Hairstyle Modelling Based on A Single Image

WENSHU ZHANG

A thesis submitted in partial fulfilment of the requirements
of Bournemouth University for the degree of
Doctor of Philosophy

23rd April, 2018

Copyright statement

Abstract

Hair is an important feature to form character appearance in both film and video game industry. Hair grooming and combing for virtual characters was traditionally an exclusive task for professional designers because of its requirements for both technical manipulation and artistic inspiration. However, this manual process is time-consuming and further limits the flexibility of customised hairstyle modelling. In addition, it is hard to manipulate virtual hairstyle due to intrinsic hair shape. The fast development of related industrial applications demand an intuitive tool for efficiently creating realistic hairstyle for non-professional users. Recently, image-based hair modelling has been investigated for generating realistic hairstyle.

This thesis demonstrates a framework *Struct2Hair* that robustly captures a hairstyle from a single portrait input. Specifically, the 2D hair strands are traced from the input with the help of image processing enhancement first. Then the 2D hair sketch of a hairstyle on a coarse level is extracted from generated 2D hair strands by clustering. To solve the inherently ill-posed single-view reconstruction problem, a critical hair shape database has been built by analysing an existing hairstyle model database. The critical hair shapes is a group of hair strands which possess similar shape appearance and close space location. Once the prior shape knowledge is prepared, the hair shape descriptor (HSD) is introduced to encode the structure of the target hairstyle. The HSD is constructed by retrieving and matching corresponding critical hair shape centres in the database. The full-head hairstyle is reconstructed by uniformly diffusing the hair strands on the scalp surface under the guidance of extracted HSD. The produced results are evaluated and compared with the state-of-the-art image based hair modelling methods. The findings of this thesis lead to some promising applications such as blending hairstyles to populate novel hair model, editing hairstyle (adding fringe hair, curling and cutting/extending hairstyle) and a case study of Bas-relief hair modelling on pre-processed hair images.

Acronyms

2D	two-dimensional
3D	three-dimensional
CHS	Critical Hair Shape
HSD	Hair Shape Descriptor
CG	Computer Graphics

Contents

List of Figures

List of Tables

List of Algorithms

Acknowledgements

This thesis becomes a reality with the kind support and help of many people. I would like to extend my sincere thanks to all of them.

Foremost, I want to express the gratitude to my supervisors Prof. Jian Chang and Prof. Jian Jun Zhang for the continuous support of my PhD research, for their patience, motivation and immense knowledge. Their guidance helped me in all the time of research and writing of this thesis.

Part of the inspirations leading to this work also arise from discussions with Dr. Shihui Guo and Yinyu Nie. Their critical comments are always there when I need them most.

My sincere thanks also goes to my labmates Dr. Min Jiang, Dr. Mathieu Sanchez, Dr. Shujie Deng, Dr. Kun Qian, Yunfei Fu and Dr. Shuang Liu in Tolpuddle Annex 2, for the sleepless nights we were working together before deadlines, and for all the fun we have had in the last five years. I also thank Dr. Xiaosong Yang, Prof. Lihua You for sharing the interesting news during the lunch break.

Besides my research, I would like to thank Jan Lewis and Sunny Choi for their generous support, especially for helping me get through the difficult time during the maternity leave. Gratitudes also go to Chinese Scholarship Council for PhD studentship and AniNex exchanging project for visiting Zhejiang University. This thesis cannot be completed without their financial supports.

Finally, I would like to thank my family, including my parents, my parents-in-law, my husband and our daughter, for their unconditional love and support.

