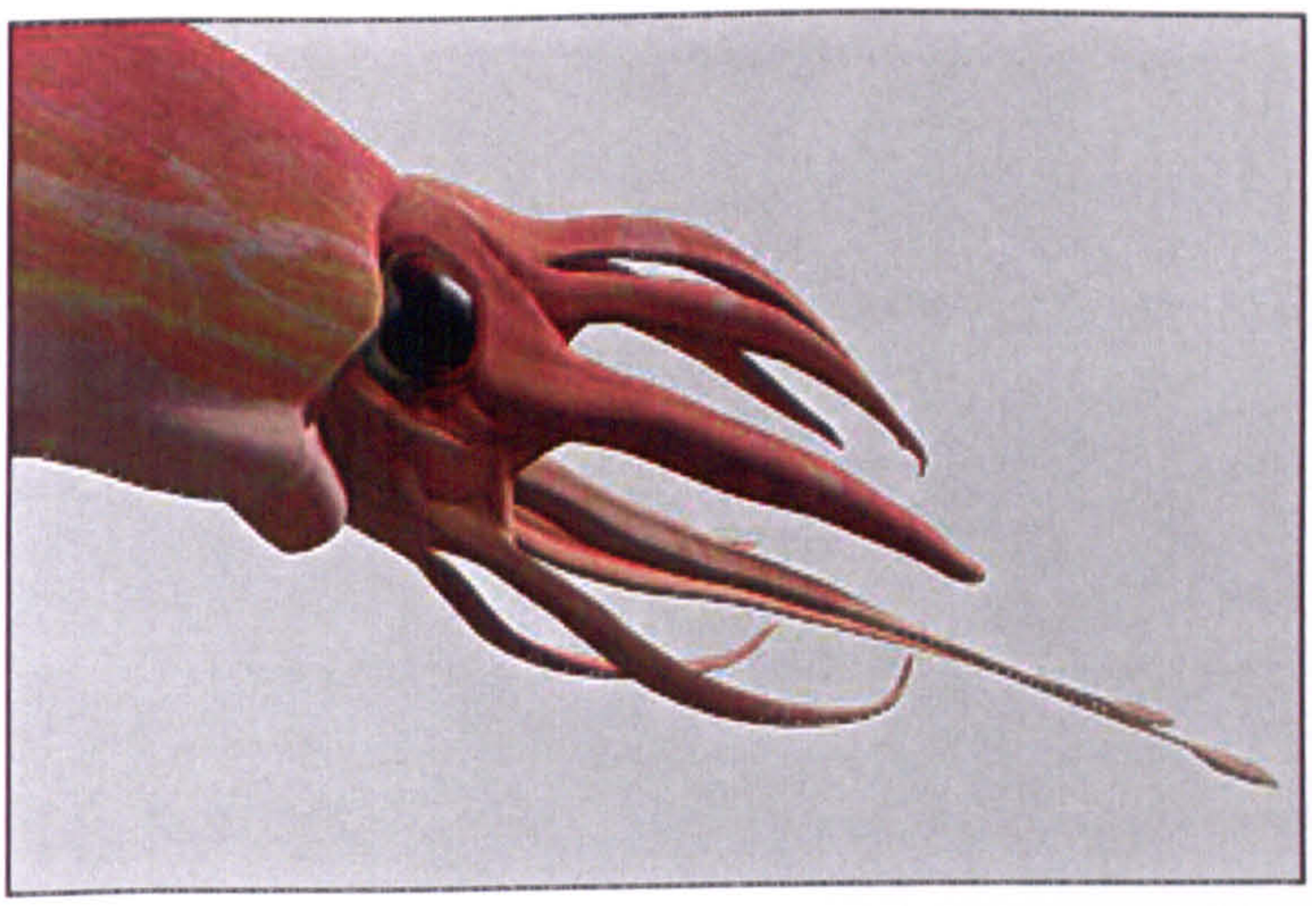


Figure 9.24 Giant squid model animated using viscoelastic VOODOO.

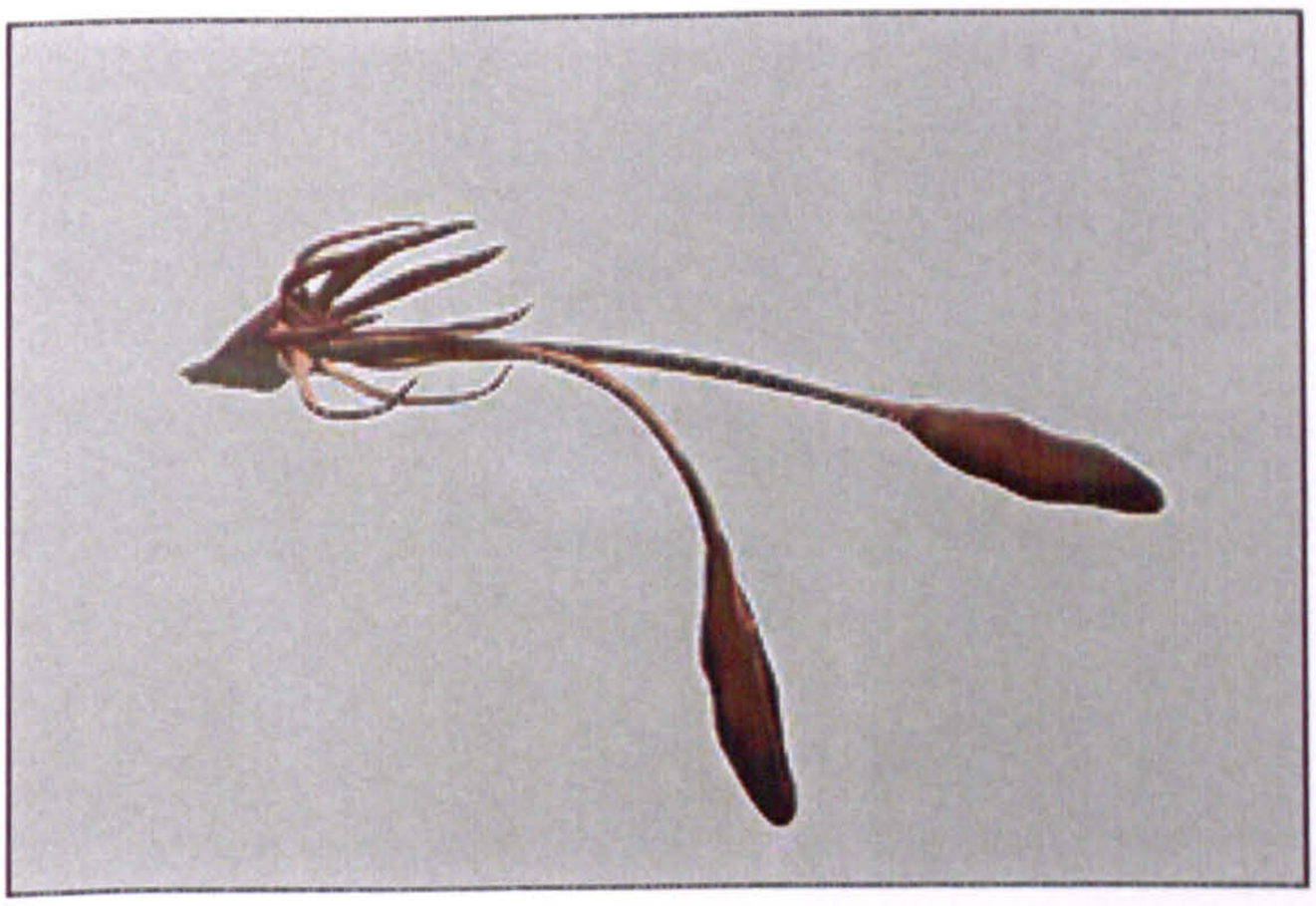


Figure 9.25 Giant squid model animated using viscoelastic VOODOO.

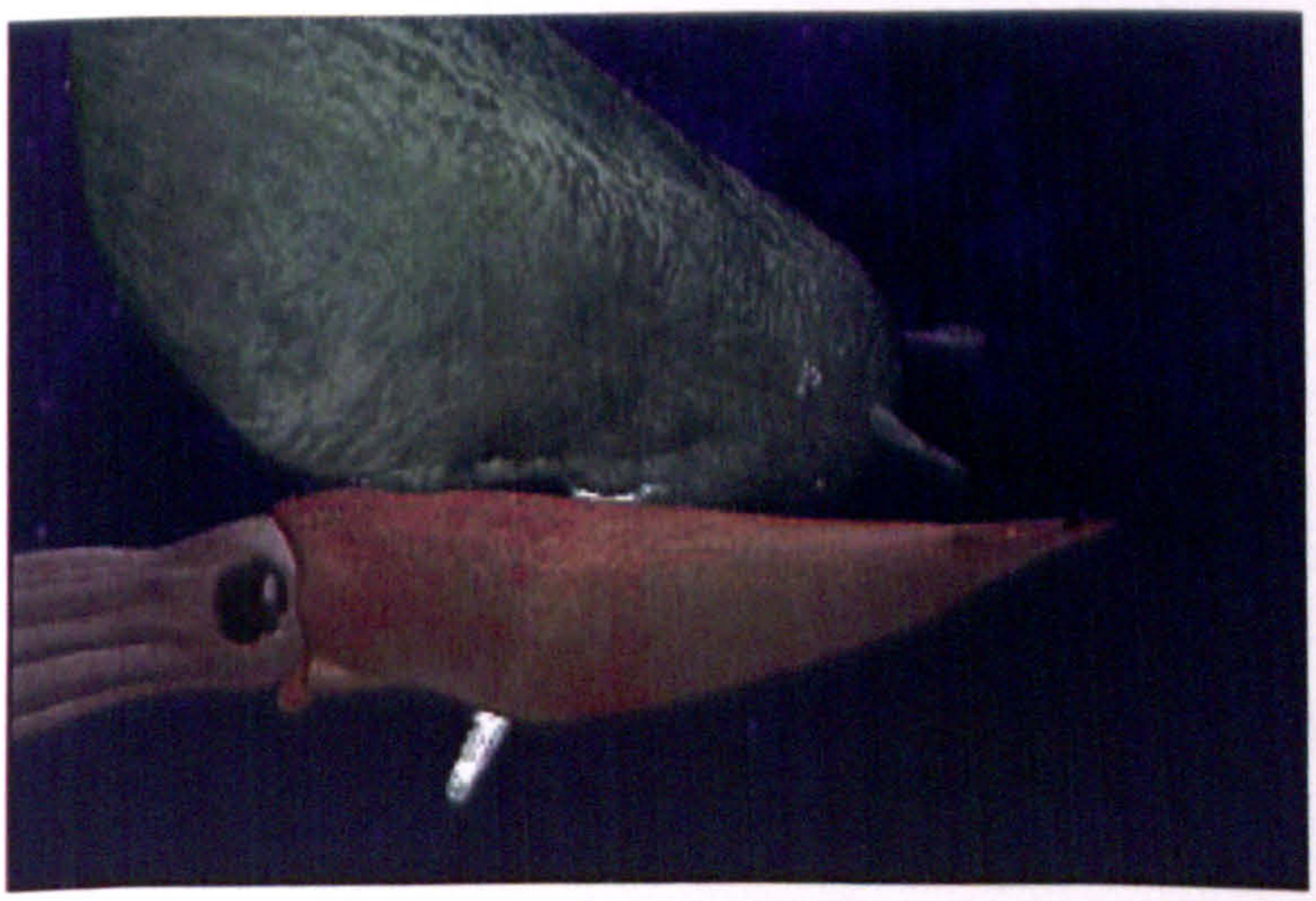


Figure 9.26 Whale and squid from the animation Incredible Suckers.

Figure 9.27 Stork from the animation after birth (see video tape, time code: 10:10:24:00).

Figure 9.28: Stork from the animation after birth.

Figure 9.29 Stork from the animation after birth.

Figure 9.30 Blue whale from the animation after birth.

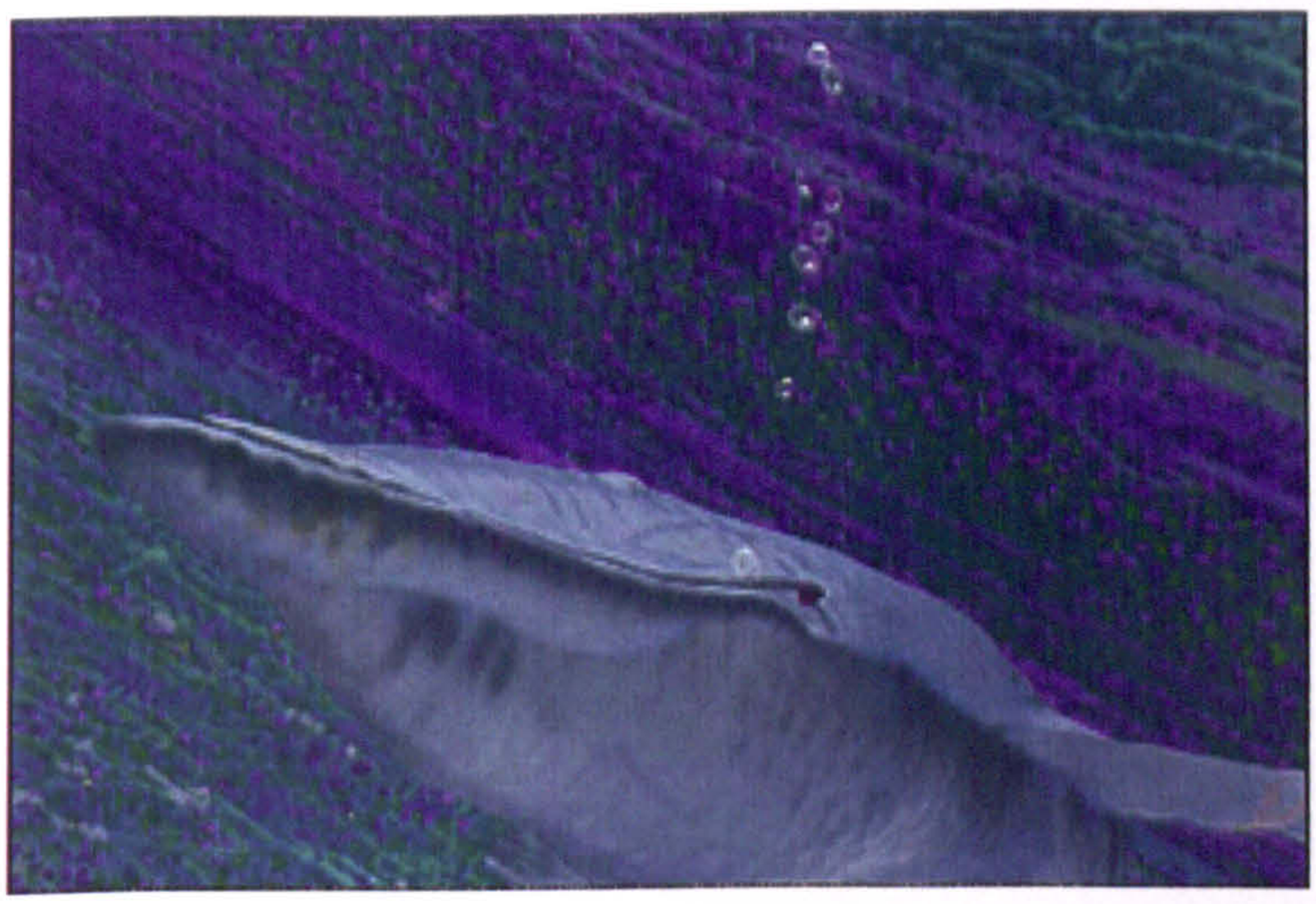


Figure 9.31 Blue whale from the animation after birth.

Figure 9.32 Blue whale from the animation after birth.

10

Epilogue

10.1 Summary of the Thesis

The first chapter of this thesis presented a general introduction to computer graphics followed by a presentation of the focus and the objectives of this research. In the second chapter, a survey of existing literature on deformable modelling techniques was attempted. In the third chapter, the background of the theory of elasticity, plasticity and viscoelasticity and the principles of organic material behaviour were introduced. In the fourth chapter, certain 3D object representation techniques were reviewed.

The fifth chapter presented the theoretic base of local deformation. Then, a new local deformation tool was introduced. A vector offset operator was introduced as a strain application tool suitable for interactive and time based manipulation of deformable objects. Interactive manipulation of VOOs offered an intuitive medium for applying complex deformations and for fine tuning the spatial behaviour of the deforming objects. Spatial damping of strain was introduced by allowing the user to define damping graphs which are complemented with strain control coefficients of flexibility, locality and weight. The volume of a deformed object was controlled by manipulating the elongation, volume preservation and cross-sectional coefficients. Fields of influence were introduced in order to localise the deforming effect of VOOs and to increase the efficiency of the model by predetermining the set of active vertices of an object. Finally, precalculation of the damping elements of an object was used to improve the efficiency of the deformation model.

In the sixth chapter, it was shown that a VOO may be engaged in a time based elastic restoration. Elastic restoration was achieved by damping the energy of the oscillation using an energy damping function over a user defined duration of time. The shape of the stress / strain graph along with the elastic and fracture strain limits were introduced as properties of a VOO and its surrounding local space. All three could be directly controlled by the user. These elements when combined with user defined mass, energy damping function and duration of elastic restoration enabled the creation of custom-made elastoplastic material behaviour. Furthermore, a stress / strain graph was based on real experimental data and thus lead to an approximation of realistic behaviour of specific materials.

The seventh chapter extended the deformation model to incorporate the property of viscoelasticity. The introduced viscoelastic model provided a mechanism for the implementation of the three characteristic viscoelastic features of strain creep, stress relaxation and stress / strain hysteresis. A spatial gradient of strain rate and a viscosity coefficient were used to generate a viscous effect in a deformable object. A strain creep graph, a strain creep coefficient and a strain creep duration provided a temporal constraint for the development of viscoelastic creeping strain inside the volume of a 3D object. A stress relaxation graph was used to diminish internal elastic stresses over a period of time. And finally an hysteresis graph was used to describe the hysteresis loop during the loading and unloading process. The user was given the ability to

customise the viscoelastic behaviour of an object by manipulating these graphs and coefficients. It was shown that an object could be made to respond like a thin fluid or a thick fluid or as an elastoplastic material with no viscosity.

The eighth chapter introduced several ways of constraining the deforming effect of VOOs in space and time. The focus was concentrated on fast, effective and intuitive ways of dynamic constraint specification that are well suited to the animation practice. VOOs were equipped with space and time constraints. Directly specified constraints were spatial constraints which restrict the position, length, direction or the aiming point of a VOO. Indirectly specified constraints were arthra and motion paths. Arthra were used to specify connection points between several VOOs. Motion paths helped restrict the dynamic motion of a VOO in space and time. Skeletons and plexi of VOOs were constructed by connecting several VOOs with arthra. Finally, vector fields were used in order to define the state of VOOs algorithmically.

The ninth chapter presented the structure of the test bed application VOODOO. VOODOO consists of the following set of structured models: the plastic deformation model, the elastoplastic model, the viscoelastic model and the constraint model. The functionality of the interactive framework of VOODOO was introduced. Interactive VOODOO was used for visual experimentation of deformation, for testing the performance of various features of the deformation model and for 3D modelling. The animation framework of VOODOO was also introduced. A sample script illustrated a method of linking VOODOO to the animation software CGAL. CGAL scripts were used to animate VOOs which in turn deformed objects. The VOODOO animation framework was used for the production of several animated sequences which were reviewed in this chapter. Finally, certain points that stretch the effectiveness and efficiency of the VOODOO model were examined.

10.2 The Objectives of the Thesis

The objectives set at the beginning of this thesis were:

- to develop a general and efficient deformation model,
- to equip the deformation model with organic material properties,
- to enable the application of spatial and temporal constraints on organic objects,
- to enable interaction with organically deformable objects and
- to enable the animation of organically deformable objects.

The concept of vector offset operators was introduced and was implemented as a strain application tool capable of specifying efficient and general deformation (see chapter 5).

The deformation model was equipped with organic material properties of elasticity, plasticity and viscoelasticity (see chapter 6 and chapter 7). The elastoplastic model was based on a stress / strain graph and an energy damping graph and the elastic restoration process was applied directly onto VOOs. The viscoelastic model was based on strain creep, stress relaxation and hysteresis graphs.

Constraining organic deformation was achieved by applying constraints directly onto VOOs (see chapter 8). Simple spatial constraints were used to confine VOOs in 3D space. Dynamic motion paths equipped with momentum / time graphs were used to apply dynamic motion to VOOs and hence to deformable objects.

An interactive framework was developed in order to facilitate visual experimentation of organic deformation, for testing the performance of various features of the model and to be used as a 3D modelling tool (see chapter 9). Using interactive VOODOO a user can interact with 3D organic objects and observe their visual behaviour.

An animation framework was developed for testing the time related parameters of the model and for animation production (see chapter 9). Using the VOODOO animation framework, a user can

construct and animate deformable organic characters and natural phenomena. Several animations were produced using the animation framework of VOODOO and have been included in the accompanying video tape.

10.3 Conclusion

VOODOO creates an intermediate layer between user interaction, the elastoplastic model, the viscoelastic model, the constraint model and the deformation model. This results to an efficient approach to deformation, facilitates the application of temporal and spatial constraints and enables further development of the deformation model.

The VOODOO generic tool disassociates deformation from object representation. Although the suggested model has been presented in a polygonal context, it may easily be adapted to deform volumetric space or parametric surface space. Furthetmore, VOODOO is independent from the underlying local deformation model. For example, VOOs may be adapted to operate upon free form deformation lattices.

The combination of an elastoplastic and a viscoelastic model with a deformation model leads one step closer to the automation of organic material behaviour in computer animated objects. As a result, much the animator's time and effort may be released towards other animation problems. The animator is freed from the burden of having to specify key-frames for every minute movement in an organically deforming object. By defining or modifying existing stress / strain, strain creep, stress relaxation and hysteresis graphs the user has a fine and high level control over the shape and the temporal behaviour of the deformation. Therefore, VOODOO offers extensive control of complex organic material behaviour and it may be used to automate the animation process of organically deformable objects.

Compared to conventional interaction and animation techniques for geometric deformation, VOODOO offers a greater degree of freedom and a more intuitive interface.

Compared to physically based simulation models, VOODOO offers the advantage of efficiency. The model introduced results in a fast implementation which is capable of achieving satisfactory interactive update rates.

The VOODOO model is the result of a hybrid geometric and physically based approach and it is general, reusable, extendible and independent of geometric representation.