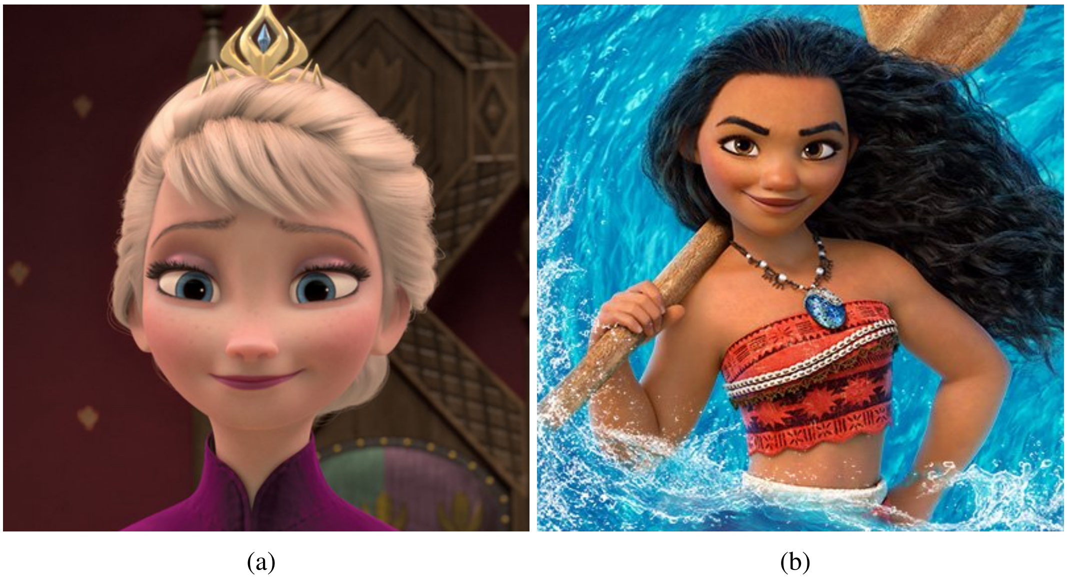
Declaration

This thesis has been created by myself and has not been submitted in any previous application for any degree. The work in this thesis has been undertaken by myself except where otherwise stated.

Chapter 1

Introduction

1.1 Motivation



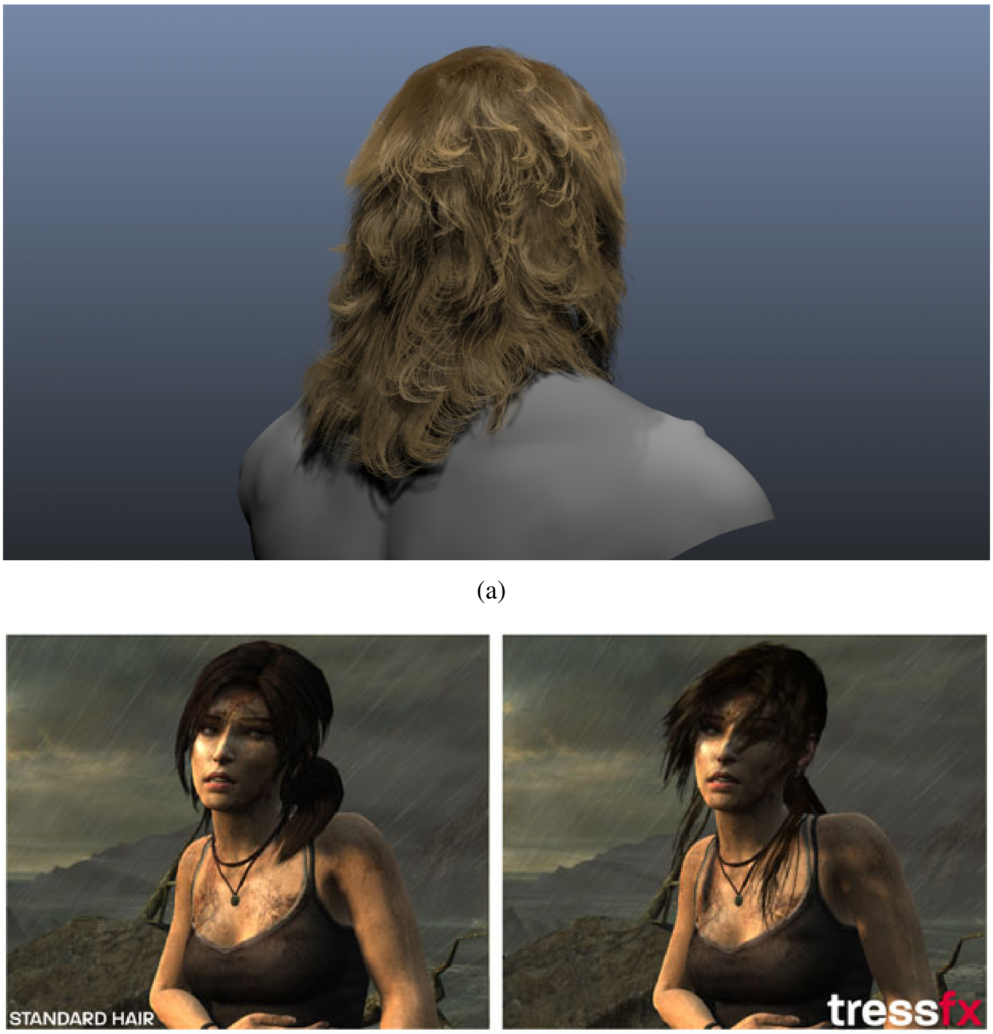
(a) (b)