The original concepts introduced in this thesis are:

- vector offset operators used as a strain application generic tool,
- precalculation of strain damping elements,
- use of user defined stress / strain graphs for the specification of elasticity coefficient and permanent plastic strain,
- incorporation of energy damping graphs for constraining elastic restoration,
- combination of viscoelastic features based on user defined graphs with a local deformation model and
- the combination of motion paths with momentum / time graphs to specify dynamic constraints.

10.4 Future Work

A redevelopment of the test bed application VOODOO will be attempted. The objective will be to provide a complete environment for the creation of physically based animation of organic characters based on living creatures.

The deformation model will be extended to incorporate complete body dynamics with linear and angular momentum. Collision and self-penetration avoidance will be built in the model to allow more natural looking deformable behaviour. The Taylor method for numerical integration will be implemented in solving the dynamic motion path problem and the elastic restoration problem. This will provide better numerical stability and will improve the efficiency of the

model. The functionality of the VOODOO generic tool will be extended to allow deformation on volumetric objects, on parametric surface control points and on free form deformation lattice control points. The application will incorporate a user friendly interactive and animation interface.

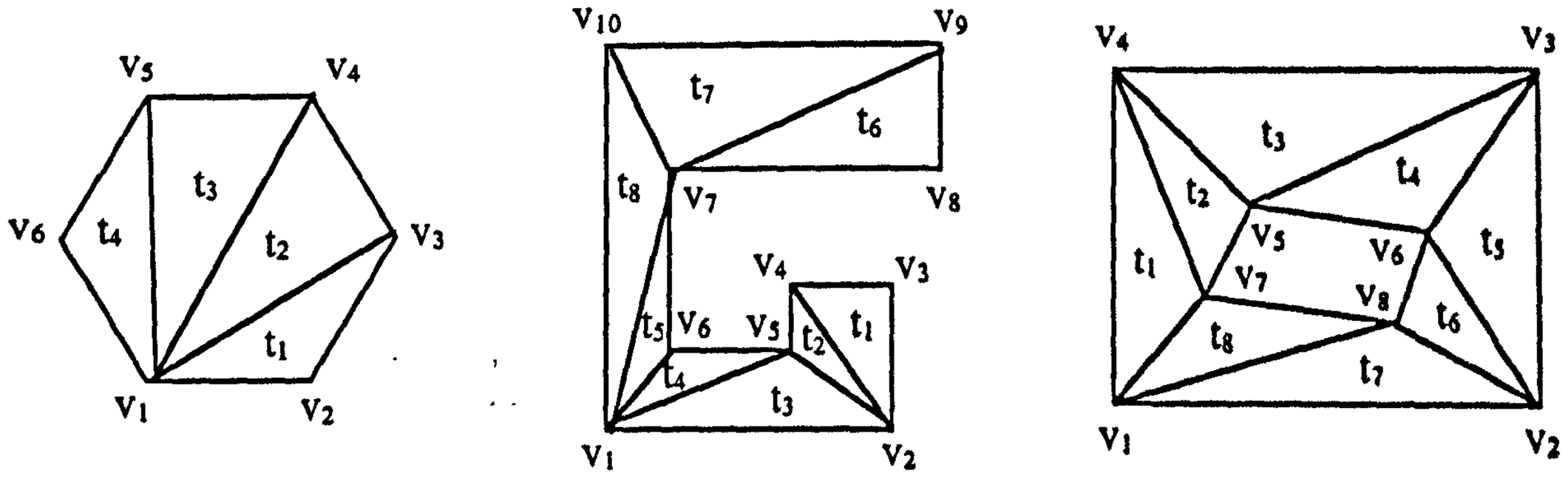
To facilitate learning of the application VOODOO, customised organic material behaviours will be provided through libraries. Stress / strain and strain creep graphs of common organic materials and motion paths based on common organic movements will be organised into such libraries to improve ease of use and speed of animation development.

PAGE NUMBERING AS ORIGINAL

Appendix A

A.1 A Triangulation Algorithm

Polygonal triangulation is the process that converts the face list of a polygonal object into a list of triangles. The process creates new edges and adds them to the edge list of the object. The vertex list, though, ought to remain unaffected. A simple triangulation algorithm was developed by the candidate in order to enable the use of objects derived from various polygonal data sources in the presented deformation methodology. This algorithm treats each polygon of the topology separately. It recursively converts a polygon of V vertices into $V + 2H - 2$ triangles (where H is the number of holes in the polygon). The algorithm is capable of triangulating convex or concave planar polygons and polygons with holes (see figure A.1).



a. convex polygon b. concave polygon c. polygon with a hole

Figure A.1 Triangulated polygons.

The algorithm is described in the following pseudocode:

TRIANGULATE(*polygon*)

Triangle List = empty

V = number of vertices in the polygon

a. IF $V > 3$ THEN

b. i = 1

c. Consider the triangle formed by the three consecutive vertices:

$\Delta(v_i, v_{i+1}, v_{i+2})$

d. IF the three vertices of the triangle are collinear THEN

i = i + 1; if i>V then i = 1

GOTO step **c.**

ELSE

e. IF the triangle is defined in clockwise order THEN (see figure A.2a)

i = i + 1; if i>V then i = 1

GOTO step **c.**

ELSE

f. FOR every vertex v of the polygon (excluding v_i, v_{i+1} and v_{i+2})

IF v lies inside $\Delta(v_i, v_{i+1}, v_{i+2})$ THEN (see figure A.2b)

i = i + 1; if i>V then i = 1

GOTO step **c.**

END of loop.

g. Create a new face $\Delta(v_i, v_{i+1}, v_{i+2})$ (see figure A.2c)

Add face to Triangle List

Remove vertex v_{i+1} from polygon,

$V = V - 1$

Add a new edge, $e(v_i, v_{i+2})$, to the polygon

Feed polygon back into step **a.** (see figure A.2d)

ELSE

h. Add last remaining triangle to Triangle List (see figure A.2h)

END.

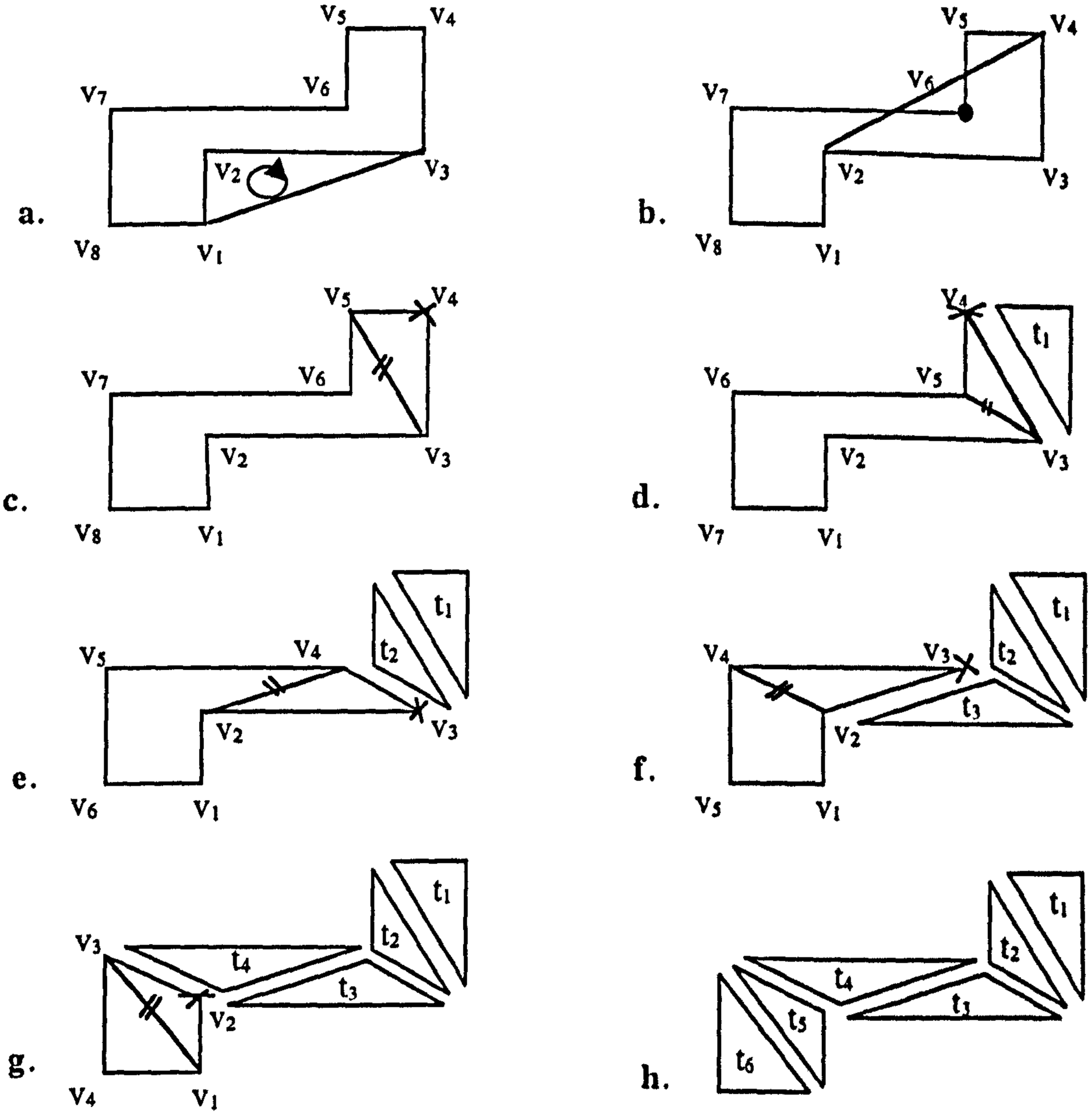


Figure A.2 Step by step triangulation of a simple polygon.

Conventionally, polygon outline contours are defined in a counter-clockwise order and hole contours, if any, in a clockwise order. A function that calculates the area of a triangle is employed to determine whether the contour of a triangle is in clockwise or counter-clockwise order. The area of a 2D triangle is given by the following formula:

$$Area = \frac{1}{2}\sum_{i=1}^{3}(y_i + y_j)(x_i - x_j) \qquad [A.1]$$

where i = 1 .. 3, j = i + 1 (if j > 3 then j = 1) and (x_i, y_i) are the coordinates of the triangle vertices. This formula returns a positive area for counter-clockwise defined triangles and a

negative area for clockwise defined triangles. 3D triangles are first projected on the XY plane. If a 3D triangle lies on a plane perpendicular to the XY plane, the triangle is projected on the XZ plane, etc. The formula returns zero, if the three vertices of the triangle are collinear.

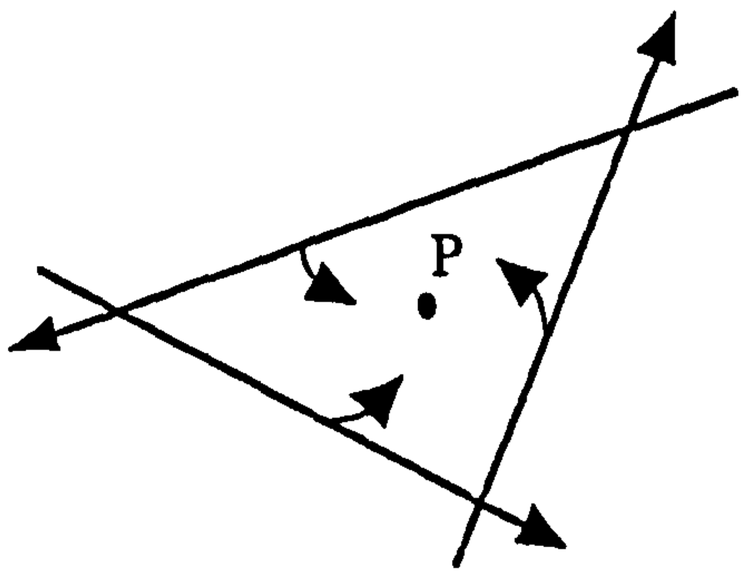


Figure A.3 Point P lies to the left side of each edge of the triangle.

A simple point-in-polygon algorithm is used to determine whether a vertex lies inside a triangle. Since a triangle is a convex polygon, this algorithm, simply, checks if the vertex in question lies on the left half-plane defined by each of the three edges (see figure A.3). If it does then the vertex lies inside the triangle.

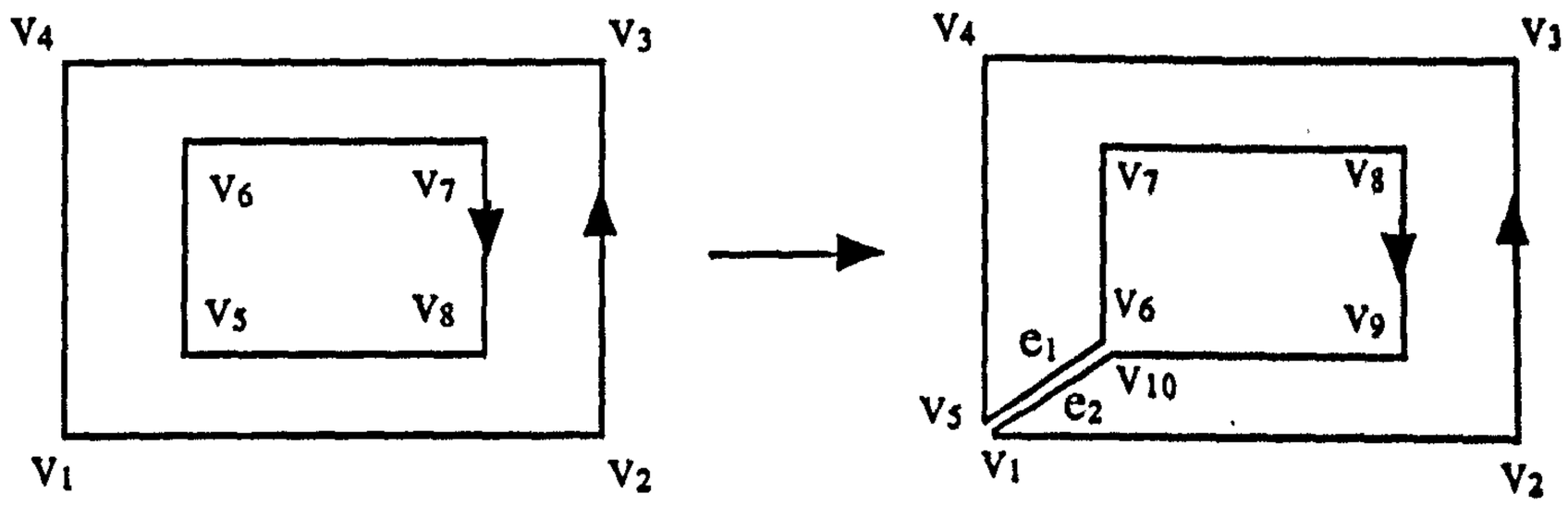


Figure A.4 Dummy vertices and dummy edges are added to the polygon.

Finally, to treat polygons with holes, the algorithm adds dummy vertices and edges in order to "bridge" the outline contour with each of the holes (see figure A.4). The resulting polygon is concave with no holes and can be triangulated by the algorithm.

Appendix B

The following paper was published in EUROGRAPHICS UK Chapter Proceedings, at Imperial College, London (Hurmusiadis 1996).

B.1 Vector Offset Operators for Dynamic Control of Deformable Objects

Abstract

The author addresses the issue of dynamic control of deformable objects in 3D computer animation. An intuitive and efficient approach to the control of deformable behaviour is introduced through the use of *Vector Offset Operators* and their related *Influence Fields*. The direct metaphor for the functionality of a vector operator is that of a rigid metal pin immersed into the mass of a solid material; or that of a pin attached to a flexible membrane by its tips. Vector operators are used for the specification and control of global and local deformations of polygonal topologies. When a vector operator is exposed to rigid-body Lagrangian dynamics, the flexible topology deforms, following rules based on simplifications of the theory of classical mechanics. The animator may control three basic types of *Permissible Strain*: elongation, bending and torsion. Flexible behaviour is achieved by the enforcement of a *Decay-of-Strain* function. External forces are applied via *Motion Paths* and *Momentum/Time* functions which convey the motion of vector operators. Emphasis is put on efficiency and visual correctness of flexible behaviour rather than on realistic simulation. Key characteristics of the proposed structured model is generality, extendibility and controllability.

Keywords and Phrases: 3D animation, deformation, motion path, dynamic control, deformable behaviour.

1 Introduction

Looking at the theoretic principles and professional practice of computer animation, it is becoming increasingly evident that it possesses a tremendous potential which has not yet been fully exploited. Recently, there has been an accelerated interest in the incorporation of laws of physics in animation techniques for 3D flexible bodies.

Computer simulation of physical laws can offer the much desired element of realism in animation, which is, otherwise, so painstakingly achieved with traditional key-framing methods. These later conventional methods are often based on simple techniques such as interpolation and massively rely on animator skill for the reproduction of controlled stylised realism. On the other hand, the approximate reproduction of natural flexible behaviour may be automated via computer implementations of mathematical models.

However, there are several serious tradeoffs that make physical simulation not a very common feature in computer animation applications. Such tradeoffs, among others, are high complexity of physically-based mathematical modelling, huge computational cost and a great difficulty of control.

As for complexity and computational cost, these are two elements that computer graphics systems developers, in general, have, somehow, learned to deal with. On the contrary, lack of direct control is something that animators will, perhaps, never be able to live with.

Despite these problems, the increasing demand for realism in animation has lead to the development of special purpose built, not extendible and awkward to use physically-based animation systems.

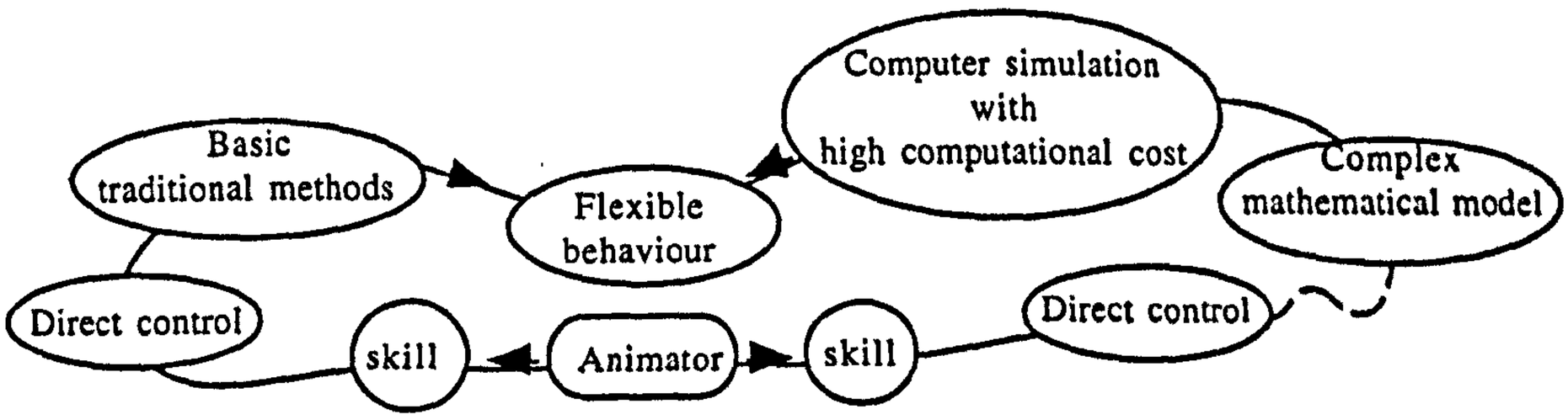


Figure 1: Computer simulation versus direct control of traditional methods in animation.

The goal of the presented research is the development of an efficient and easy to control model for the direct and dynamic manipulation of flexible bodies. The implementation of such a model ought to be structured and modularised, so that it is general, reusable, extendible and independent of representation.

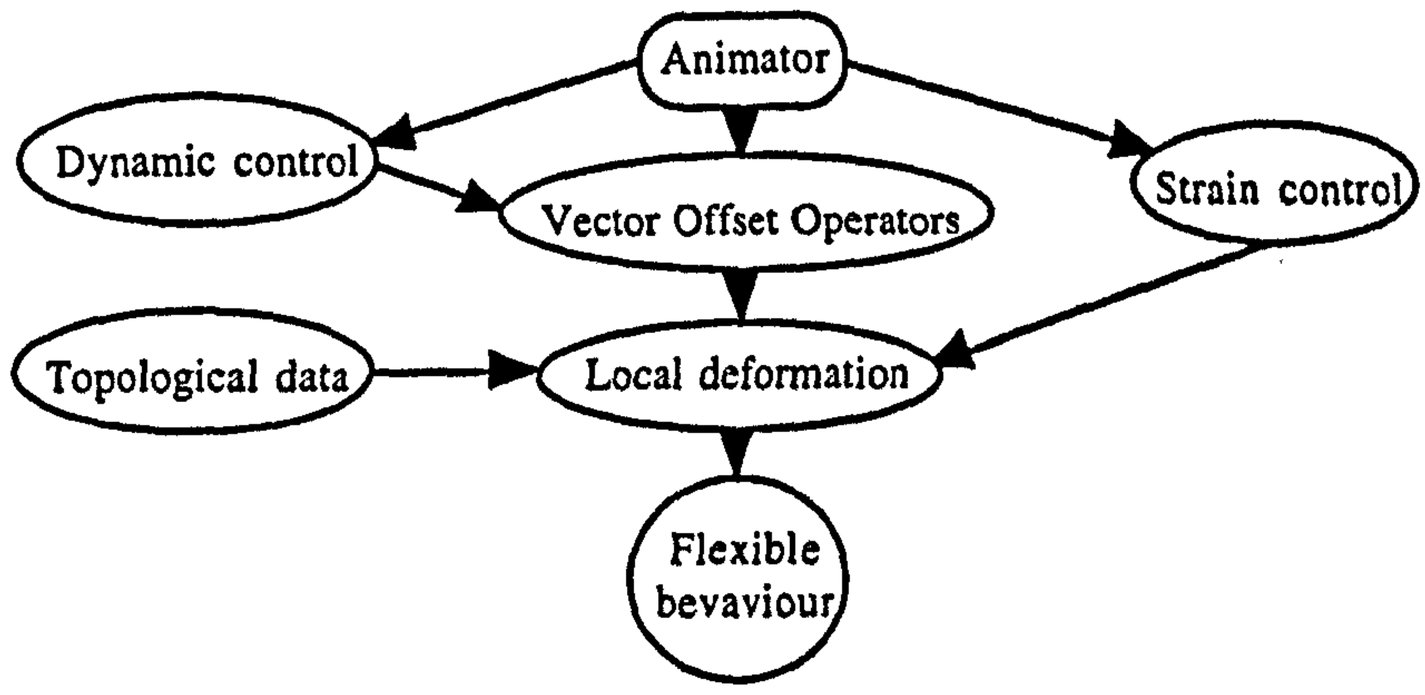


Figure 2: The proposed model maximizes animator control.