Figure 1.1: *Hair in recent released animation films*[1]. *1.1(a) is Elsa in* ***Frozen****, year 2013. 1.1(b) is Moana in* ***Moana****, year 2016.*

In computer animation, when modelling a virtual character, hairstyling is very vital as an essential part for showing a character's personality. As computer graphics hardware and software dramatically improved in today's world, virtual hair modelling is wildly used in film production, video games and other various CG applications. Recently released animation films such as *Frozen* and *Moana* (see Figure 1.1) benefit from delicate virtual hairstyles to demonstrate the characters. In video games, to solve the problem of realism, two world leading companies of computer graphics cards -

[1] Images credit: IMDB, Frozen and Moana.

Nvidia & AMD - have both developed their own hair technologies, HairWorks & TressFX respectively (see Figure 1.2). They make great efforts to avoid the plastic-molded like hairs appearing in video games. Furthermore, in interactive media and video games, perhaps the most popular facet of any character-customization tool is the ability to customize the appearance of a character in terms of hairstyling, dress and other forms of body modification (Sloan, 2015). All of these reasons make hair modelling recognized as a widespread topic in the computer graphics community.



(a)

(b)

Figure 1.2: *Hair techniques comparison: Nvidia vs AMD. 1.2(a) is the hair rendered by Nvidia hair technique HairWorks. 1.2(b) is the AMD hair technique TressFX used in video game Tomb Raider, the left picture and the right picture are the actress hair with and without TressFX.*

In Computer Graphics and Human Simulation, the hair modelling problem is divided into three sub-questions: styling, simulation and rendering (Ward *et al.*, 2007), each of them focuses on different aspects for better presenting a virtual character. The hair styling covers creating the model to demonstrate the appearance of a hairstyle.

Based on the built virtual model, researchers continue to work on hair physics, including the implementation of gravity, elasticity and collision. This process aims to generate high quality, plausible hair simulation. To further introduce realistic virtual hair closely matched to our daily observation, the presence of lighting, shadowing and coloring should be carefully investigated, which is in the scope of hair rendering.

In this thesis, the author focuses on approaches to hair modelling which automatically calculate the geometric shape of the hairstyle for generating high fidelity results. Specifically, I find a method that is able to analyse the basic structure of a hairstyle and reconstruct the full model from it. Ideally, this method should improve the efficiency of game designers and animators to style and comb virtual hair. Besides, it should contribute to the modelling software as complementary to the current hair system framework.

1.2 Background

1.2.1 Hair

Morphologically, there are two distinct parts of a human hair fibre, the follicle and the hair shaft. The hair fibre grows in three stages to form the visible hair shape (Robbins and Robbins, 2002). The growing stage takes place in the hair follicle with intensive production of keratin. When this structural protein is approaching the skin of the scalp and continuing to grow upward, the metabolic activity slows down and steps into a transition stage. After the growth has completely stopped, the keratin dehydrates and forms the hair shaft, which is the resting stage. The formation of the hair strand is the hair shaft extrusion from the associated follicle. The entire strand is shaped by the pressure on the hair shaft of a continuous sheath.

Additionally, hair shapes are different among various human ethnicities (Vernall, 1961). Generally, Asian people have straight hair with a circular cross section which is smooth and thick; African people have strongly curly hair with a flattened elliptical shape in cross section which is frizzy and irregular; Caucasian hair appears between these two extremes (see Figure 1.3). This rough categorisation does not take account of hair shapes which possess past or recent mixed origins.

Real-world hair itself is very complicated. There are almost 100,000 to 150,000 individual hairs growing on a human head. Not only the individual hair shape varies in different ethnicities, but also the distribution of the hair strands affect their collective

[2] SEM image credit: Xue Wei. http://www2.optics.rochester.edu/workgroups/cml/opt307/spr06/xue/project.htm.