The introduced *Vector Offset Operator* (referred to as *VOO* from now on) is a deformation tool that offers intuitive and fine direct and dynamic control over flexible topologies. The motion of a VOO generates a local deformation that affects anything that lies inside its *Field of Influence*. The shape and the magnitude of the deformation is determined by the three *Permissible Strain Factors* and the *Decay-of-Strain Function*. The dynamic control of the motion of a VOO is achieved by the specification of a *Momentum/Time Function* and a *Motion Path* . Evidently, the focus of the proposed system is put on the needs of an animator rather than on an extensive physical simulation (see figure 2).

2 Prior Work

During the last decade there has been a great deal of research activity on flexible behaviour and the control of flexible bodies in computer animation. This has resulted to a large body of previous published research on this subject.

As early as in 1984, Barr suggests a simple yet very effective way to treat solids as "topological clay" which may be bent, twisted, tapered, compressed or expanded (Barr 1984). The normal or tangent vector of an arbitrarily deformed surface can be calculated from the surface normal or tangent of the undeformed surface and an affine transformation matrix. Thus, by applying affine transformations with a gradient of transformation amounts, over the whole or part of a solid, the shape of the solid is deformed smoothly. The beauty of Barr's approach lies in its simplicity, efficiency and general applicability. Many researchers, thereafter, will refer to "Barr's transformations" as a standard upon which they will build more complex deformation models.

A very different approach to flexible bodies was, later, adopted by Terzopoulos *et al.* (1987). They employed elasticity theory from classical mechanics to construct differential equations that model the behaviour of non-rigid curves, surfaces and solids as a function of time. They suggested a new class of "active elastically deformable models" which can be made to behave like a metal spring, rubber, cloth, paper etc. The next year Terzopoulos and Fleischer (1988) proceeded with the simulation of "inelastically deformable models". They attempted the simulation of the common physical properties of plasticity, viscoelasticity and fracture. Such an approach to flexible behaviour offers a thorough framework which is more suitable to visualisation rather than animation. The computational overhead could become a prohibitive factor for the incorporation of this method in fast, interactive animation applications.

Later in 1988, Platt and Barr presented a framework for flexible body constraints (Platt and Barr 1988). They introduced two types of constraints. Reaction constraints, "RCs", allow fast computation of collisions of flexible models with polygonal models. RCs also, allow the models to be pushed and pulled under the control of the animator. Augmented Lagrangian constraints, "ALCs", create effects such as volume preserving squashing and stretching. By adding physical constraints to elastic models, they achieve a compromise between a complete specification of the motion in one hand and a pure numerical simulation solution on the other.
In 1988, Witkin and Kass (1988) proposed their "space time constraint" methodology. They adopted a formulation whose central characteristic is the numerical solution of the character's motion and time-varying muscle forces over the entire time interval of interest. Their method "optimises" certain functions, like for instance, energy consumption, that specify how the motion should be performed in terms of efficiency, smoothness etc. Solving the constrained optimisation problem yields optimal, physically valid motion that achieves the "goals" specified by the animator. They applied this method on rigid articulated bodies with uniform mass and frictionless joints.

Witkin and Welch (1990) described a fast method for physically based animation of flexible bodies. Their approach treats motion control of non-rigid bodies with attachment constraints. To achieve non-rigid body behaviour they apply global deformations (Barr 1984) on objects with mass. Such a formulation can produce objects that deform in a geometrically simple but physically correct way. Attachment constraints are achieved by calculating constraint forces that counter any external or internal forces that tend to pull objects apart. Motion control, finally, is achieved by "attaching" a point on an arbitrary trajectory. In other words this method can produce articulated puppets whose parts are made of jellylike material, with specified points under full control of the animator. By moving these control points as functions of time, the puppet moves with correct passive dynamics.

Hilton and Egbert (1994) have developed an interactive tool for physically based 3D particle

systems which is also extendible to soft objects. Their method offers flexibility in animation, modelling and simulation. The tool consists of bounded dynamic vector fields which apply forces, accelerations or velocities onto particles or object elements. This method can attack a wide range of problems in computer animation such as rigid and non-rigid body motion, natural phenomena, modelling, image processing, simulation of gases etc.
Recently, Lerios *et al.* considered 3D metamorphosis applied to volume-based representations of objects (Lerios *et al.* 1995). They adopted a similar approach to the one described in (Beier and Neely 1992). By applying feature-based 3D volume warping, their method allows fine animator control and achieves realistic looking intermediate objects.

Finally, Snibbe (1995) presented a new set of interface techniques for visualising and editing animation directly in a 3D scene. Snibbe suggested the separation of spatial and temporal control of position by using two curves for each animated object: the "motion path" which describes the 3D spatial path along which an object travels and the "motion graph", a function describing the distance travelled along this curve over time.

Several commercial animation systems have recently incorporated techniques for the creation and control of flexible models. The two most dominant applications are reviewed below.

Silicon Graphics' Alias/Wavefront v7 (Alias 1997) supports "Mold and Sculpt", an intuitive sculpting tool that allows the animator to interactively shape the model. It also supports "Blobby Objects", a feature that enables the creation of organic objects using implicit surfaces; "Deformation", a set of tools for bending, twisting, squashing and stretching; "Character Builder" for the animation of muscle and skin deformations.
Microsoft's Softimage/3D Extreme (Softimage 1997) supports "Meta-Clay Modelling", a density based modelling tool for organic, sculptured objects; "Deformation" via cluster, control point, effector, lattice, patch, spline or vertex manipulation; "Q-Stretch" for automatic squash and stretch deformation based on speed and acceleration; "Skin" for automatic local and global weighted envelopes; "Dynamics" for the specification of density, mass, elasticity, force etc.

3 The Deformation model

The suggested methodology provides with a tool for the manipulation of triangulated polygonal 3D topologies in space and time. The choice to use triangulated polygonal geometry representation was made due to its simplicity, efficiency and portability. Another advantage in using triangles is that collision detection and ray tracing can be optimised for them. However, the proposed model is not limited to polygons and it may, easily, be extended to treat surface patches, solids or volumetric data.

A wider concept of a model incorporates a spatial topology and a set of rules that convey the topological elements. In the following, a topological model will be provided with the ability to deform in space and time and with a set of rules that harness this ability.

3.1 Vector Offset Operators

A VOO is a deformation primitive which facilitates the modelling and animation of flexible objects. The physical counterpart of a VOO is that of a metal pin inside a real object. If the object is soft any movement of the pin will move and deform the object. If the object is rigid then the pin will simply force the object into rigid motion.

A VOO is in fact a fixed vector with magnitude, direction, and location and is independent from the topologies that surround it. The animator may directly specify a set of VOOs which may be located at any position or orientation in relation to the surrounding objects (see figure 3).

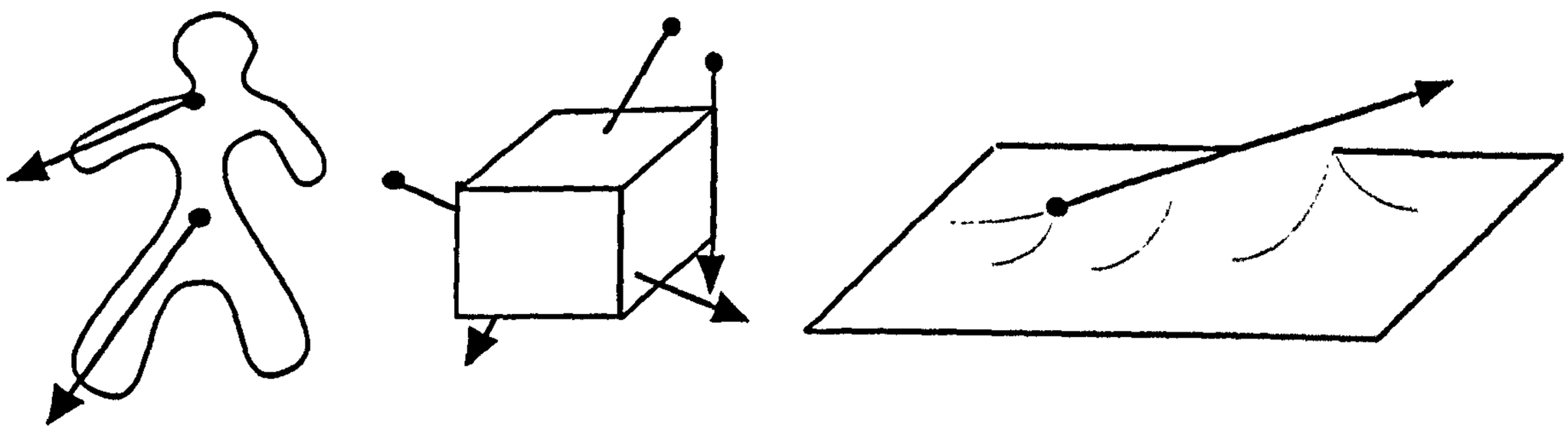


Figure 3: The VOO deformation primitive and some arbitrary topologies.

A direct offsetting of the tips of a VOO will result to a deformation on the topological elements around it. VOOs offer an intuitive way of specifying complex changes to a 3D shape and they serve as a direct strain application tool for flexible objects.

3.1.1 Global Deformation

Let us consider a VOO as a fixed vector **V** in 3D space. A local frame of reference, **L**, is attached to **V** in order to aid the definition of the global deformation. At first the local frame will be used to represent the topology at rest state (see figure 4).

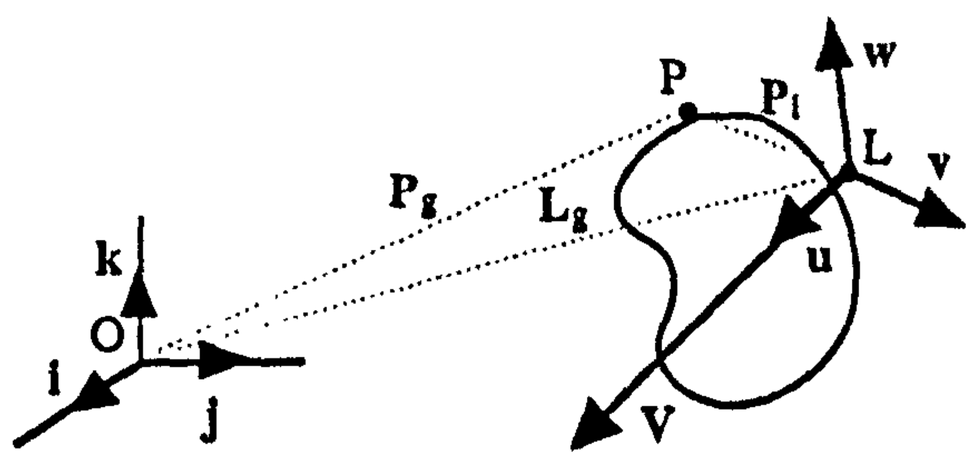


Figure 4: A VOO and local frame of reference at rest state.

The local frame of reference, L, is derived as follows:

i, j, k:	unit vectors of the global frame of reference.
u, v, w:	unit vectors of the local frame of reference.
L:	origin of the local frame (start tip of **V**).
$\mathbf{u} = \mathbf{V} / \lvert\mathbf{V}\rvert$	normalised **V**.
if **u** parallel to **k** then	$\mathbf{v} = \mathbf{j} \otimes \mathbf{u}$
else	$\mathbf{v} = \mathbf{k} \otimes \mathbf{u}$
$\mathbf{v} = \mathbf{v} / \lvert\mathbf{v}\rvert$	
$\mathbf{w} = \mathbf{u} \otimes \mathbf{v}$	

Each vertex of the topology is then mapped from global to local frame. For each vertex P (see figure 4), a vector P_l must be calculated (subscript $_l$ denotes local and $_g$ global):

$\mathbf{P_g} = \mathbf{OP}$	position vector of P in global frame.
$\mathbf{L_g} = \mathbf{OL}$	position vector of L in global frame.
$\mathbf{q_g} = \mathbf{P_g} - \mathbf{L_g}$	temporary vector.

$\mathbf{P_l} = [\ (\mathbf{q_g} \odot \mathbf{u})\ \ (\mathbf{q_g} \odot \mathbf{v})\ \ (\mathbf{q_g} \odot \mathbf{w})\ ]$ position vector of P in local frame. **[1]**

At a time *t* of an animation sequence a vector operator **V** may assume an arbitrary position, direction and magnitude given by **V**(*t*) (see figure 5). A VOO may undergo dynamic or direct manipulation by offsetting its tips in 3D space. This process will apply an arbitrary concatenation of global translation and local rotation and scaling to its attached local frame **L**(*t*).

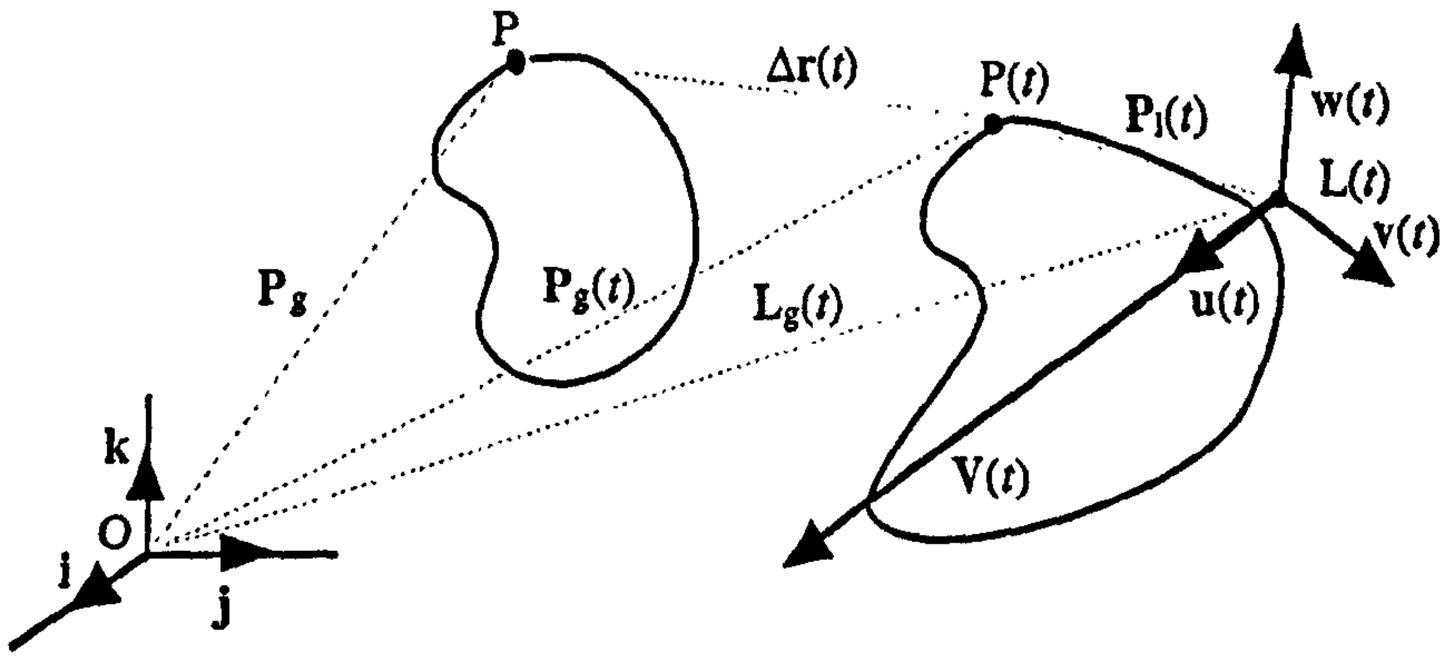


Figure 5: The deformed topology and a VOO at time *t*.

The local frame **L**(*t*) on **V**(*t*) is calculated as follows:

L(*t*): origin of local frame (start tip of **V**(*t*)).
$\mathbf{u}(t) = \mathbf{V}(t) / |\mathbf{V}(t)|$ normalised **V**(*t*).
if **u**(*t*) parallel to **k** then $\mathbf{v}(t) = \mathbf{v} \otimes \mathbf{u}(t)$
else $\mathbf{v}(t) = \mathbf{w} \otimes \mathbf{u}(t)$
$\mathbf{v}(t) = \mathbf{v}(t) / |\mathbf{v}(t)|$
$\mathbf{w}(t) = \mathbf{u}(t) \otimes \mathbf{v}(t)$

Vector $\mathbf{P_l}$ (see figure 4) undergoes the above mentioned composite transformation and becomes vector $\mathbf{P_l}(t)$ (see figure 5). At this stage only the local scaling and the local rotation about axis **u** will be considered. The global translation and local rotation about **v** and **w** will be discussed later.

$$\mathbf{P_l}(t) = \begin{vmatrix} s_1 & 0 & 0 \\ 0 & s_2 * \cos(\theta) & -s_2 * \sin(\theta) \\ 0 & s_2 * \sin(\theta) & s_2 * \cos(\theta) \end{vmatrix} * \mathbf{P_l} \qquad \textbf{[2]}$$

The scaling factors s_1 , s_2 and the rotation angle θ will be discussed below in §3.2.1. Vector $\mathbf{P_l}(t)$ is the position vector for P(*t*) in local frame **L**(*t*) and ought to be mapped back onto global frame. But first, the global unit vectors **i**, **j** and **k** are mapped onto local frame **L**(*t*):

$$\mathbf{I}(t) = [\ (\mathbf{i} \odot \mathbf{u}(t))\ (\mathbf{i} \odot \mathbf{v}(t))\ (\mathbf{i} \odot \mathbf{w}(t))]$$
$$\mathbf{J}(t) = [\ (\mathbf{j} \odot \mathbf{u}(t))\ (\mathbf{j} \odot \mathbf{v}(t))\ (\mathbf{j} \odot \mathbf{w}(t))]$$
$$\mathbf{K}(t) = [\ (\mathbf{k} \odot \mathbf{u}(t))\ (\mathbf{k} \odot \mathbf{v}(t))\ (\mathbf{k} \odot \mathbf{w}(t))]$$

Then, vector $\mathbf{P_l}(t)$ is mapped onto **I**(*t*), **J**(*t*) and **K**(*t*):

$$\mathbf{q_g}(t) = [(\mathbf{P_l}(t) \odot \mathbf{I}(t))\ \ (\mathbf{P_l}(t) \odot \mathbf{J}(t))\ \ (\mathbf{P_l}(t) \odot \mathbf{K}(t))]$$

Vector $\mathbf{q}_g(t)$ is in fact vector $\mathbf{P}_l(t)$ expressed in global components. From figure (5), the following may be derived:

$$\mathbf{P}_g(t) = \mathbf{L}_g(t) + \mathbf{q}_g(t) \quad \text{position vector of P}(t) \text{ in global frame.} \qquad [3]$$

The total displacement from P to P(t) is represented by the deformation vector at time t, $\Delta\mathbf{r}(t)$ and is calculated combining equations [1] and [3] (see figure 4 and 5).

$$\begin{aligned} \Delta\mathbf{r}(t) &= \mathbf{P}_g(t) - \mathbf{P}_g \\ &= (\mathbf{L}_g(t) - \mathbf{L}_g) + (\mathbf{q}_g(t) - \mathbf{q}_g) \\ &= \Delta\mathbf{l}(t) + \Delta\mathbf{q}(t) \end{aligned} \qquad [4]$$

The factor $\Delta\mathbf{l}(t)$, in the above equation [4], contains the global translation and the factor $\Delta\mathbf{q}(t)$ contains the combined local rotation and scaling. Separating these two factors in the deformation vector, actually separates the rigid motion in global space from the flexible deformation in local space.

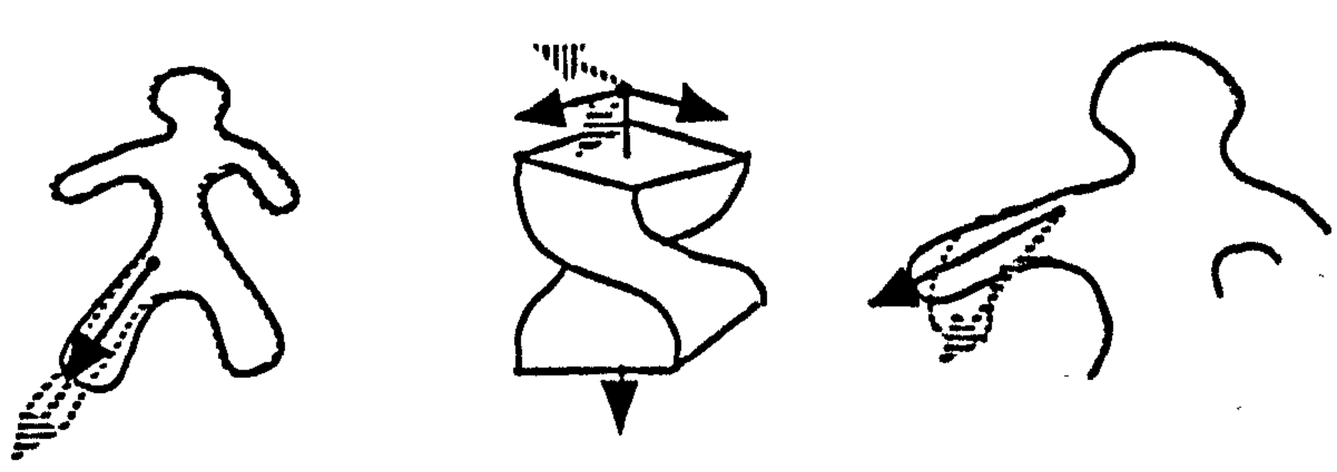
A similar approach to global deformation was introduced in (Barr 1984) and then later extended in (Witkin and Welch 1990). The introduction, though, hereby, of VOOs offers fine and efficient control and enables direct and dynamic manipulation of flexible topologies.

3.2 Strain Control

An ideal plastic material yields to any stress applied to it and it accepts any resulting strain as permanent deformation (Fung 1965). Under certain stress limitations, modelling clay behaves like an ideal plastic material. In the present formulation, the behaviour of a plastic material may well be simulated using the deforming properties of a VOO. Since there are no restoring forces to counteract the external ones, it seems logical to directly apply strain instead of stress onto the model. This strain-level approach to plastic modelling has the following physical metaphor: When an animator manipulates a piece of real clay in his/her hands does not really think of the applied stress. The amount of applied stress is actually, subconsciously, controlled through the haptic interaction with the mass of clay. What the animator is, really thinking of, during the modelling session, is the visual control of strain and the relative proportions of the model as it changes shape.

3.2.1 Permissible Strains

There are three distinct intuitive ways to deform a flexible object, which are also suggested by classical mechanics (Fung 1965). The three basic strain types are (see figure 6):



a. Elongation (stretch). **b.** Torsion (twist). **c.** bending (bend).