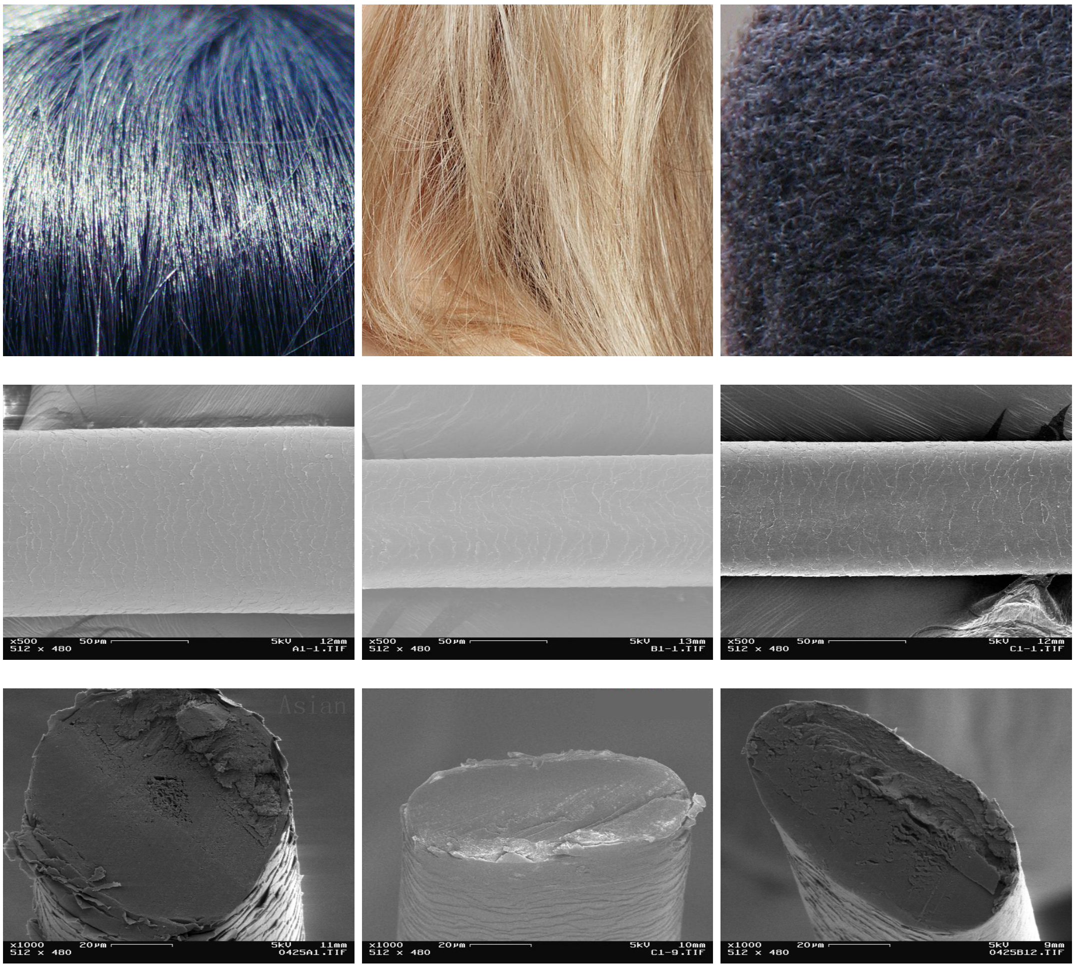


Figure 1.3: *Hair from different ethnic groups*[2]. *From left to right, are the hair properties of Asian people, Caucasian people and African people respectively. From up to bottom, are the color photographs, SEM images and cross-sections of various hair.*

behaviours. The tilted cuticle scales around the hair shaft to form an irregular surface, which cause anisotropic friction forces to make hair aggregation. This is the reason that curly hair is more likely to form clumps than straight hair (Bertails *et al.*, 2006). Thus, the shapes of the individual hairs and the aggregation of them result in the difficulties of modelling realistic virtual hairstyles.

1.2.2 Hairstyle

The hairstyle, hairdo or haircut refers to the styling of hair, usually on the human scalp. The fashioning of hair can be considered an aspect of personal grooming, fashion and cosmetics, although practical, cultural and popular considerations also influence some hairstyles. Cheang (2008) has conducted a research work with detailed investigating the connections among hairstyle, culture and fashion. In addition, the

British Broadcasting Corporation (BBC) has produced a video to discuss people's hairstyles changing involved in time shifting[3].