Figure 6: The three basic strain types from classical mechanics.

Elongation

With reference to a VOO and its local frame, the elongation strain is defined as a deformation along the direction **u** (see figure 6.a). An arbitrary offsetting of the tips of a VOO that may result to a change in magnitude for the VOO, will cause a scaling of the surrounding topology along the axis **u**. The actual elongation amount, E, and the elongation fraction, e, are given by:

$$E = |\mathbf{V}(t)| - |\mathbf{V}| \text{ and } e = E / |\mathbf{V}|$$

The animator may directly specify or modify a *permissible elongation strain factor*, ε. Through this permissible elongation factor, the animator can describe the level of elongation permitted on a deforming object. The scaling factor along the axis **u** may then be defined as follows:

$$s_1 = 1 + \varepsilon * e$$

where $0 \leq \varepsilon \leq 1$

when $\varepsilon = 0$ then $s_1 = 1$ — no elongation is permitted.

when $\varepsilon = 1$ then $s_1 = |\mathbf{V}(t)| / |\mathbf{V}|$ — any level of elongation is permitted.

Volume preservation during elongation may be achieved by applying a uniform scaling along the axes **v** and **w**. The animator may control the extend of volume preservation by specifying the *permissible volume preservation factor*, α. The scaling factor along the axes **v** and **w** may then be defined as follows:

$$s_2 = \alpha / s_1$$

where $\alpha > 0$

when $\alpha = 1$ a uniform volume preservation will follow an elongation.

when $\alpha < 1$ an exaggerated compression will follow an elongation.

when $\alpha > 1$ a reduced compression will follow an elongation.

s_1 and s_2 are the scaling factors that appear inside the matrix in equation [2].

Torsion

The torsion strain is caused by a rotation of a VOO about the local axis **u**, by a directly specified angle, Θ (see figure 6.b). The animator may control the torsion by specifying the *permissible torsion strain factor*, ϕ. The angle of rotation is then calculated as follows:

$$\theta = \phi * \Theta$$

where $0 \leq \phi \leq 1$

when $\phi = 0$ no torsion is permitted.

when $\phi = 1$ any torsion is permitted.

Bending

When a stress is applied onto a small area of a flexible object, the effect is transferred onto the neighbouring molecules, thus spreading on the entire volume of the object. However, in flexible materials, the connecting forces among neighbouring molecules are weak and allow for local deformations which store deformation energy. The result of this is a spatial dumping of stress and of its effects on the material. This dumping is somehow proportional to the distance from the stress application centre and affects the magnitude and the direction of the applied stress. The bending strain is caused exactly by this dumping of the direction of stress. Therefore, in order to implement the bending element of strain, a spatial dumping must be

applied on the total local strain $\Delta q(t)$, from equation [4] (see figure 7).

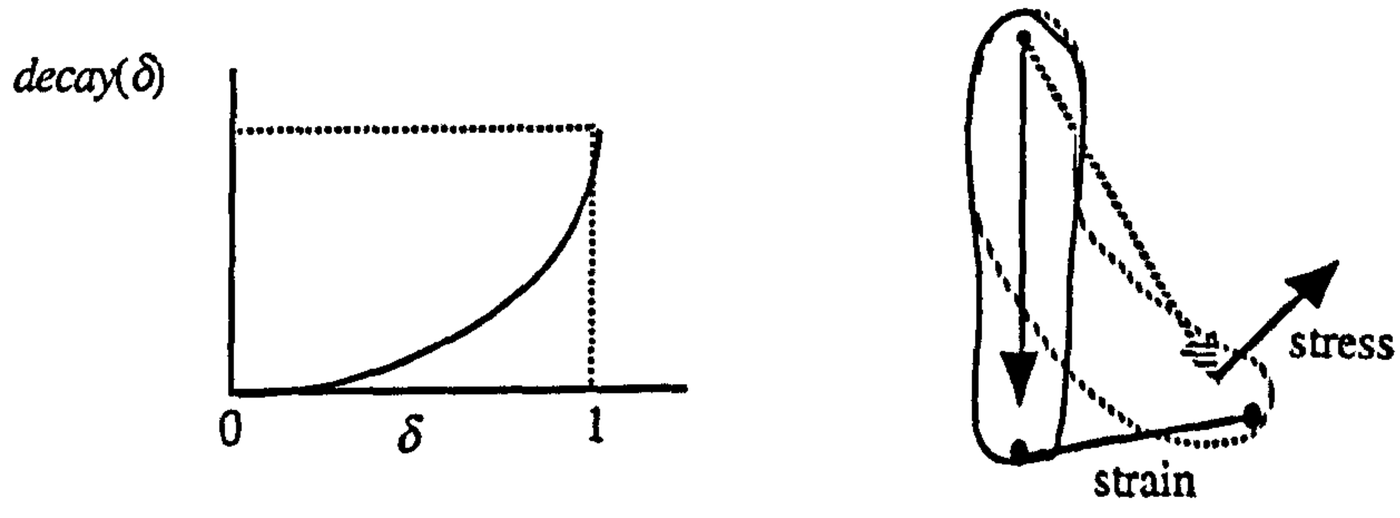


Figure 7: Decay-of-strain function and bending strain caused by an applied stress.

The spatial dumping of strain is described by a *decay-of-strain* function, which is directly controlled by the animator (see figure 7). The decay-of-strain curve provides with a scalar function *decay*(δ), which is then used in combination with a *dumping factor*, γ, to dump $\Delta q(t)$:

$$\Delta q'(t) = \Delta q(t) * (1 - \gamma * decay(\delta)) \qquad [5]$$

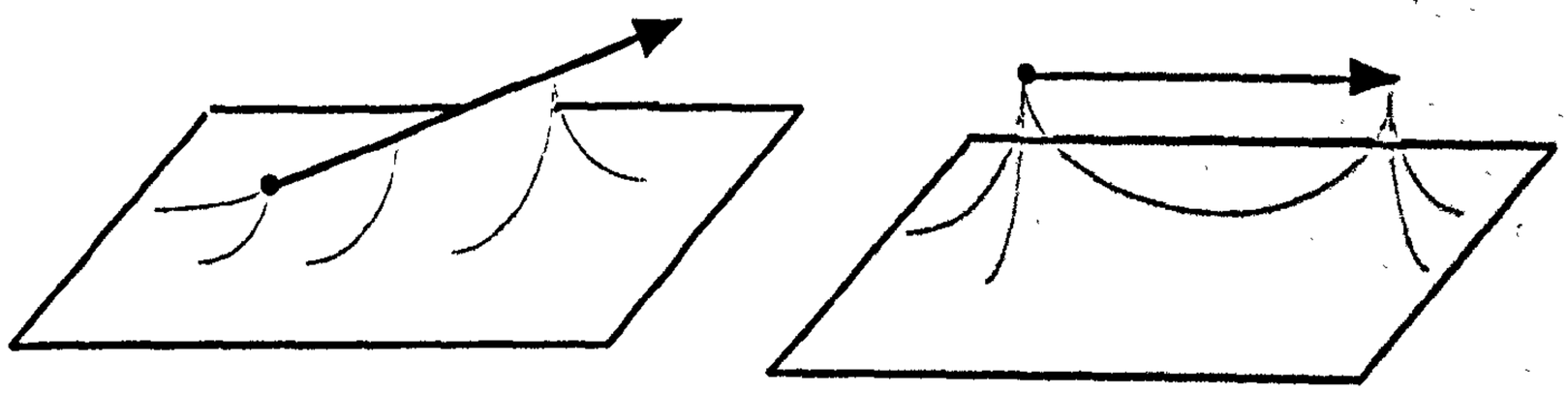
where	$0 \leq \delta \leq 1$	the normalised distance (see §3.3).
and	$0 \leq \gamma \leq 1$	
when	$\gamma = 0$	no bending strain.
when	$\gamma = 1$	bending strain is dictated fully by *decay*(δ).

$\Delta q(t)$ represents the total local strain and when it is dumped to $\Delta q'(t)$, all three basic strains are dumped together. Therefore, the dumping factor, γ, controls the total flexibility of the model:

when $\gamma = 0$	rigid behaviour.
when $\gamma = 1$	flexible behaviour.

3.2.2 Stress / Strain Application

In the physical metaphor, a rigid pin may by used to apply stress/strain on a flexible object in various ways. For the purpose of this document, only two distinct methods will be examined. In the first method a VOO-pin is used in such a way that it distributes stress/strain along its entire length (see figure 8.a). Imagine the VOO-pin immersed inside a piece of modelling clay. This type of VOO will be referred to as *full-length* VOO. In the second method, only the tips of the VOO are capable of applying stress/strain (see figure 8.b). Imagine the VOO-pin attached to a flexible membrane. This second type of VOO will be referred to as *tip-only* VOO.

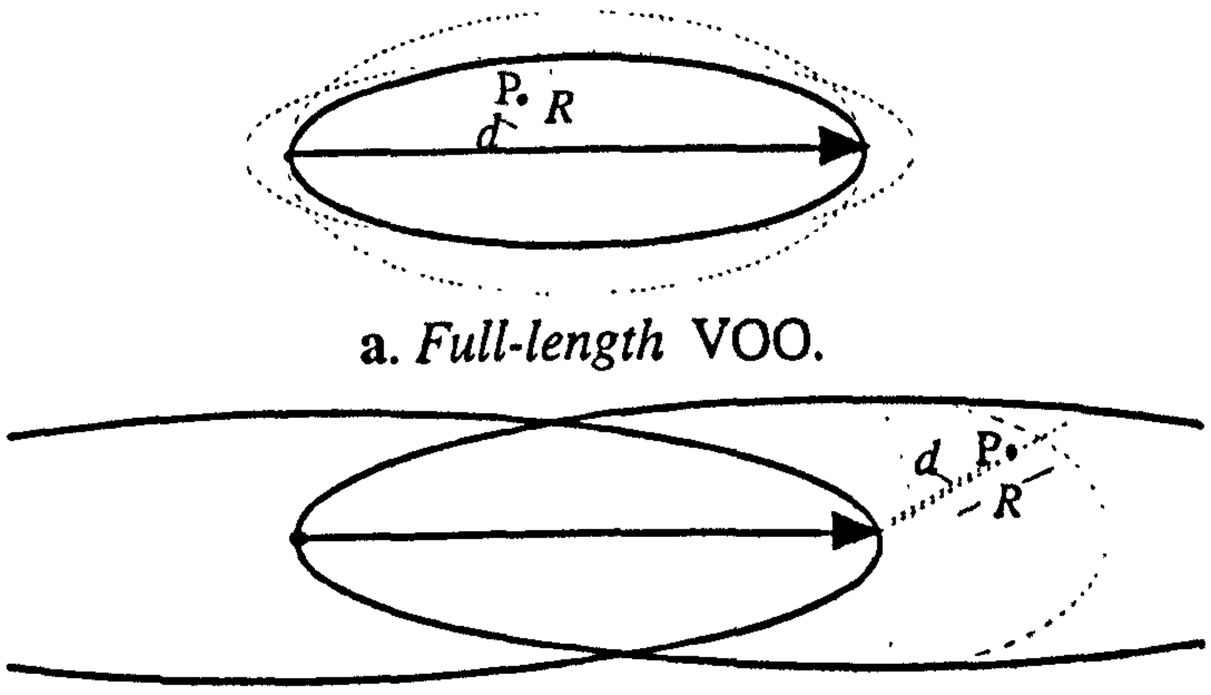


a. *Full-length* VOO. b. *Tip-only* VOO.

Figure 8: The two basic types of VOO.

3.3 Field of Influence

For the purpose of localising the global deformation effect, that is effected by a VOO, the space around the VOO may be enclosed by an implicit surface. This will offer a finer control to the animator and will allow manipulation of complex concave topologies with greater freedom. For simplicity, the enlimiting surface is taken to be an ellipsoid (in fact a superquadric is used) (see figure 9).



a. *Full-length* VOO.

b. *Tip-only* VOO.

Figure 9: Ellipsoids as fields of influence.

The two types of VOO apply local deformation on the surrounding space inside the field as shown in figure 9.a and b. The ellipsoid or pair of ellipsoids may be scaled freely by the animator. In the case of a *tip-only* VOO, the area of any of the two ellipsoids that extends beyond the length of the VOO itself, may by scaled/squashed as shown in figure 9.b. For an arbitrary location P inside the field of influence, there are two lengths that must be calculated, d and R (see figure 9). These lengths are used for the calculation of the normalised distance, δ, that appears in the calculation of the decayed total strain (see equation [5]):

$$\delta = d \; / \; R \qquad [6]$$

As for the space where the two ellipsoids overlap (see figure 9.b), a decay amount is calculated for each field separately and then the amounts are weighed.

3.4 Locality & Weight

Two further fine tuning parameters may control the *locality* and the *weight* of a VOO deformation. *Locality*, λ , is a factor that is added to the distance d (see equation [6]) and thus locally reduces or expands the influence of a VOO in space. *Weight*, ω , is a factor that is applied as an exponential to the total decay amount and thus intensifies or diminishes the influence of a VOO in space. Similar parameters were introduced in (Beier and Neely 1992). So equation [5] finally becomes:

$$\Delta q'(t) = \Delta q(t) * (1 - \gamma * decay((d + \lambda) \; / \; R))^{\omega} \qquad [7]$$

3.5 Discussion

A similar technique to the one presented hereby, was used in (Beier and Neely 1992) for "feature-based image metamorphosis" and later in (Lerios *et al.* 1995) for "feature-based volume metamorphosis". In (Hilton and Egbert 1994), "vector fields" were implemented based on the combination of global deformations and superquadric influence fields.

A VOO and its field of influence offers a direct way of local strain application on flexible objects. By locating a VOO and scaling the field of influence appropriately, the animator specifies a local area of the topology to be deformed. By offsetting anyone of the two tips of a VOO in space, the animator may visually control the shape and the amount of strain desired. A further degree of freedom is added by allowing rotation of a VOO about its own axis **u**. Such a rotation can apply torsion strain (twist) onto an object. By offsetting both tips of a VOO, the animator may apply rigid-body control (global translation and rotation) onto the entire object. The direct control of the permissible strains and the decay-of-strain function allow for fine control over the shape of a deformation at interactive rates.

4 Dynamic Control

Apart from manipulating a VOO in a direct interactive way, the animator may wish to specify dynamic constraints that will convey the motion of a VOO. In the proposed methodology the focus is put on fast, effective and intuitive ways of dynamic constraint specification that are well suited to the animation practice.

4.1 Mass-point Moving on Specified Trajectory

A point with constant mass, m , may be forced to move on a specified 3D trajectory under the influence of an external force function, $\mathbf{F}(t)$ (see figure 10). In the following simplified formulation there is no gravity or viscous damping taken into account. When the force function, $\mathbf{F}(t)$ and initial velocity, υ_0, at time t_0 are known, then the distance travelled by the mass-point on the trajectory, $s(t_1)$, at time t_1, can be calculated as follows:

$$s(t_1) = \upsilon_0 + (1 / m) * \int_{t_0}^{t_1}\int \mathbf{F}(t)\, dt^2 \qquad [8]$$

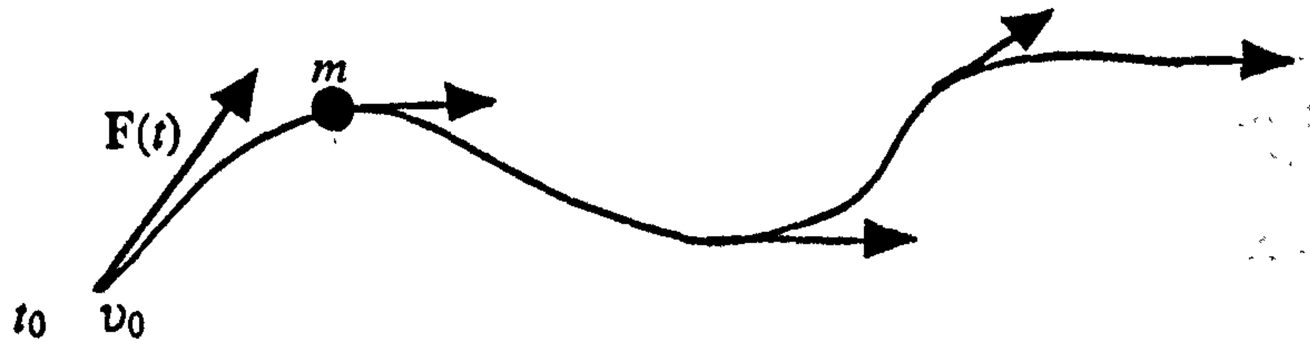


Figure 10: A mass-point on a path under the influence of a force function.

Equation [8] is an ordinary differential equation (ODE) of first degree and can be solved efficiently and with adequate approximation using the fourth order Runge-Kutta method with adaptive step-sizing (Press *et al.* 1986).

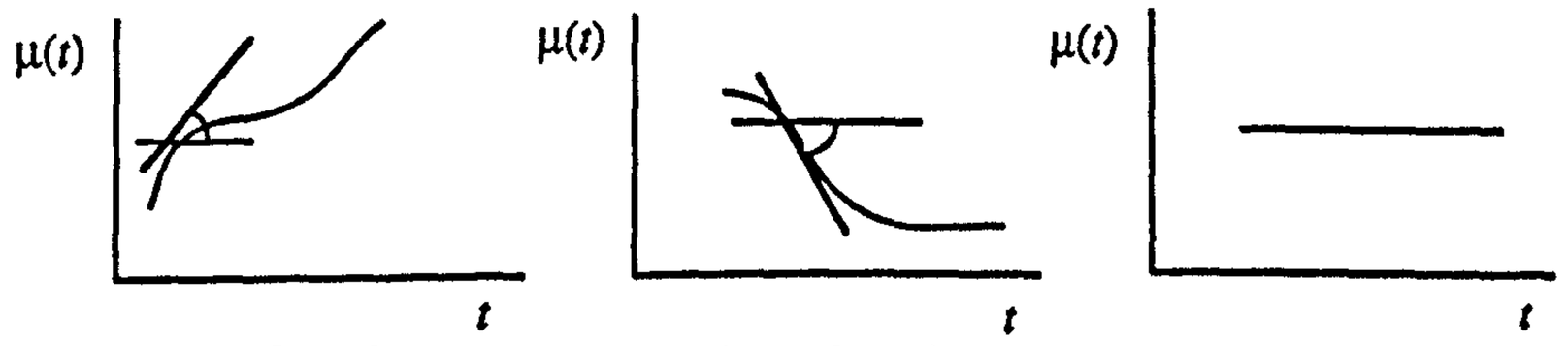
The trajectory or *motion path*, in figure 10, may be directly described by the animator as an

interpolation of key-points or it may be derived via motion capture. For the mass-point to keep travelling on the *motion path*, the force function $\mathbf{F}(t)$ must be combined with a constraint force function $\mathbf{R}(t)$, which is unknown. The total external force $\mathbf{F}(t) + \mathbf{R}(t)$ would then have to replace $\mathbf{F}(t)$ in equation [8]. Assuming that the point-mass will always travel on the *motion path*, the velocity and acceleration will be tangent to the path. Thus the animator need only provide with the magnitude of the the total external force function. An intuitive way of doing so is via a *momentum/time function* (see figure 11). The momentum, $\mu(t)$, of a point-mass moving under an external force function is defined as follows:

$$\mu(t) = m * \upsilon(t)$$

With the momentum function, $\mu(t)$, known, the total external force is calculated as the first derivative of $\mu(t)$:

$$\mathbf{F}_{tot}(t) = d\mu(t) / dt$$



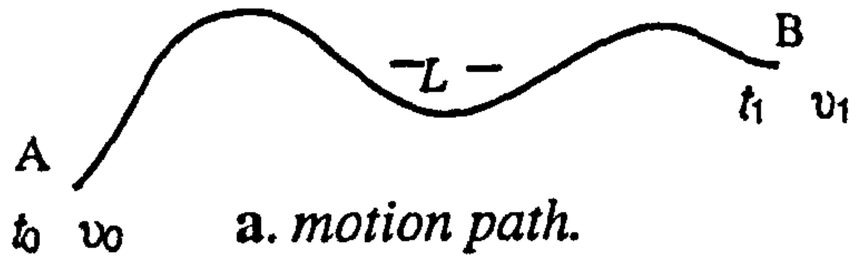
a. acceleration. b. decceleration. c. constant velocity.

Figure 11: The "ups" in the momentum curve generate accelerations and the "downs" generate deccelerations.

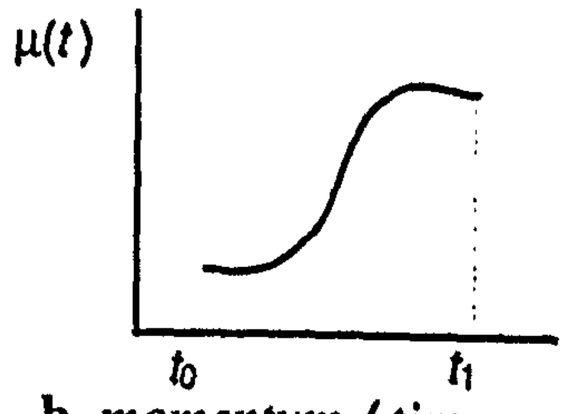
The *momentum/time function* must have a finite first derivative at any time: $t_0 \leq t \leq t_1$.

4.2 Initial/Final Values

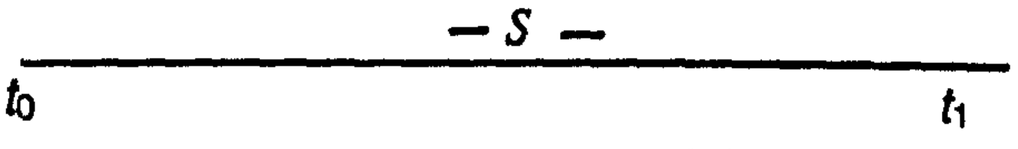
In animation the initial and final values for time and space become crucial. In most cases the duration of a sequence is determined well beforehand, depending on external time restrictions. Space is also, usually, constrained by the combined motion of several objects and the size and morphology of the animation environment (background set). Most commonly, the animator has worked out the *motion path* which describes the spatial action of an object and has predetermined the duration ($t_1 - t_0$) (see figure 12.a). By letting the animator suggest an arbitrary *momentum/time function* (see figure 12b), there is no warranty that an object starting at A at time t_0, will ever reach the end of the *motion path*, B, at time t_1. The object may overshoot at the end of the path at the direction of the last tangent or come to rest on the path without having covered the entire length, *L*.



a. *motion path.*



b. *momentum / time curve.*



c. Length travelled under the influence of the momentum function.

Figure 12: A *motion path* and a *momentum / time curve.*

The total length, S, that is dictated by the suggested *momentum/time function*, is calculated with a prointegration of equation [8]. From then on, the scale factor (L/S) is incorporated in equation [8], in order to scale the suggested curve to the initial/final constraints set by the animator:

$$s(t_1) = (L / S) * (v_0 + (1 / m) * \iint_{t_0}^{t_1} \mathbf{F}_{tot}(t)\, dt^2) \qquad [9]$$

The final velocity at B, is important when the object needs to "jump" from one motion path to another and is calculated as follows:

$$v(t_1) = (L / S) * (v_0 + (1 / m) * \int_{t_0}^{t_1} \mathbf{F}_{tot}(t)\, dt)$$

4.3 Discussion

The notion of separating spatial from temporal control in animation is also suggested in (Snibbe 1995). In the presented methodology this separation is achieved via the *motion path* and the *momentum/time function*. A pair of *motion paths* and *momentum/time functions* may be used to constrain both the tips of a VOO. This will generate a constrained dynamic deformation on the surrounding topology. The advantage of this method is the efficiency that allows the animator to manipulate the momentum curve, at interactive rates and visually judge the result. This fine tuning may easily lead to a desired and properly constrained motion.

5 Results

The introduced *Vector Offset Operators* offer intuitive control of flexible behaviour. The efficiency of the implementation allows for direct manipulation of large polygonal topologies via offsetting and twisting VOOs. The *permissible strains* and *decay-of-strain function* along with the *field of influence* of a VOO enable the harnessing of flexible behaviour and offer maximum control to the animator, preserving, though, the natural look of deforming objects. The spatial and dynamic constraints render a VOO a powerful and easy to use stress/strain application tool for flexible or rigid objects. Mass, in fact, becomes a property of the space around a VOO rather than a property of the deforming object. The suggested approach

disassociates the dynamic formulation problem from the deformation problem which is further disassociated from the geometric representation problem. This schema increases the modularisation, generality and extendibility of the system. An interactive test-bed application was implemented for the HP Apollo 730 and linked to the animation system CGAL®.

6 Future Work

This paper presents part of the author's PhD research. The author is currently looking at the development of a fast methodology for the incorporation of physical properties such as elasticity and viscoelasticity in the presented schema. The aim is to provide with an interactive tool for the animation of organic material behaviour such as the behaviour of skin, muscle and biological fluids. A further aim is to enable hierarchical linking of sets of VOOs.

Acknowledgments

I would like to gratefully acknowledge the support of the supervisory team, Prof. P.P. Comninos and Prof. J.A. Vince. I wish to thank the rest of the staff and the students at the National Centre for Computer Animation for their suggestions and criticism. Also, I thank CGAL ltd for allowing me to link to their system. Finally, I wish to thank Hewlett Packard for Sponsoring the Centre with their powerful machinery.

References

Alias / Wavefront (1997). *Learning Alias v8*. Silicon Graphics Ltd.

Barr, A.H. (1984). Global and Local Deformations of Solid Primitives. In: *COMPUTER GRAPHICS Proceedings, Minneapolis MN '84*. ACM SIGGRAPH, 21-30.

Beier, T. and Neely, S. (1992). Feature-Based Image Metamorphosis. In: *COMPUTER GRAPHICS Proceedings, Chicago IL '92*. ACM SIGGRAPH, 35-42.

Fung, Y.C. (1965). *Foundations of Solid Mechanics*. Englewood Cliffs NJ, Prentice Hall.

Hilton, T.L. and Egbert, P.K. (1994). Vector Fields: An Interactive Tool for Animation, Modelling and Simulation with Physically Based 3D Particle Systems and Soft Objects. In: *EUROGRAPHICS Proceedings, Barcelona SP '94*. Blackwell Publishers, 329-338.

Lerios, A., Garfinkle, C.D. and Levoy, M. (1995). Feature-Based Volume Metamorphosis. In: *COMPUTER GRAPHICS Proceedings, Los Angeles CA '95*. ACM SIGGRAPH, 449-456.

Platt, J.C. and Barr, A.H. (1988). Constraint Methods for Flexible Models. In: *COMPUTER GRAPHICS Proceedings, Atlanta GA '88*. ACM SIGGRAPH, 279-288.

Press, W.H., Flannery, B.P., Teukolsky, S.A. and Vetterling, W.T. (1986). *Numerical Recipes: The Art of Scientific Computing*. Cambridge UK, Cambridge University Press.

Snibbe, S.S. (1995). A Direct Manipulation Interface for 3D Computer Animation. In: *EUROGRAPHICS Proceedings '95*. Blackwell Publishers, 271-283.

Softimage 3D (1997). *Reference Guide*. Version 3.5, Softimage Inc., Microsoft Corporation.

Terzopoulos, D., Platt, J., Barr, A. and Fleischer, K. (1987). Elastically Deformable Models. In: *COMPUTER GRAPHICS Proceedings, Anaheim CA '87*. ACM SIGGRAPH, 205-214.

Terzopoulos, D. and Fleischer, K. (1988). Modelling Inelastic Deformation: Viscoelasticity, Plasticity, Fracture. In: *COMPUTER GRAPHICS Proceedings, Atlanta GA '88*. ACM SIGGRAPH, 269-278.

Witkin, A. and Kass, M. (1988). Spacetime Constraints. In: *COMPUTER GRAPHICS Proceedings, Atlanta GA '88*. ACM SIGGRAPH, 159-168.

Witkin, A. and Welch, W. (1990). Fast Animation and Control of Nonrigid Structures. In: *COMPUTER GRAPHICS Proceedings, Dallas TX '90*. ACM SIGGRAPH, 243-250.

The following paper was published in EDUGRAPHICS & COMPUGRAPHICS Combined Proceedings, Vilamoura Portugal 1997 (Hurmusiadis 1997).

B.2 Fast Implementation of Viscoelastic Creep Suitable for Manipulation of Organic Objects

Abstract

The author addresses the problem of organic material behaviour in computer graphics objects. The majority of living tissues in vertebrate or invertebrate creatures exhibit a complex mechanical behaviour that may be characterised as viscoelastic. The presented research is focused on efficient direct and dynamic control of organic objects, whose behaviour is a combination of plasticity, elasticity and viscoelasticity. An intuitive and efficient approach to stress and strain control is introduced through the use of Vector Offset Operators. Vector operators are used for the specification and control of fast local deformations of triangulated polygonal surface topologies. External forces are applied via user defined pairs of motion paths and momentum/time function curves. A user defined stress/strain function curve controls the elastoplastic behaviour of the material. An, also, user defined viscoelastic creep/time function curve controls the viscoelastic behaviour by enforcing a space and time related rate of strain through a deformable body. The resulting model may be used to achieve efficient interactive control and animation of visco-elasto-plastic objects, using a bottom of the range workstation.

Key Words: physically-based modelling, organic material behaviour, viscoelasticity, elasticity, plasticity.

Introduction

In recent years there has been a growing interest in animation and interactive control of characters based on living creatures. Vertebrate or invertebrate creatures consist of soft biological tissues like muscle, skin and fatty tissue. Bones are usually considered rigid, although this is not always true, i.e. in birds, bones are considerably flexible. All these tissues exhibit mechanical behaviours of great complexity. This is primarily because of their high water content and the fact that they, mostly, appear in layers of materials with different properties (Fung 1981). A viscoelastic material exhibits the phenomenon of strain creep. Under constant stress application, the strain does not develop instantly, but it rather creeps over time inside the body (Christensen 1982). It is this rate of strain propagation inside a deforming body that characterises viscoelasticity and it generates the complex behaviour exhibited by most organic materials. Other characteristic phenomena in viscoelastic materials are stress relaxation and stress/strain hysteresis. Stress relaxation occurs when the strain is kept constant for a considerable length of time. In such case, the internal elastic stresses gradually diminish and the strain is being converted to permanent plastic deformation. Hysteresis is manifested by a deviation from a stress/strain graph during a loading and unloading process.

Several researchers have suggested elastic models, based on simplifications of elasticity theory, assuming perfectly elastic behaviour. In (Terzopoulos *et al.* 1987) an elastic model is based on the construction of a set of mass nodes, interconnected with springs. Their model, is capable of producing realistic results, but it requires numerical solution of a large system of differential equations, which is computationally demanding and prohibits fast implementations on low-end workstations. A different, hybrid approach may combine the efficiency in deformation, provided by global and local deformation models (Barr 1984]), with simplified elasticity in

order to achieve interactive rates with elastically behaving objects. In (Witkin and Welch 1990) a hybrid elastic global deformation model is controlled by motion and attachment constraints. Their implementation allows interactive control of articulated puppets made of rubber. Several other physically-based elastic models have been applied in modelling flexible animated characters (Chadwick *et al.* 1989), in biomechanical simulation of animal muscle (Chen and Zeltzer 1992), in human character modelling (Turner and Thalmann 1993), in human facial animation (Waters 1987, Terzopoulos and Waters 1990, Lee *et al.* 1995), etc. Although accurate and efficient, perfectly elastic models, unavoidably, produce a look of synthetic rubbery materials.

More elaborate physically-based deformable models may be developed by extending a standard elastic model to incorporate the properties of plasticity and viscoelasticity. (Wu *et al.* 1995) proposed a particle system for the purpose of animating an elastoplastic deformable surface, whose behaviour is described by a user defined stress/strain graph. They applied their method on animating skin with wrinkles and other surfaces. (Terzopoulos and Fleischer 1988) extended the elastic model, presented in (Terzopoulos *et al.* 1987), to incorporate viscoelastic and plastic deformations. Based on classical mechanics theory, they suggested that viscoelastic behaviour may be approximated by the combined use of a conceptual elastic unit (i.e. a spring) and a viscous unit (i.e. a dash pot). A linear elastic unit obeys Hooke's Law: $f = ke$, where f is the external force, k is the spring constant and e is the deformation. In a linear viscous unit the rate of deformation, de/dt, is proportional to the external force: $f = n(de/dt)$, where n is a viscosity constant. A mechanical viscoelastic model may be constructed by connecting one or more elastic units to one or more viscous units in series or in parallel (Fung 1981). Mechanical models exhibit a form of idealised, synthetic viscoelastic behaviour, which differs from real organic material behaviour. Moreover, the (Terzopoulos and Fleischer 1988) model poses difficulties in space and time control. Additionally, the resulting large system of simultaneous differential equations poses a computational overhead, which renders a 3D implementation of their viscoelastic model a super-computer task.

This paper introduces a visco-elasto-plastic extension built upon a core local deformation model. The local deformation model is built around deformation control primitives, vector offset operators (VOOs), which can be directly or dynamically manipulated. External forces are applied onto these primitives via user defined motion path constraints and momentum/time function curves. The elastoplastic material behaviour is described by a user defined stress/strain function curve, which determines the speed of elastic restoration and the amount of permanent plastic deformation. The viscoelastic material behaviour is described by an, also, user defined creep/time function curve, which enforces a rate of strain propagation from the undeformed reference body to a deformed body at time t. These two features are combined to generate complex organic behaviours which can easily be fine-tuned and customised to achieve the desired deformation. Stress relaxation is also incorporated using a user defined relaxation/time function curve.

The suggested approach is capable of achieving interactive rates in manipulating visco-elasto-plastic objects of considerable size, on low-end workstations or high-end PCs. Vector offset operators offer direct spatial and temporal control over complex objects. The overall behaviour of an object can be fine tuned to resemble real material behaviour by importing experimental data into the creep/time and stress/strain function curves. Alternatively, the user can experiment by interactively modifying default curves and, thus, custom-design a real or unreal material behaviour.

Plastic Model

Vector Offset Operators

A Vector Offset Operator (Hurmusiadis 1996) is a local deformation primitive which can be used as a modelling and animation tool for deformable objects. The physical counterpart of a VOO is that of a rigid pin immersed inside a real object. When inside a deformable object, any movement of the pin will move and deform the object. When inside a rigid object, though, the pin will force the object into rigid motion. A VOO is, in fact, a fixed vector with magnitude, direction and position. A VOO is independent from the topologies that may surround it and it is used for the construction of a local frame of reference. The user may directly specify a set of VOOs which may be placed at any position or orientation in relation to the surrounding objects. A direct offsetting of the tips of a VOO causes local space transformations, which in turn are passed on to the surrounding topological elements.

Local Deformation

Local deformation is achieved by successive mappings from a local frame of reference at rest state, L, to a local frame, L(t), at time t (see figure 1).

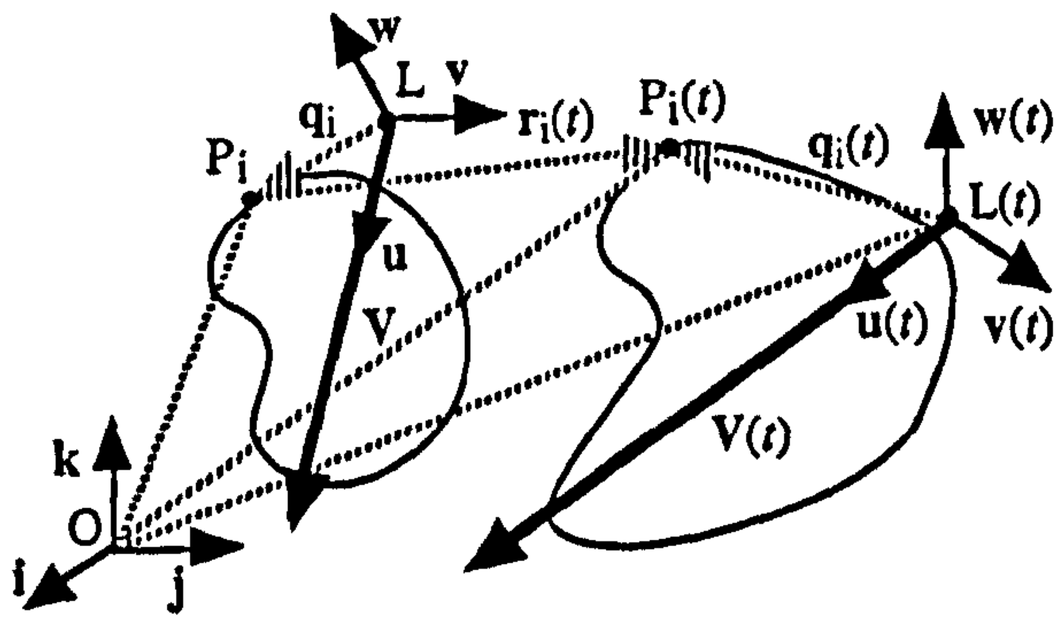


Figure 1: V represents a VOO at rest state and V(t) represents the same VOO at time t.

Local frame, L(t), follows the VOO, V(t), in space and time. A VOO may undergo a concatenation of global translation, rotation and local rotation and scaling transformations. As a result, a vertex P_i of a nearby topology, is being transformed to $P_i(t)$. The total displacement from P_i to $P_i(t)$ is represented by vector $\mathbf{r}_i(t)$ and is extracted from figure (1) (see equation [1]).

$$\mathbf{r}_i(t) = \mathbf{OP}_i(t) - \mathbf{OP}_i \Rightarrow$$

$$\mathbf{r}_i(t) = (\mathbf{OL}(t) - \mathbf{OL}) + (\mathbf{q}_i(t) - \mathbf{q}_i) \Rightarrow$$

$$\mathbf{r}_i(t) = \mathbf{T}(t) + \mathbf{Q}_i(t) \qquad [1]$$

Subscript $_i$ denotes a vertex of the topology. Vector $\mathbf{T}(t)$, in equation [1], represents global translation and vector $\mathbf{Q}_i(t)$ represents local rotation and scaling. The separation of these two elements in the total displacement vector, in fact, helps separate rigid motion in global space from deformation in local space. This enables further treatment of strain vector $\mathbf{Q}_i(t)$.

Strain Control

When an external stress is applied on a small area of a deformable object, the afflicted strain is being transferred onto neighbouring areas, thus, spreading on a greater part of the volume of the object. In a flexible material, though, the connecting forces among neighbouring molecules are weak and allow for local deformations which absorb deformation energy. The result of this process is a spatial damping of the spreading strain, which grows proportionately to the distance from the stress application centre. This spatial damping can be graphically described by a user defined strain damping function curve (see figure 2).

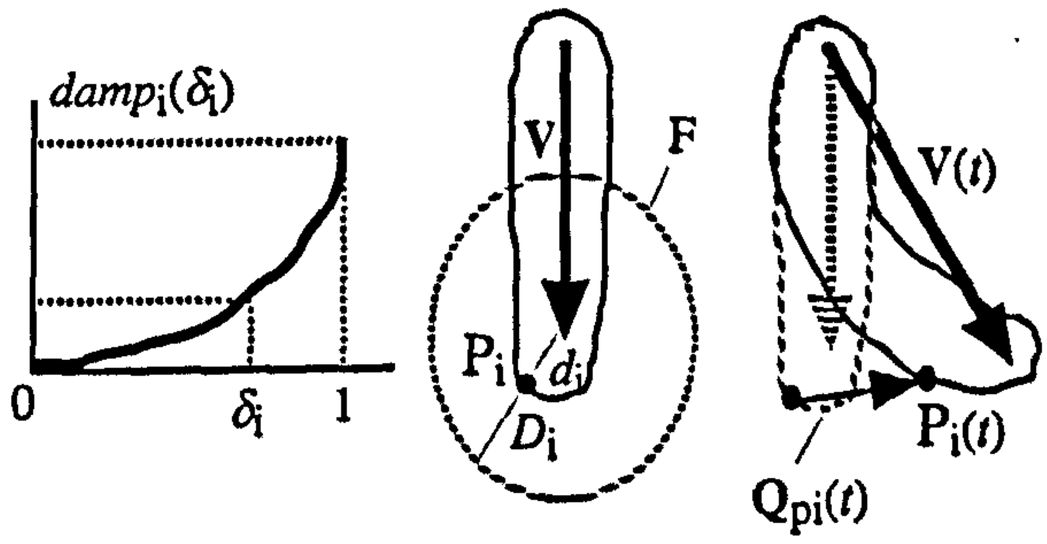


Figure 2: A strain damping function curve and its effect on spatial strain damping.

In figure 2, d_i represents the distance of vertex P_i from the tip of the VOO and D_i represents the maximum distance from the tip to the edge of a field of influence, F. The normalised distance δ_i is then given by

$$\delta_i = \frac{d_i}{D_i} \qquad [2]$$

The damping function curve (see figure 2) produces a damping percentage, $damp_i(\delta_i)$, of the deformation, that will reach vertex P_i, when VOO is deformed. The plastic strain vector $Q_{pi}(t)$ (from equation [1]) may become

$$Q_{pi}(t) = Q_i(t) \cdot \left(1 - \gamma \cdot damp_i(\delta_i + \lambda)^{\omega}\right) \qquad [3]$$

In equation [3], $Q_{pi}(t)$ is the spatially damped strain vector for vertex P_i. The damping coefficient γ controls the overall rigidity of the space inside the field of influence of a VOO. When $\gamma = 0$, no damping of strain is applied and the topology behaves like a rigid object. When $\gamma > 0$, the damping coefficient acts as a scaling factor for the amount of spatial damping received by each vertex. Factor λ, acts as a locality coefficient and factor ω acts as a weighing coefficient for the deformation effect. The set of damping values that correspond to all the active vertices of a topology (see $(1-\gamma damp_i(\delta_i+\lambda)\cdot)$ in equation [3]) may be precalculated. While a VOO is being interactively manipulated, the precalculated damping values are multiplied by the rigid deformation vectors Q_i. These damping values need only be recalculated every time a new VOO is added to the scene or when the influence field of a VOO is modified.

A perfectly plastic material yields under any applied stress and it accepts any resulting strain as permanent deformation (Mendelson 1968). Under a limited stress range, modelling clay behaves like a perfectly plastic material. In the present formulation, the behaviour of a plastic material may well be simulated using the deforming properties of a VOO. Since there are no restoring forces to counteract the external ones, it seems logical to directly apply strain instead

of stress onto the model. Elongation, bending and torsion strain can be applied to a topology by offsetting the tips of a VOO and by rotating a VOO about its major axis, **u**. VOOs can be connected to each other by constraining a tip of one VOO and a tip of another. This allows the creation of structures of VOOs, which may be used to deform space and objects in it.

Stress Control

The application of external stress on a deformable object is facilitated by the application of forces directly onto the tips of a VOO.

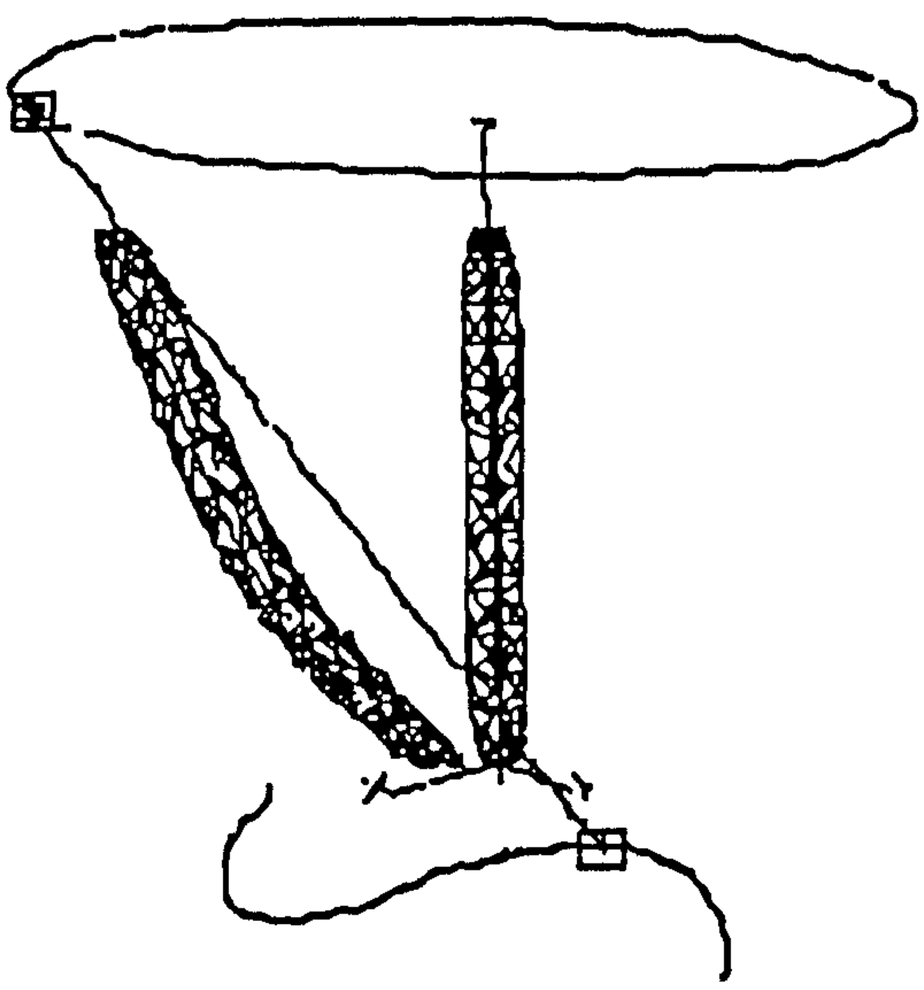


Figure 3: Both tips of a VOO are constrained on motion paths.