When discussing about the hairstyles for virtual characters, there is few research mentioned about the definition of the digital hairstyle. This thesis is an extension research of previous hair modelling techniques, which pays an extra attention on intrinsic structure of a hairstyle. According to the existing interactive hairstyling systems (Choe and Ko, 2005; Fu *et al.*, 2007), a virtual hairstyle consists of master hair strands/curves to demonstrate the basic appearance of the hairstyle. To add hair volume, more strands will be populated around the master strands to form different hair wisps. The distribution of those wisps further represents the whole shape of a hairstyle. The master strands' representations and the corresponding wisps arrangements make hairstyles are different from each other. Therefore, if the master strands - we assume it is the basic structure of a hairstyle - can be extracted from the real-world data like images, it will improve many hair modelling applications.

1.2.3 Challenge

For human beings, the most familiar things to us in this world are ourselves - the voices made by us, the appearances of what we look like and the behaviours of how we act. Increasing numbers of lifelike replicas have been made by animators and robotic engineers during the last decades. But the computer rendered human characters will trigger an uneasy feeling when they are so close to us and yet so far. This experience of familiar but eerie is identified as the uncanny in the field of robotics and computer animation. To map out the *uncanny valley*[4], success will require matching the level of realism of all aspects of the characters, as said by Karl MacDorman. With the well studied face and body capturing and modelling, unrealistic hair attached to those highly human-like avatars makes audiences drop into the uncanny valley and feel creepy. Hence, it is a challenge to model natural shapes of hairstyle to bridge the valley between the living human beings and the virtual character world.

Most early hair modelling research focused on simulation with a simple hairstyle due to the lack of processing power. With the fast development of computer graphical devices, the subject of precisely creating hair models has been an ongoing research topic since the 1980s. Hair combing and grooming for virtual characters was traditionally an exclusive task for professional designers because of its high

[3] BBC video: Bouffants, Beehives and Bobs: The Hairstyles That Shaped Britain

[4] In aesthetics, the uncanny valley is the hypothesis that human replicas which appear almost, but not exactly, like real human beings elicit feelings of eeriness and revulsion (or uncanniness) among some observers (MacDorman and Ishiguro, 2006). From Wikipedia

requirements for both technical manipulation and artistic inspiration. However, this manual process is time-consuming, which further limits the flexibility of customised hairstyle modelling. In addition, it is hard to manipulate virtual hairstyles due to the intrinsic hair shape. Therefore, an intuitive tool with the capability of efficiently creating realistic hairstyles for non-professional users is highly demanded by the fast developing computer animation industry.

To make an easy hair modelling tool accessible to a very wide range of users, image-based modelling is the growing technique in CG community. Image-based modelling methods use 2D image measurements to recover 3D object information. They calculate 3D data by using image cues like the shape from shading (Prados and Faugeras, 2006), shape from texture (Aloimonos, 1988), shape from contour (Witkin, 1980) and shape from defocus (Favaro and Soatto, 2005) etc. Such image-based modelling methods usually obtain 3D geometry information from multiple views, although other information is necessary to be acquired from each single image. Most of the reconstructed geometries from them are impressive and accurate. As images provide intuitive observations of the hairstyle, many approaches have been developed to capture hairstyles from real-world data. These methods can reconstruct dense and detailed 3D hair geometry with little difference compared with reference images. The image-based hair modelling algorithms use multi-viewed images to recover the missing depth information (Kong and Nakajima, 1998; Grabli *et al.*, 2002; Paris *et al.*, 2004; Wei *et al.*, 2005; Paris *et al.*, 2008; Jakob *et al.*, 2009). They mainly rely on complicated set-up of the capturing devices, which is difficult to be implemented by common users.

Recently, the research direction of image-based hair modelling has been changed to reconstruction from a single image (Chai *et al.*, 2012, 2013, 2015). This approach does not require the setup of a specialised system and allows almost an arbitrary portrait image as input. It also reduces the computational cost by eliminating the need for image registration. But as there is no sufficient depth information to infer the structure of 3D object from one still image, the single-viewed reconstruction is an inherently ill-posed problem. An abundant researches have been proposed to realize the reconstruction by additional assumptions, including geometric/learned shape priors or user input (Shen *et al.*, 2012; Shao *et al.*, 2012; Chen *et al.*, 2014; Remil *et al.*, 2017).

In hair modelling research, it is hard to establish the assumptions of geometric priors due to the irregular shapes of hair. In order to improve this situation, the latest hair modelling researches Hu *et al.* (2015) and Chai *et al.* (2016) have built hairstyle databases as their assumptions of shape priors to overcome the ill-posedness. Both of them retrieved and combined the candidate hairstyle models from the database to construct a similar hairstyle model compared to reference image. The distinguish difference between them is that Hu *et al.* (2015) requires user interaction but Chai *et al.* (2016) is fully automatic. An optimization process is needed afterwards to reshape the model to match the original input. This type of top-down approach witness the success of a data-driven technique. But as little attention has been paid to the local hair shape, it has a drawback of loosing details.

To sum up, the main challenge of hair shape modelling is to find a connection between real-world data and naturalness hair model, which needs less input and can be easily bringing into daily life for a wild range of users. This challenge leads to the main research question of this thesis discussed in the next section.

1.3 Research questions

According to the aforementioned challenge, this thesis raises a main research question that is can the structure of a hairstyle be analysed to generate a full-head hairstyle model? To prove the hypothesis of the basic structure of a hairstyle, there needs a solution to decompose the limit information of the image. This generates another question, what is the description of a hairstyle. To the best knowledge of the author, there is no quantitatively analysis of the basic structure of hairstyles. Therefore, this thesis assumes that basic elements can be extracted to fill the gap between reference image and target 3D hairstyle model.

To design a bottom-up pipeline and deliver a practical framework of modelling realistic hairstyle in a structure-aware manner, the following questions have to be carefully studied:

- How to extract the hair information from the single 2D image? There is inevitable noise existing in the image due to unconstrained illumination condition of the arbitrary portrait, which impairs the quality of feature extraction. How to extract useful information to prepare the 2D geometrical shape of hair?

- How to model the hair structure? Once the 2D hair strands have been captured from the image, how to model their structure? Is it capable to retain the local hair features as well as the global shape of the appearance?

- How to reconstruct the 3D hair structure? The most challenging question in this research is how to recover the missing depth? This step bridges the gap between the 2D hair strokes and 3D hair model. Will data-driven method contribute to simplify the reconstruction problem?

- How to diffuse the hair on the scalp? Once the 3D hair structure has been built, how to interpolate the complete hairs around the scalp is the last but an important question. The designed approach requires hair shapes diffusing smoothly under the guidance of the 3D structure.

The following hypotheses are proposed to address each of theses questions. Firstly, image processing methods should be a feasible solution of 2D hair shape acquisition. Secondly, a hair shape descriptor needs to be designed for encoding the representation of hair structure. Thirdly, a data-driven technique would benefit constructing the 3D hair structure from 2D hair geometry. Last, a 3D hair orientation field should be developed to diffuse the hair strands to obtain the complete hairstyle model.

1.4 Aim and objectives

Inspired by those successful research outcomes mentioned in section 1.2.3, this thesis endeavours to model the hairstyle based on a single image. The main research aim is to carry out a framework to capture natural look hair model from a single image, which sheds new light on the image-based 3D hair geometry processing with structure analysis. This framework will broaden the current image based hair modelling technique, explore novel structure-aware hair modelling framework and improve the existing hair modelling system. With this in mind, this thesis tries to analyse the structure of hairstyle and reconstruct the whole head hair model from the extracted basic elements. To achieve the aim of this thesis, there are several objectives listed as follows:

- Develop a purpose-built image processing algorithm to extract 2D hair information and trace the 2D hair strands for 3D reconstruction preparation.
- Carefully design a hair shape descriptor to analyse and represent the hairstyle structure. This hair shape descriptor will keep local hair geometric information and empower complete hairstyle consistency.
- Build a critical hair shape database to involve a data-driven technique for 3D hair structure recovery. The database contains basic shapes of hair wisp, which will guide the 3D hair shape descriptor completion.
- Construct 3D orientation field to diffuse dense full head hair strands from the sparse hair structure. The orientation field ensures the hair strands are distributed evenly around the scalp and their shapes morph gradually, which avoids sharp discontinuity between hair regions.