Constraining a tip of a VOO onto a 3D motion path and assuming that a mass, m, is concentrated on the tip, the problem of stress application turns into the problem of a mass-point travelling on a specified trajectory under the influence of a force function (see figure 3). A spatial motion path dictates the direction for the total external force and, hence, the direction of the velocity and acceleration at every time step. The magnitude of the total external force may be derived from a user defined momentum/time function curve (see figure 4).

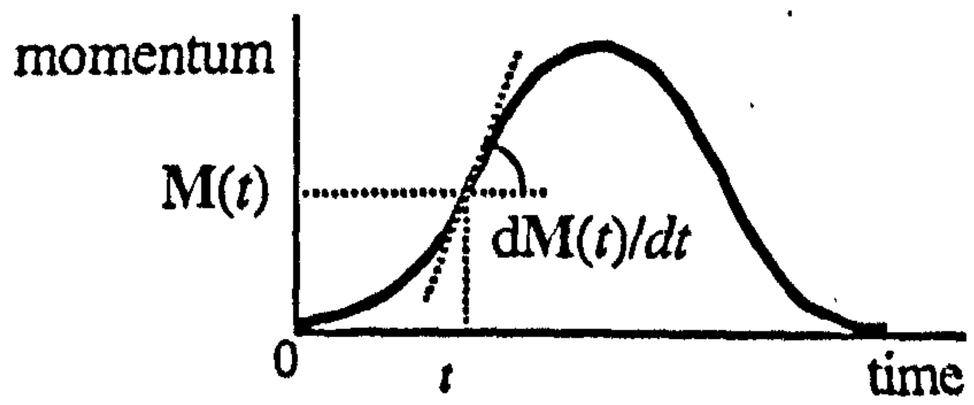


Figure 4: A momentum/time function curve.

$$\mathbf{F}(t) = \frac{d\mathbf{M}(t)}{dt} \qquad [4]$$

In equation [4], $\mathbf{M}(t)$ is the momentum of the travelling mass-point at time t. The total external force function, $\mathbf{F}(t)$, is derived by differentiating the momentum function. Lagrange's equation of motion can be used to describe the movement of the mass-point on the path. This results to

an ordinary differential equation of first degree, which can be solved efficiently and with adequate approximation, using the fourth order Runge-Kutta method with adaptive step sizing (Press *et al.* 1986). The equation need only be solved once per motion constrained VOO tip and not per object vertex.

Elastoplastic Model

A perfectly elastic material restores its reference shape immediately after removal of the external stress. In elastoplastic materials, though, if the stress exceeds a certain level, the strain cannot be restored completely. Permanent internal damage on the molecular structure causes an amount of permanent strain. The relationship between strain and the restoring stress may be specified with the aid of a user defined stress/strain function curve, which describes a general elastoplastic behaviour (see figure 5).

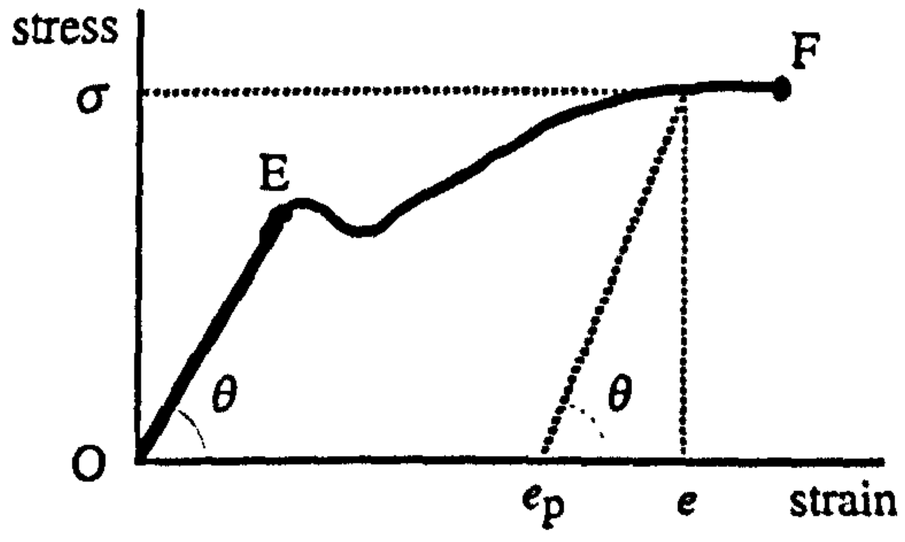


Figure 5: A typical stress/strain function curve of an elastoplastic material.

In figure 5, E represents the elastic limit, F the fracture limit and $k = \tan(\theta)$ is the elastic spring constant. Any strain that falls inside the (OE) region, is completely restored. A larger strain, though, that falls inside the elastoplastic region (EF) is only partially restored, leaving a permanent plastic strain e_p, on the object.

In the suggested deformation model, a change in the magnitude and direction of a VOO causes deformation to local space and to any objects inside it. So, instead of attempting to elastically restore each deformed vertex of an object, the model attempts to restore a VOO to its initial state. To achieve this, it is assumed that an elastic spring is attached at the tip of the VOO at rest state and at the same tip of the VOO at time t_0. It is, also, assumed that the initial length of the spring is equal to 0.

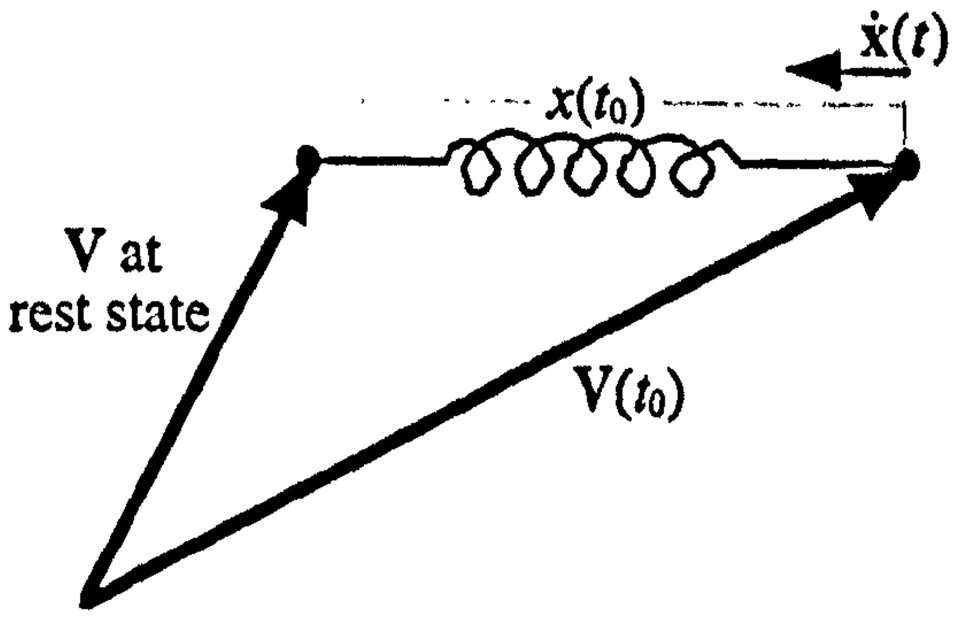


Figure 6: A spring connects the tips of **V** and $\mathbf{V}(t_0)$.

The rate of elastic restoration may be calculated assuming that all the elastic energy of the deformation is converted to kinetic energy.

$$\dot{x}(t) = \sqrt{\frac{k}{m}\left(x_{max}^2 \cdot \zeta(t) - x(t)^2\right)} \qquad [5]$$

$$\text{where} \quad x_{max} = |\mathbf{V}(t_0) - \mathbf{V}|$$

In equation [5], k is the spring constant and m is a mass point concentrated on the tip of the VOO.

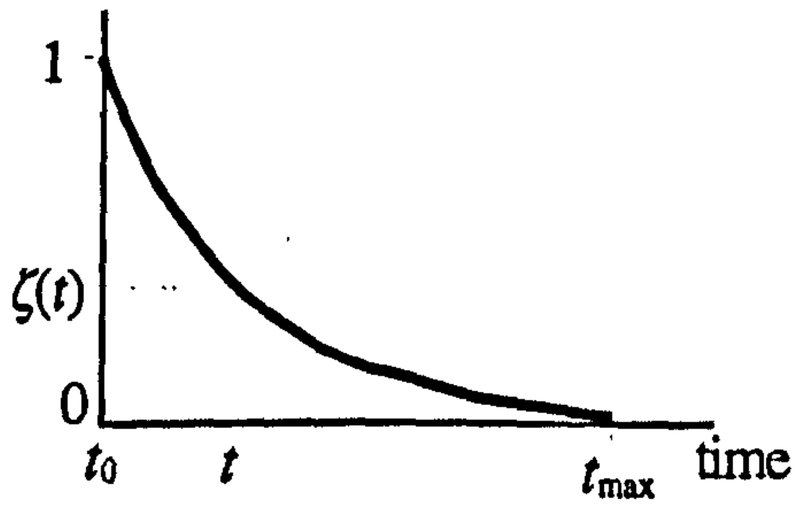


Figure 7: Energy damping function.

The factor $\zeta(t)$ is a damping function, which may be used to damp the total energy of a spring-mass system (kinetic + potential energy) over a specified period of time (t_{max} - t_0). Function $\zeta(t)$ may be specified by a user defined graph (see figure 7).

The efficiency of the proposed method is based on the fact that the elastic restoration process is only applied to the time related state of a VOO. So, the process does not attempt to restore individual topological elements.

Viscoelastic Model

Organic materials, such as living tissue in physiological state, are viscoelastic (Fung 1981). The phenomena of strain creep, stress relaxation and hysteresis are characteristic features of viscoelastic material behaviour. An automated model for interaction with and animation of viscoelastic objects; could replace a great amount of repetitive animation work. As a result, the creation of animated characters based on living creatures would be greatly facilitated. The deformation model, introduced in the previous sections, is extended to incorporate the property of viscoelasticity.

Strain Creep

Organic materials exhibit the viscoelastic feature of strain creep. In such materials, strain is not applied instantaneously on the entire body of an object, as it has been assumed so far in this study, but it rather develops in time. Therefore, strain creep is an inherently time related phenomenon. Strain creep creates a fluid like motion and, if constrained properly, it may be used in the animation of organic characters and natural phenomena.

A VOO may be subjected to time based motion. As a result, a strain rate is imposed on every active vertex of an influenced object. Viscoelastic strain rate may be achieved by applying a spatial and temporal gradient on the strain rate (see figure 8 and 9).

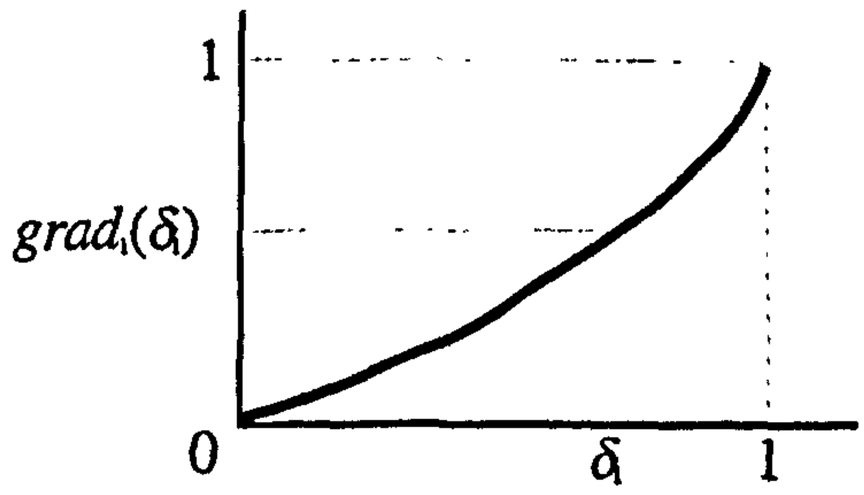


Figure 8: Spatial gradient.

The value of the spatial gradient $grad_i(\delta_i)$ depends on the relative position of a vertex inside the field of influence of a VOO and may be precalculated.

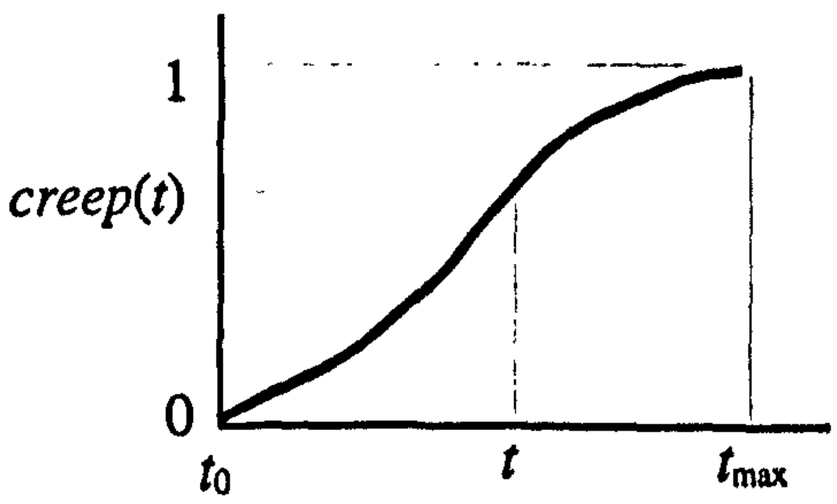


Figure 9: Temporal gradient.

The value of the temporal gradient $creep(t)$ depends on time. Both functions can be specified via user defined graphs. The viscoelastic strain rate of a vertex may then be calculated as follows

$$\dot{\mathbf{Q}}_{vi}(t) = \dot{\mathbf{Q}}_{pi}(t)_{(1-(vgrad_i(\delta_i)(1-\mu creep(t))))} \qquad [6]$$

In equation [6], v is a viscosity coefficient and μ is a strain creep coefficient. Both of these factors may be used to scale the effect of the two gradient graphs, respectively. Assuming that $\mathbf{Q}_{vi}(t_0) = 0$, the viscoelastic strain $\mathbf{Q}_{vi}(t)$ of a vertex at time t, may be calculated with the following integration

$$\mathbf{Q}_{vi}(t) = \int_{t_0}^{t} \dot{\mathbf{Q}}_{vi}(t)dt \qquad [7]$$

The global position vector of a viscoelastically deformed vertex may then be calculated as

$$\mathbf{r}_{vi}(t) = \mathbf{r}_i + \mathbf{T}(t) + \mathbf{Q}_{vi}(t) \qquad [8]$$

In equation [8], $\mathbf{r}_i$ is the global position vector of an undeformed vertex and $\mathbf{T}(t)$ is the global translation vector.

Stress Relaxation

In organic materials, internal elastic stresses decrease with time. As a consequence, the amount of elastic energy stored inside an object, is gradually being converted to plastic deformation.

The elastic restoration process becomes slower and is finally diminished after a considerable time lapse. So, the occurrence of stress relaxation in a viscoelastic material, is only expressed indirectly, through the elastic restoration process. Similarly to strain creep, stress relaxation is an inherently time related phenomenon. A user defined stress relaxation graph may be used to "relax" the internal elastic stresses (see figure 10).

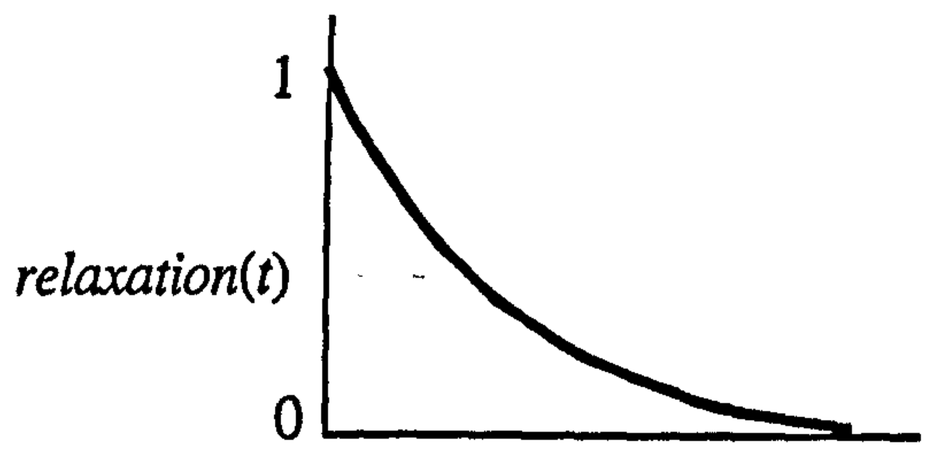


Figure 10: Stress relaxation graph.

The time dependent value of the relaxation function may be used to scale the stress axis of a stress/strain function curve (see figure 5).

The efficiency of the proposed viscoelastic model is based on the precalculation of the spatial gradient of the viscous strain rate, $grad_i(\delta_i)$, for every active vertex of an object. The strain creep, *creep*(*t*) and stress relaxation, *relaxation*(*t*), values are only calculated once per time step, so this only adds a minor computation overhead to the model. The integration time step, *dt*, may be increased to 2 or more frame steps, in order to speed up the integration process. This may deteriorate the smoothness of a deforming object, but it will, nevertheless, enable interaction with viscoelastic objects of considerable size.

Results

The primary motivation for developing the presented method was to achieve an efficient and constrainable implementation of viscoelastic behaviour.

The efficiency of the model is based on precalculations of deformation damping and viscous gradient values, which greatly lighten all vertex related calculations. Efficiency in elastic restoration is achieved by solving one spring-mass system per VOO and not per object vertex. Also, efficiency in stress application is achieved using dynamic motion paths and by solving one motion ODE per constrained VOO tip. Strain creep and relaxation values are also calculated once per VOO. The following graph (see figure 11) shows the performance of the model using one VOO with viscoelastic properties operating on objects with 200..1200 vertices (deformation and drawing calculations are included).

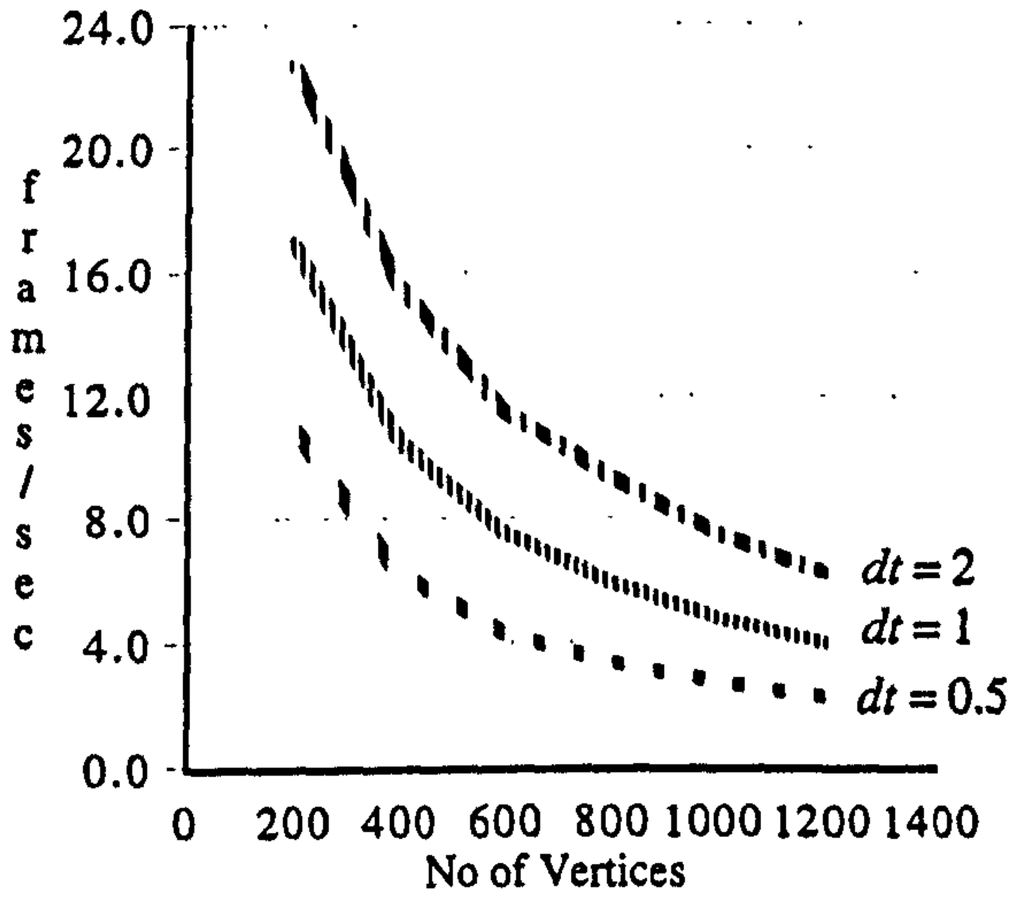


Figure 11: Graph of speed of viscoelastic deformation with varying integration step.

In figure 11, it is also shown that an increase of the integration step, can dramatically improve the performance of the viscoelastic model. However, setting the time step to 3 or more frame steps may result to noticeable discontinuities in the motion of a deforming object.

The implementation of the proposed model is modular and independent of any topological representation. This makes it a portable unit that can migrate to any platform and be plugged into existing animation software. The model was implemented into a test-bed application "VOODOO" (Vector Offset Operators for Deformable Organic Objects) on an HP-715 UNIX workstation.

The viscoelastic deformation model, presented in this paper, may be used in conjunction with stress/strain, creep/time and relaxation/time graphs, derived from experimenting with real organic materials, like skin, muscle, bone, etc. (Yamada 1970). Thus, a good approximation, yet not exact simulation, of organic properties may be, efficiently, achieved in interaction and animation of 3D computer graphics objects.

Conclusion and Future Work

Organic material behaviour is inherently difficult to constrain in time and space. However, due to increasing demand for deformable living creatures in 3D character animation and interactive applications, a fast and constrainable model of viscoelasticity is needed.

The presented methodology introduces a deformation model suitable for fast manipulation of visco-elasto-plastic objects. The introduced VOO primitive can be used as a stress or direct strain application tool. The shape of a deformation may be controlled using a spatial strain damping function curve. Dynamic constraints may be applied using motion paths and momentum/time function curves. The elastoplastic model is based on user defined stress/strain and energy damping function curves. The viscoelastic model is based on spatial gradient of viscous strain rate, creep/time and relaxation/time function curves. This approach offers maximum user control and enables fine-tuning of the material behaviour to achieve a desired look and feel.

A further development of the test-bed application VOODOO, will possibly incorporate a real-time graphics engine and an immersive VR interface.

Acknowledgements

I would like to gratefully acknowledge the support of the supervisory team, Prof. P.P. Comninos and Prof. J.A. Vince. I, also, wish to thank all the staff and students at the National Centre for Computer Animation for their suggestions and criticism.

References

Barr, A.H. (1984). Global and Local Deformations of Solid Primitives. In: *COMPUTER GRAPHICS Proceedings, Minneapolis MN '84*. ACM SIGGRAPH, 21-30.
Chadwick, J.E., Haumann, D.R. and Parent, R.E. (1989). Layered Construction for Deformable Animated Characters. In: *COMPUTER GRAPHICS Proceedings, Boston MA '89*. ACM SIGGRAPH, 243-252.
Chen, D. T. and Zeltzer, D. (1992). Pump It Up: Computer Animation of a Biomechanically Based Model of Muscle Using the Finite Element Method. In: *COMPUTER GRAPHICS Proceedings, Chicago IL '92*. ACM SIGGRAPH, 89-98.
Christensen, R.M. (1982). *Theory of Viscoelasticity*. New York NY, Academic Press.
Fung, Y.C. (1965). *Foundations of Solid Mechanics*. Englewood Cliffs NJ, Prentice Hall.
Fung, Y.C. (1981). *Biomechanics: Mechanical Properties of Living Tissues*. New York, Springer-Verlag.
Hurmusiadis, V. (1996). Vector Offset Operators for Dynamic Control of Deformable Objects. In: Jones, H. *et al*. (editors). *EUROGRAPHICS UK Chapter Proceedings, London '96*. Imperial College, London, 2, 59-71.
Lee, Y., Terzopoulos, D. and Waters, K. (1995). Realistic Modelling for Facial Animation. In: *COMPUTER GRAPHICS Proceedings, Los Angeles CA '95*. ACM SIGGRAPH, 55-62.
Mendelson, A. (1968). *Plasticity: Theory and Application*. New York, Macmillan Co.
Press, W.H., Flannery, B.P., Teukolsky, S.A. and Vetterling, W.T. (1986). *Numerical Recipes: The Art of Scientific Computing*. Cambridge UK, Cambridge University Press.
Terzopoulos, D., Platt, J., Barr, A. and Fleischer, K. (1987). Elastically Deformable Models. In: *COMPUTER GRAPHICS Proceedings, Anaheim CA '87*. ACM SIGGRAPH, 205-214.
Terzopoulos, D. and Fleischer, K. (1988). Modelling Inelastic Deformation: Viscoelasticity, Plasticity, Fracture. In: *COMPUTER GRAPHICS Proceedings, Atlanta GA '88*. ACM SIGGRAPH, 269-278.
Terzopoulos, D. and Waters, K. (1990). Physically-Based Facial Modelling, Analysis and Animation. *Journal of Visualisation and Computer Animation*. 1(2), 73-80.
Turner, R. and Thalmann, D. (1993). The Elastic Surface Layer Model for Animated Character Construction. In: *Computer Graphics International Proceedings, Tokyo'93*. Springer-Verlag.
Waters, K. (1987). A Muscle Model for Animating Three-Dimensional Facial Expression. In: *COMPUTER GRAPHICS Proceedings, Anaheim CA '87*. ACM SIGGRAPH, 17-24.
Witkin, A. and Welch, W. (1990). Fast Animation and Control of Nonrigid Structures. In: *COMPUTER GRAPHICS Proceedings, Dallas TX '90*. ACM SIGGRAPH, 243-250.
Wu, Y., Thalmann, D. and Magnenat-Thalmann, N. (1995). Deformable Surfaces Using Physically-Based Particle Systems. In: Ernshaw, R.A. and Vince, J.A. (editors). *Computer Graphics: Developments in Virtual Environments*. Academic Press, 205-215.
Yamada, H. (1970). *Strength of Biological Materials*. Baltimore, Williams and Wilkins.

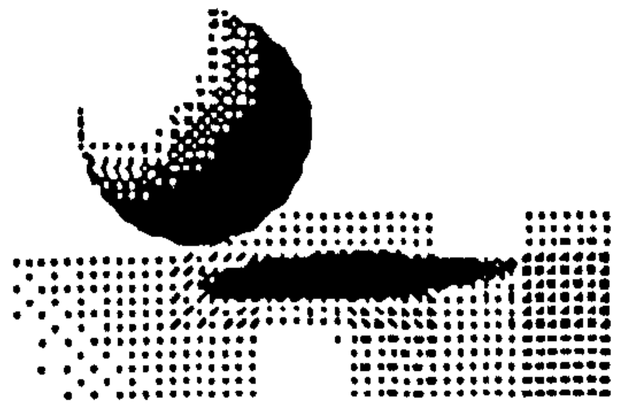


Figure 12: A viscoelastic sphere constrained by a VOO.

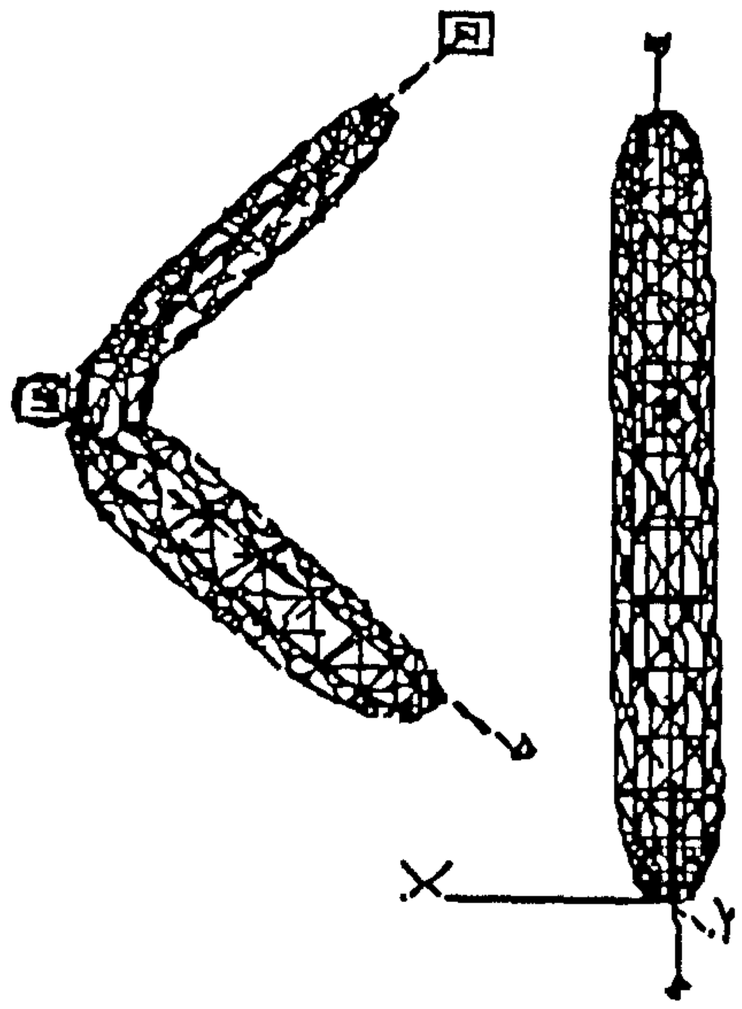


Figure 13: An articulated structure of two connected VOOs.

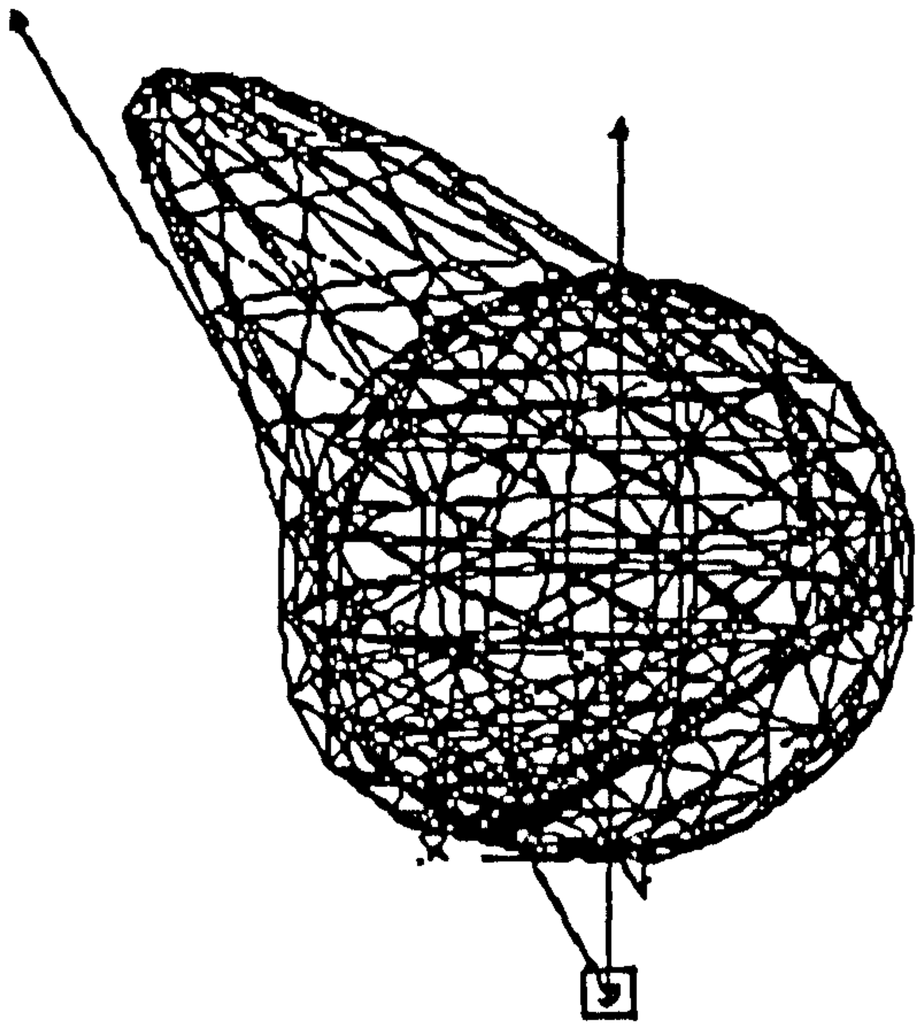


Figure 14: A VOO and a sphere at rest state and at time *t*.

List of Symbols and Abbreviations

List of Greek Symbols

$\alpha(t)$	acceleration vector of mass point at time t
$\alpha(t)$	magnitude of acceleration vector at time t
γ	flexibility coefficient
$\Delta\mathbf{V}(t)$	the difference between vectors $\mathbf{V}(t_0)$ and $\mathbf{V}$
δ_i	normalised distance of vertex v_i
ε	elongation coefficient
$\zeta(t)$	energy damping function
$\theta_b(t)$	bending strain
$\theta_t(t)$	twist strain
κ	cap coefficient of influence field ellipsoid
λ	locality coefficient
μ	strain creep coefficient
ν	viscosity coefficient
σ	nominal stress
σ_e	stress at elastic limit in a stress / strain graph
τ	strain
$\upsilon(t)$	velocity vector of mass point at time t
$\upsilon(t)$	magnitude of velocity vector at time t
υ_0	initial velocity at time t_0
ω	weight coefficient

List of Latin Symbols

$\mathbf{A}$	cross-sectional area
A	minor radius of influence field ellipsoid
$\mathbf{a}$	amplitude of harmonic oscillation
a	volume preservation coefficient
B	minor radius of influence field ellipsoid
b	cross-sectional coefficient
C	major radius of influence field ellipsoid
$creep(t)$	strain creep function
D	distance travelled by a mass point under the influence of a momentum graph
$dampf_i(\delta_i)$	spatial strain damping function
d_{maxi}	radius of influence field ellipsoid that passes from vertex v_i
E	elongation
$\mathcal{E}$	Young's modulus of elasticity
e	nominal strain
e_e	elastic strain
e_e	strain at elastic limit in a stress / strain graph
e_f	strain at fracture limit in a stress / strain graph
e_p	permanent plastic strain
$e_p(e)$	permanent plastic strain at strain e
$\dot{e}$	strain rate
$e(t)$	elongation strain at time t
$\mathbf{F}$	force vector
F	force magnitude
$\mathbf{F}(t)$	force vector acting on mass point at time t
$F(t)$	magnitude of force vector at time t
$\mathbf{F}_{el}$	elastic spring force
$\mathbf{F}_{ext}$	external force
$grad_i(\delta)$	spatial gradient of viscous strain rate of vertex v_i

Head	start of VOO vector
i	unit vector of the global frame of reference
j	unit vector of the global frame of reference
k	unit vector of the global frame of reference
L	origin of local frame of reference
L	length of path curve
l	length
$\mathbf{M}(t)$	momentum vector of mass point at time t
$M(t)$	magnitude of momentum vector at time t
m	mass
O	origin of global frame of reference
$\mathbf{Q}_i(t)$	strain vector (local deformation vector) of vertex v_i
$\mathbf{Q}_{pi}(t)$	plastic strain vector of vertex v_i
$\mathbf{Q}_{pij}(t)$	plastic strain vector of vertex v_i from jth VOO
$\dot{\mathbf{Q}}_{pi}(t)$	plastic strain rate for vertex v_i at time t
$\dot{\mathbf{Q}}_{vi}(t)$	viscous strain rate for vertex v_i at time t
$\dot{\mathbf{Q}}_{vei}(t)$	viscoelastic strain rate for vertex v_i at time t
$\mathbf{q}_i$	local position vector of vertex v_i
relaxation(t)	stress relaxation function
$\mathbf{r}_i$	global position vector of vertex v_i
$\mathbf{r}_{vei}(t)$	global position vector of a viscoelastically deformed vertex v_i at time t
s	tensile stiffness, elasticity coefficient, elastic spring constant
s_u	local scaling factor along **u**
s_v	local scaling factor along **v**
s_w	local scaling factor along **w**
$s(e)$	elasticity coefficient at strain e
$s(t)$	position on path at time t
s_0	initial position on path at time t_0
T	period of harmonic oscillation

$\mathbf{T}(t)$	global translation vector
Tip	end of VOO vector
t	time variable
t_0	start time
t_1	end time
$(t_{max} - t_0)$	duration of elastic restoration
	duration of stress relaxation process
	duration of strain creep process
$\mathbf{t}_i(t)$	total transformation vector of vertex v_i
U_e	elastic energy
U_{tot}	total energy
U_k	kinetic energy
$\mathbf{u}$	unit vector of a local frame of reference
$\mathbf{u}(t)$	direction of tangent on a path curve at time t
$\mathbf{v}$	unit vector of a local frame of reference
$\mathbf{V}$	a VOO vector
v_i	ith vertex
$\mathbf{w}$	unit vector of a local frame of reference
$x(t)$	magnitude of vector $\Delta\mathbf{V}(t)$
x_{max}	magnitude of vector $\Delta\mathbf{V}(t_0)$
$\dot{\mathbf{x}}(t)$	velocity of $\mathbf{x}(t)$
$\ddot{\mathbf{x}}(t)$	acceleration of $\mathbf{x}(t)$

List of Abbreviations

3D	three dimensional
CA	computer animation
CG	computer graphics
ICG	interactive computer graphics
LBE	location based entertainment

VOO	Vector Offset Operator
VOODOO	Vector Offset Operators for Deformable Organic Objects

List of References

Alias / Wavefront (1997). *Learning Alias v8*. Silicon Graphics Ltd.

Alexander, R.McN. (1962). Viscoelastic Properties of the Body Wall of Sea Anemones. *Journal of Experimental Biology*. 39, 373-386.

Algori, M.E. and Schmitt, F. (1996). Mesh Simplification. In: *EUROGRAPHICS Proceedings, Poitiers FR 1996*. Blackwell Publishers, 15, 77-86.

Anjyo, K., Usami, Y. and Kurihara, T. (1992). A Simple Method for Extracting the Natural Beauty of Hair. In: *COMPUTER GRAPHICS Proceedings, Chicago IL 1992*. ACM SIGGRAPH, 111-120.

Aono, M. (1990). A Wrinkle Propagation Model for Cloth. In: *Computer Graphics International Proceedings, Tokyo 1990*. Springer-Verlag, 96-115.

Badler, N.I., Barsky, B.A. and Zeltzer, D. (editors). (1991). *Making Them Move: Mechanics, Control and Animation of Articulated Figures*. San Mateo CA, Morgan Kaufmann.

Badler, N.I., Phillips, C.B. and Webber, B.L. (1993). *Simulating Humans, Computer Graphics Animation and Control*. New York NY, Oxford University Press.

Baraff, D. and Witkin, A. (1992). Dynamic Simulation of Non-Penetrating Flexible Bodies. In: *COMPUTER GRAPHICS Proceedings, Chicago IL 1992*. ACM SIGGRAPH, 303-308.

Barr, A.H. (1984). Global and Local Deformations of Solid Primitives. In: *COMPUTER GRAPHICS Proceedings, Minneapolis MN 1984*. ACM SIGGRAPH, 21-30.

Barzel, R. (1992). *Physically-Based Modelling for Computer Graphics: A Structured Approach*. San Diego CA, Academic Press.

Bechmann, D. and Dubreuil, N. (1992). Animation through Space and Time Based on a Space Deformation Model. In: *EUROGRAPHICS Proceedings 1992*, Blackwell Publishers.

Beier, T. and Neely, S. (1992). Feature-Based Image Metamorphosis. In: *COMPUTER GRAPHICS Proceedings, Chicago IL 1992*. ACM SIGGRAPH, 35-42.

Brewer, J.A. and Anderson, D.C. (1977). Visual Interaction with Overhauser Curves and Surfaces. *Computer Graphics*. 11(2), 132-137.

Bruderlin, A. and Williams, L. (1995). Motion Signal Processing. In: *COMPUTER GRAPHICS Proceedings, Los Angeles CA 1995*. ACM SIGGRAPH, 97-104.

Burtnyk, N. and Wein, M. (1976). Interactive Skeleton Techniques for Enhancing Motion Dynamics in Key Frame Animation. *Communications of the ACM*. 19, 10.

Burton, R. (1990). *Bird Flight*. London, Eddison Sadd.

Carignan, M., Yang, Y., Magnenat-Thalmann, N. and Thalmann, D. (1992). Dressing Animated Synthetic Actors with Complex Deformable Clothes. In: *COMPUTER GRAPHICS*

Proceedings, Chicago IL 1992. ACM SIGGRAPH, 99-104.

Chadwick, J.E., Haumann, D.R. and Parent, R.E. (1989). Layered Construction for Deformable Animated Characters. In: *COMPUTER GRAPHICS Proceedings, Boston MA 1989*. ACM SIGGRAPH, 243-252.

Chang, Y.K. and Rockwood, A.P. (1994). A Generalised de Casteljau Approach to 3D Free-Form Deformation. In: *COMPUTER GRAPHICS Proceedings, Orlando FL 1994*. ACM SIGGRAPH, 257-260.

Chazelle, B. (1991). Triangulating a Simple Polygon in Linear Time. *Discrete Computational Geometry*. 6, 485-524.

Chen, D. T. and Zeltzer, D. (1992). Pump It Up: Computer Animation of a Biomechanically Based Model of Muscle Using the Finite Element Method. In: *COMPUTER GRAPHICS Proceedings, Chicago IL 1992*. ACM SIGGRAPH, 89-98.

Christensen, R.M. (1982). *Theory of Viscoelasticity*. New York NY, Academic Press.

Cohen, M.F. (1992). Interactive Spacetime Control for Animation. In: *COMPUTER GRAPHICS Proceedings, Chicago IL 1992*. ACM SIGGRAPH, 293-302.

Comninos, P.P. (1980). CGAL: Computer Graphics and Animation Language. In: *EUROGRAPHICS Proceedings, Geneva '80*. Blackwell Publishers.

Comninos, P.P. (1986). CGAL: The Soft Machine. In: *Computer Graphics Proceedings, London 1986*. Online International.

Coquillart, S. (1990). Extended Free-Form Deformation: A Sculpting Tool for 3D Geometric Modelling. In: *COMPUTER GRAPHICS Proceedings, Dallas TX 1990*. ACM SIGGRAPH, 187-196.

Coquillart, S. and Jançene, P. (1991). Animated Free-Form Deformation: An Interactive Animation Technique. In: *COMPUTER GRAPHICS Proceedings, Las Vegas NV 1991*. ACM SIGGRAPH, 23-26.

Daldegan, A., Magnenat-Thalmann, N., Kurihara, T. and Thalmann, D. (1993). An Integrated System for Modelling, Animating and Rendering Hair. In: Thalmann, D. (editor). *Recent Techniques in Human Modelling, Animation and Rendering, COMPUTER GRAPHICS Course Notes, Anaheim CA 1993* . ACM SIGGRAPH, 80(3), 40-49.

Dias, J.M.S., Galli, R. (1995). Physically-Based Deformable Objects with Realistic Behaviour for 3D Virtual Film-Sets Backgrounds. In: *EU RACE Internal Report '95*. 2(1).

Dimery, N.J., Alexander, R.McN. and Deyst, K.A. (1985). Mechanics of the Ligamentum Nuchae of Some Artiodactyls. *Journal of Zoology*. A(206), 341-351.

Fournier, A. and Montuno, D.Y. (1984). Triangulating Simple Polygons and Equivalent Problems. *ACM Transactions on Graphics*. 3(2), 153-174.

Fournier, A. and Reeves, W.T. (1986). A Simple Model of Ocean Waves. In: *COMPUTER GRAPHICS Proceedings, Dallas TX 1986*. ACM SIGGRAPH, 75-83.

Foley, J.D., van Dam, A., Feiner, S.K. and Hughes, J.F. (1990). *Computer Graphics: Principles and Practice*. 2nd ed. Addison-Wesley.

Fung, Y.C. (1965). *Foundations of Solid Mechanics*. Englewood Cliffs NJ, Prentice Hall.

Fung, Y.C. (1981). *Biomechanics: Mechanical Properties of Living Tissues*. New York, Springer-Verlag.

Fung, Y.C. (1990). *Biomechanics: Motion, Flow, Stress and Growth*. New York, Springer-Verlag.

Gosline, J.M. (1971). Connective Tissue Mechanics of Metridium Senile. *Journal of Experimental Biology*. 55, 775-795.

Gourret, J.P, Magnenat-Thalmann, N. and Thalmann, D. (1989). Simulation of Object and Human Skin Deformations in a Grasping Task. In: *COMPUTER GRAPHICS Proceedings, Boston MA 1989*. ACM SIGGRAPH, 21-30.

Gray, J. (1968). *Animal Locomotion*. London, Weindenfeld and Nicholson.

Hilton, T.L. and Egbert, P.K. (1994). Vector Fields: An Interactive Tool for Animation, Modelling and Simulation with Physically Based 3D Particle Systems and Soft Objects. In: *EUROGRAPHICS Proceedings, Barcelona SP 1994*. Blackwell Publishers, 329-338.

Holton, M. and Alexander, S. (1995). Soft Cellular Modelling: A Technique for the Simulation of Non-rigid Materials. In: Ernshaw, R.A. and Vince, J.A. (editors). *Computer Graphics: Developments in Virtual Environments*. Academic Press, 449-460.

Hsu, W.M., Hughes, J.F. and Kaufman, H. (1992). Direct Manipulation of Free-Form Deformations. In: *COMPUTER GRAPHICS Proceedings, Chicago IL 1992*. ACM SIGGRAPH, 177-184.

Hurmusiadis, V. (1991). An Animation Model for Birds in Flight. In: *SCA, Bournemouth 1991*. N.C.C.A., Bournemouth University UK, 1-22.

Hurmusiadis, V. (1996). Vector Offset Operators for Dynamic Control of Deformable Objects. In: Jones, H. *et al.* (editors). *EUROGRAPHICS UK Chapter Proceedings, London 1996*. Imperial College, London, 2, 59-71.

Hurmusiadis, V. (1997). Fast Implementation of Viscoelastic Creep Suitable for Manipulation of Organic Objects. In: Santo, H. (editor). *EDUGRAPHICS & COMPUGRAPHICS Combined Proceedings, Vilamoura Portugal 1997*. GRASP, 241-249.

Impelluso, T. (1996). Physically-Based Virtual Reality in a Distributed Environment. *Computer Graphics*. 30(4), 60-61.

Kalra, P., Mangili, A., Magnenat-Thalmann, N., Thalmann, D. (1991). SMILE: A Multilayered Facial Animation System. In: *Modelling in Computer Graphics, Tokyo 1991*. Springer-Verlag.

Kass, M. and Miller, G. (1990). Rapid, Stable Fluid Dynamics for Computer Graphics. In: *COMPUTER GRAPHICS Proceedings, Dallas TX 1990*. ACM SIGGRAPH, 49-57.

Kent, J.R., Carlson, W.E. and Parent, R.E. (1992). Shape Transformation for Polyhedral Objects. In: *COMPUTER GRAPHICS Proceedings, Chicago IL 1992*. ACM SIGGRAPH, 47-54.

Ker, R.F., Dimery, N.J. and Alexander, R.McN. (1986). The Role of Tendon Elasticity in

Hopping in a Wallaby (Macropus Rufogriseus). *Journal of Zoology*. A(208), 417-428.

Koch, R.M., Gross, M.H., Carls, F.R., vonBuren, D.F., Frankhauser, G. and Parish, Y.I.H. (1996). Simulating Facial Surgery Using Finite Element Models. In: *COMPUTER GRAPHICS Proceedings, New Orleans LA 1996*. ACM SIGGRAPH, 421-428.

Koehl, M.A.R. (1977). Mechanical Diversity of Connective Tissue of the Body Wall of Sea Anemones. *Journal of Experimental Biology*. 69, 107-125.

Kunii, T.L. and Gotoda, H. (1990). Modelling and Animation of Garment Wrinkle Formation Processes. In: *Computer Animation Proceedings, Tokyo 1990*. Springer-Verlag, 131-147.

Lafleur, B., Magnenat-Thalmann, N. and Thalmann, D. (1991). Cloth Animation with Self-Collision Detection. In: Kunii, T.L. (editor). *Modelling in Computer Graphics Proceedings, Tokyo 1991*. Springer-Verlag.

Lee, Y., Terzopoulos, D. and Waters, K. (1995). Realistic Modelling for Facial Animation. In: *COMPUTER GRAPHICS Proceedings, Los Angeles CA 1995*. ACM SIGGRAPH, 55-62.

Lerios, A., Garfinkle, C.D. and Levoy, M. (1995). Feature-Based Volume Metamorphosis. In: *COMPUTER GRAPHICS Proceedings, Los Angeles CA 1995*. ACM SIGGRAPH, 449-456.

Lianis, M.G. (1981). *Mechanics of Deformable Materials: Elastic Behaviour*. Thessaloniki, Greece, Aristotle University Press.

Magnenat-Thalmann, N. and Thalmann, D. (1987). The Direction of Synthetic Actors in the Film "Rendez Vous à Montréal". *IEEE Computer Graphics and Applications*. 7(12), 9-19.

Magnenat-Thalmann, N. and Thalmann, D. (1993). The World of Virtual Actors. In: Magnenat-Thalmann, N. and Thalmann, D. (editors). *Virtual Worlds and Multimedia*. John Wiley and Sons.

Metaxas, D. and Terzopoulos, D. (1992). Dynamic Deformation of Solid Primitives with Constraints. In: *COMPUTER GRAPHICS Proceedings, Chicago IL 1992*. ACM SIGGRAPH, 309-312.

Mendelson, A. (1968). *Plasticity: Theory and Application*. New York, Macmillan Co.

Miller, G. (1988). The Motion Dynamics of Snakes and Worms. In: *COMPUTER GRAPHICS Proceedings, Boston MA 1989*. ACM SIGGRAPH, 169-178.

Miller, G. and Pearce, A. (1989). Globular Dynamics: A Connected Particle System for Animating Viscous Fluids. In: Barr, A. *et al.* (editors). *Topics in Physically-Based Modelling, COMPUTER GRAPHICS Course Notes, Boston MA 1989*. 30, ACM SIGGRAPH.

Miyazaki, S., Ishiguro, M., Yasuda, T., Yokoi, S. and Toriwaki, J. (1995). A Study of Virtual Manipulation of Elastic Objects. In: Ernshaw, R.A. and Vince, J.A. (editors). *Computer Graphics: Developments in Virtual Environments*. Academic Press, 381-391.

Mortenson, M.E. (1985). *Geometric Modeling*. J. Wiley and Sons.

Muybridge, E. (1899). *Animals in Locomotion*. London.

Mylonas, K.M. (1978). *Mechanics I*. Athens, Greece, National Metsovio University Press.

Parent, R.E. (1977). A System for Sculpting 3D Data. *Computer Graphics*. 11(2), 138-147.

Peachey, D.R. (1986). Modelling Waves and Surf. In: *COMPUTER GRAPHICS Proceedings, Dallas TX 1986*. ACM SIGGRAPH, 65-74.

Pentland, A. and Williams, J.(1989). Good Vibrations: Modal Dynamics for Graphics and Animation. In: *COMPUTER GRAPHICS Proceedings, Boston MA 1989*. ACM SIGGRAPH, 215-222.

Platt, J.C. and Barr, A.H. (1988). Constraint Methods for Flexible Models. In: *COMPUTER GRAPHICS Proceedings, Atlanta GA 1988*. ACM SIGGRAPH, 279-288.

Press, W.H., Flannery, B.P., Teukolsky, S.A. and Vetterling, W.T. (1986). *Numerical Recipes: The Art of Scientific Computing*. Cambridge UK, Cambridge University Press.

Ronfard, R.P. and Rossignac, J. (1994). Triangulating Multiply-Connected Polygons: A Simple, Yet Efficient Algorithm. In: *EUROGRAPHICS Proceedings, Barcelona SP 1994*. Blackwell Publishers, 13(3), 281-292.

Rossignac, J. and Kaul, A. (1994). AGRELs and BIPs: Metamorphosis as a Bezier Curve in the Space of Polyhedra. In: *EUROGRAPHICS Proceedings, Barcelona SP 1994*. Blackwell Publishers, 13(3), 179-184.

Sederberg, T.W. and Parry, S.R. (1986). Free-Form Deformation of Solid Geometric Models. In: *COMPUTER GRAPHICS Proceedings, Dallas TX 1986*. ACM SIGGRAPH, 151-160.

Slijper, E.J. (1961). Locomotion and Locomotory Organs in Whales and Dolphins (Cetacea). In: *Symposium - Zoological Society London*. 5, 77-94.

Snibbe, S.S. (1995). A Direct Manipulation Interface for 3D Computer Animation. In: *EUROGRAPHICS Proceedings 1995*. Blackwell Publishers, 271-283.

Softimage 3D (1997). *Reference Guide*. Version 3.5, Softimage Inc., Microsoft Corporation.

Studio 3D Max (1996). *Tutorials*. Autodesk, Inc. Publication.

Szeliski, R. and Tonnesen, D. (1992). Surface Modelling with Oriented Particle Systems. In: *COMPUTER GRAPHICS Proceedings, Chicago IL 1992*. ACM SIGGRAPH, 185-194.

Terzopoulos, D., Platt, J., Barr, A. and Fleischer, K. (1987). Elastically Deformable Models. In: *COMPUTER GRAPHICS Proceedings, Anaheim CA 1987*. ACM SIGGRAPH, 205-214.

Terzopoulos, D. and Fleischer, K. (1988). Modelling Inelastic Deformation: Viscoelasticity, Plasticity, Fracture. In: *COMPUTER GRAPHICS Proceedings, Atlanta GA 1988*. ACM SIGGRAPH, 269-278.

Terzopoulos, D. and Waters, K. (1990). Physically-Based Facial Modelling, Analysis and Animation. *Journal of Visualisation and Computer Animation*. 1(2), 73-80.

Thomas-Ngo, J. and Marks, J. (1993). Spacetime Constraints Revisited. In: *COMPUTER GRAPHICS Proceedings, Anaheim CA 1993*. ACM SIGGRAPH, 343-350.

Tonnesen, D. (1991). Modelling Liquids and Solids Using Thermal Particles. In: *Graphics Interface Proceedings, Calgary, Canada 1991*. 255-262.

Ts'o, P.Y. and Barsky, B.A. (1987). Modelling and Rendering Waves: Wave-Tracing Using Beta-splines and Reflective and Refractive Texture Mapping. *ACM Transactions on Graphics*. 6(3), 191-213.

Tu, X. and Terzopoulos, D. (1994). Artificial Fishes: Physics, Locomotion, Perception, Behaviour. In: *COMPUTER GRAPHICS Proceedings, Orlando FL 1994*. ACM SIGGRAPH, 43-50.

Turk, G. (1992). Re-Tiling Polygonal Surfaces. In: *COMPUTER GRAPHICS Proceedings, Chicago IL 1992*. ACM SIGGRAPH, 55-64.

Turner, R. and Thalmann, D. (1993). The Elastic Surface Layer Model for Animated Character Construction. In: *Computer Graphics International Proceedings, Tokyo 1993*. Springer-Verlag.

Turner, R. (1995). LEMAN: A System for Constructing and Animating Layered Elastic Characters. In: Ernshaw, R.A. and Vince, J.A. (editors). *Computer Graphics: Developments in Virtual Environments*. Academic Press.

Vince, J.A. (1992). *3D Computer Animation*. Addison-Wesley.

Vince, J.A. (1995). *Virtual Reality Systems*. Addison-Wesley.

Wang, S.W. and Kaufman, A. (1995). Volume Sculpting. In: *Symposium on Interactive 3D Graphics Proceedings, New York 1995*. ACM Press, 151-156.

Ward-Smith, A.J. (1984). *Biophysical Aerodynamics and the Natural Environment*. London, John Wiley & Sons.

Waters, K. (1987). A Muscle Model for Animating Three-Dimensional Facial Expression. In: *COMPUTER GRAPHICS Proceedings, Anaheim CA 1987*. ACM SIGGRAPH, 17-24.

Waters, K. (1988). *The Computer Synthesis of Expressive Three-Dimensional Facial Character Animation*. Ph.D. Thesis, Middlesex Polytechnic.

Watt, A. and Watt, M. (1992). *Advanced Animation and Rendering Techniques*. Addison-Wesley.

Werneck, J. (1994). *The Inventor Mentor: Programming Object-Oriented 3D Graphics with Open Inventor Release 2, Open Inventor Architecture Group*. Addison-Wesley.

Witkin, A. and Kass, M. (1988). Spacetime Constraints. In: *COMPUTER GRAPHICS Proceedings, Atlanta GA 1988*. ACM SIGGRAPH, 159-168.

Witkin, A. and Popoviç, Z. (1995). Motion Warping. In: *COMPUTER GRAPHICS Proceedings, Los Angeles CA 1995*. ACM SIGGRAPH, 105-108.

Witkin, A. and Welch, W. (1990). Fast Animation and Control of Nonrigid Structures. In: *COMPUTER GRAPHICS Proceedings, Dallas TX 1990*. ACM SIGGRAPH, 243-250.

Wu, Y., Thalmann, D. and Magnenat-Thalmann, N. (1995). Deformable Surfaces Using Physically-Based Particle Systems. In: Ernshaw, R.A. and Vince, J.A. (editors). *Computer Graphics: Developments in Virtual Environments*. Academic Press, 205-215.

Yamada, H. (1970). *Strength of Biological Materials*. Baltimore, Williams and Wilkins.

Yang, Y., Magnenat-Thalmann, N. and Thalmann, D. (1992). A New Design Tool for the Garment Industry. In: *Computers in Industry*. 19, Elsevier, 185-191.

Zeltzer, D. (1988). Towards an Integrated View of 3D Computer Animation. *The Visual Computer*.

Zienkiewicz, O.C. (1977). *The Finite Element Method*. London, McGraw-Hill.

Bibliography

Adler, H.E. (1975). *Fish Behaviour: Why fishes do what they do.* TFH Publications.

Alexander, R.M. (1982). *Locomotion of Animals.* Blackie and Son.

Alexander, R.M. (1983). *Animal Mechanics.* Cambridge University Press.

Alexander, R.M. (1988). *Elastic Mechanisms in Animal Movement.* Cambridge University Press.

Ames, A.L., Nadeau, D. and Moreland, J. (1996). *The VRML Sourcebook.* John Wiley and Sons.

Arvo, J. (editor) (1991). *Graphics Gems.* vol II, Academic Press.

Aymar, G. (1936). *Bird Flight.* Bodley Head.

Badler, N.I., Barsky, B.A. and Zeltzer, D. (1991). *Making Them Move: Mechanics, Control and Animation of Articulated Figures.* Morgan Kaufmann.

Badler, N.I., Phillips, C.B. and Webber, B.L. (1993). *Simulating Humans, Computer Graphics Animation and Control.* Oxford University Press.

Barzel, R. (1992). *Physically-Based Modelling for Computer Graphics: A Structured Approach.* Academic Press.

Begault, D.R. (1995). *3D Sound For Virtual Reality and Multimedia.* Academic Press.

Blinn, J. (1996). *Jim Blinn's Corner: A Trip Down the Graphics Pipeline.* Morgan Kaufmann Publications.

Borelli, G.A. (1680). *De Motu Animalium Ex Principio Mechanico Statico.* Rome.

Brown, J.R., Earnshaw, R.E., Jern, M. and Vince, J.A. (editors). (1995). *Visualization: Using Computer Graphics to Explore Data and Present Information.* John Wiley and Sons.

Chen, M., Townsend, P. and Vince, J.A. (editors). (1996). *HIGH Performance Computing for Computer Graphics and Visualisation.* Springer-Verlag.

Childress, S. (1981). *Mechanics of Swimming and Flying.* Cambridge University Press.

Christensen, R.M. (1982). *Theory of Viscoelasticity.* Academic Press.

Earnshaw, R.A. (1993). *Virtual Reality Systems.* Academic Press.

Earnshaw, R.A., Jones, H. and Vince, J.A. (editors). (1996). *Digital Media & Electronic Publishing.* Academic Press.

Earnshaw, R.A., Vince, J.A. and Jones, H. (editors). (1995). *Virtual Reality Applications.* Academic Press.

Earnshaw, R.A and Vince, J.A. (editors). (1995). *Computer Graphics: Developments in Virtual Environments.* Academic Press.

Earnshaw, R.A. and Vince, J.A. (editors). (1995). *Multimedia Systems and Applications.* Academic Press.

Ebert, D.S., Musgrave, F.K., Peachey, D., Perlin, K. and Worley, S. (1995). *Texturing and Modeling.* Academic Press.

Eldred, E. (1960). *Posture and Locomotion.* Magoum and Hall.

Foley, J.D., vanDam, A., Feiner, S.K. and Hughes, J.F. (1991). *Computer Graphics: Principles and Practice.* 2nd ed., Addison-Wesley.

Foley, J.D., vanDam, A., Feiner, S.K., Hughes, J.F. and Phillips, R.L. (1994). *Introduction to Computer Graphics.* Addison-Wesley.

Fowler, S. and Stanwick, V. (1995). *The GUI Style Guide.* Academic Press.

Fradford, R.E. (1995). *Real Time Animation Toolkit In C++.* John Wiley and Sons.

Fung, Y.C. (1965). *Foundations of Solid Mechanics.* Prentice-Hall.

Fung, Y.C. (1981). *Biomechanics: Mechanical Properties of Living Tissues.* Springer-Verlag.

Fung, Y.C. (1990). *Biomechanics: Motion, Flow, Stress and Growth.* Springer-Verlag.

Glaeser, G. (1994). *Fast Algorithms For 3D Graphics.* Springer-Verlag.

Glassner, A.S. (1989). *An Introduction to Ray-Tracing.* Academic Press.

Glassner, A.S. (editor) (1990). *Graphics Gems.* vol I, Academic Press.

Goldstein, H. (1950). *Classical Mechanics.* Addison-Wesley.

Gray, J. (1968). *Animal Locomotion.* Weidenfeld and Nicholson.

Greenewalt, C.H. (1961). *Hummingbirds.* Doubleday.

Hertel, H. (1926). *Structure, Form, Movement.* Rheinhold Publishing.

Hill, A.V. (1927). *Muscular Movement in Man.* McGraw-Hill.

Hook, B. (1995). *Building A 3D Game Engine In C++.* John Wiley and Sons.

Inman, V.T., Ralston, H.J. and Todd, F. (1981). *Human Walking.* Williams and Wilkins.

Jameson, W. (1958). *The Wandering Albatros.* Hart-Davis.

Kardestuncer, H. and Norrie, D.H. (1987). *Finite Element Handbook.* McGraw-Hill.

Kirk, D. (editor) (1992). *Graphics Gems.* vol III, Academic Press.

Kunii, T.L. (editor). (1991). *Modelling in Computer Graphics.* Springer-Verlag.

Landau, L.D. and Lifshitz, E.M. (1959). *Theory of Elasticity.* Pergamon Press.

Lapidus, L. and Pinder, G.F. (1982). *Numerical Solution of Partial Differential Equations in Science and Engineering.* John Wiley and Sons.

Lianis, M.G. (1981). *Mechanics of Deformable Materials: Elastic Behaviour.* Thessaloniki, Greece, Aristotle University Press.

Magnenat-Thalmann, N. and Thalmann, D. (editors). (1991). *New Trends in Animation and Visualization.* John Wiley and Sons.

Magnenat-Thalmann, N. and Thalmann, D. (editors). (1992). *Creating and Animating the Virtual World.* Springer-Verlag.

Magnenat-Thalmann, N. and Thalmann, D. (editors). (1993). *Virtual Worlds and Multimedia.* John Wiley and Sons.

McFarland, D. (1985). *Animal Behaviour.* Pitman.

Mendelson, A. (1968). *Plasticity: Theory and Application.* Macmillan.

Mortenson, M.E. (1985). *Geometric Modeling.* John Wiley and Sons.

Muvdi, B.B. and McNabb, J.W. (1991). *Engineering Mechanics of Materials.* 3rd ed., Springer-Verlag.

Muybridge, E. (1899). *Animals in Locomotion.* London.

Mylonas, K.M. (1978). *Mechanics I.* Athens, Greece, National Metsovio University Press.

Paeth, A.W. (editor) (1995). *Graphics Gems.* vol V, Academic Press.

Pedley, T.J. (1977). *Scale Effects in Animal Locomotion.* Academic Press.

Pennycuik, C.J. (1972). *Animal Flight.* Edward Arnold.

Pennycuik, C.J. (1989). *Bird Flight Performance.* Oxford University Press.

Pesce, M. (1995). *VRML: Browsing & Building Cyberspace.* New Riders Publishing.

Preparata, F.P. and Shamos, N.I. (1985). *Computational Geometry: An Introduction.* Springer-Verlag.

Press, W., Teukolsky, S.A., Vettering, W.T. and Flannery, B.P. (1992). *Numerical Recipes in C.* Cambridge University Press.

Rainey, R.C. (1976). *Insect Flight.* Blackwell Scientific Publications.

Sih, G.C. (1981). *Mechanics of Fracture.* Martinus Nijhoff.

Steindler, A. (1955). *Kinesiology of the Human Body.* Springfield, Illinois, C.C. Thomas Publisher.

Thalmann, D. (1990). *Scientific Visualization and Graphics Simulation.* John Wiley and Sons.

Thompson, D.W. (1952). *On Growth and Form.* 2nd ed., Cambridge.

Vacca, J. (1996). *VRML: Bring Virtual Reality to the Internet.* Academic Press.

Vince, J.A. (1992). *3D Computer Animation.* Addison-Wesley.

Vince, J.A. (1995). *Virtual Reality Systems.* Addison-Wesley.

Ward-Smith, A.J. (1984). *Biophysical Aerodynamics and the Natural Environment.* London, John Wiley & Sons.

Watt, A.(1993). *3D Computer Graphics.* Addison-Wesley.

Watt, A. and Watt M. (1992). *Advanced Animation and Rendering Techniques.* Addison-Wesley.

Wells, D.A. (1967). *Theory and Problems of Lagrangian Dynamics.* McGraw-Hill.

Werneck, J. (1994). *The Inventor Mentor: Programming Object-Oriented 3D Graphics with Open Inventor Release 2, Open Inventor Architecture Group.* Addison-Wesley.

Wexelblat, H. (1995). *Virtual Reality Applications and Explorations.* Academic Press.

White, R.E. (1985). *Introduction to the Finite Element Method with Applications to Non-Linear Problems.* John Wiley and Sons.

Wittenburg, T. (1995). *Photo-Based 3D Graphics In C++.* John Wiley and Sons.

Wolf, R, and Yaeger, L. (1993). *Visualization Of Natural Phenomena.* Spinger-Verlag.

Wu, T-Y.T., Brokaw, C.J. and Bremen, C. (1975). *Swimming and Fluing in Nature.* Plenum Press.

Yamada, H. (1970). *Strength of Biological Materials.* Williams and Wilkins.

Young, J. (1975). *The Life of Mammals.* Oxford University Press.

Zienkiewicz, O.C. (1977). *The Finite Element Method.* 3rd ed., McGraw-Hill.