1.5 Main contribution

To solve the questions raised ahead, the following experiments have been meticulously conducted, whose outcomes are dedicated to the main contribution of this thesis. In summary, this thesis proposed the pipeline Struct2Hair, a single-viewed hair modelling method based on a novel hair shape descriptor (HSD). To the best of the author's knowledge, Struct2Hair is the first bottom-to-top fashion of hair model creation from a single portrait input. The pop-up hairstyle is easily imported into the most popular industrial software including Houdini and Maya for realistic rendering. The detailed contributions of this system are listed below:

- This thesis provides a theoretical analysis of image filters applied to 2D hair orientation map detection. The image pre-processing results themselves are good inputs for art creation, like knowing the hair orientations and the pattern features helps to generate believable relief models of hair. There is a related contribution of a case study of hair Bas-relief modelling.

- A critical hair shape database has been built for retrieving the basic hair elements when generating the 3D structure of the hairstyle. Different from the previous hairstyle database, the Critical Hair Shape (CHS) database emphasizes the shape of hair wisps to give a special attention to the hairstyle structure. Potential structure-based hair modelling research will take advantage of this database. Also, the centre of each exemplar in CHS database might be the primal sketch to further benefit other sketch-based modelling researches. This database will be available to the public for non-commercial research after accomplishing clean-up and post-process.

- A novel hair shape descriptor (HSD) is proposed in this thesis to analyse the hair structure. The complete 3D hairstyle model created under the guidance of HSD preserves local geometric features of hair and retains the whole shape of hairstyle globally. Meanwhile, HSD is a compact structure for hairstyle storage and reloading. Beyond this, HSD also offers an easy solution for blending and editing the hairstyles discussed in application.

I have co-authored as the main contributor to produce the following academic papers to disseminate the work related to this research:

- Wenshu Zhang, Yinyu Nie, Kun Qian, Shihui Guo, Jian Chang, Jian Jun Zhang and Ruofeng Tong, 2017, Struct2Hair: A hair shape descriptor for hairstyle modelling, submitted to *The Graphical Models*, under revision.

- Wenshu Zhang, Meili Wang, Jian Chang, Jian Jun Zhang and Ruofeng Tong, 2015, Image-based hair pre-processing for art creation: A case study of bas-relief modelling. In 2015 19th International Conference on Information Visualisation, 411–418.

1.6 Thesis structure

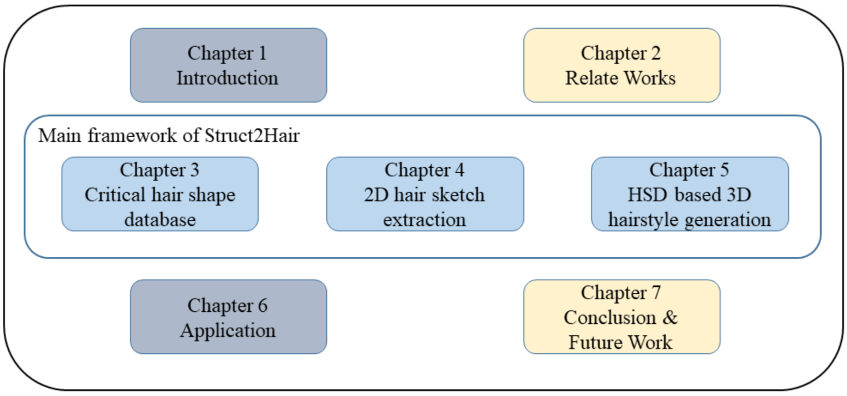


Figure 1.4: *The block diagram of the thesis structure.*

Chapter 1 introduces the background of the hair modelling. The essence of the single-viewed hair modelling method is addressed in this chapter, followed by the main contributions of this thesis. It outlines the individual components consisting the proposed Struct2Hair framework (Figure 1.4).

Chapter 2 reviews the related works to this thesis. It covers the history of hairstyling to state-of-the-art approaches, including explicit hair representation and image based hair acquisition. The hair simulation methods related to the hair synthesis techniques are also presented in this chapter. As the data-driven approach is participant in hair model construction, this chapter does a quick view of the most related projects.

Chapter 3 (Figure 1.5(a)) presents the method for critical hair shape database construction based on an existing hairstyle database. The Hausdorff distance with gradient coefficient is introduced to measure the distance and similarity between the hair strands. Based on the metric, this chapter applies the K-means++ method to group the hair strands into wisps with similar geometrical shapes and close distance. The hair wisps are the fundamental shapes of the hairstyles and gathered together as the critical hair shape database.

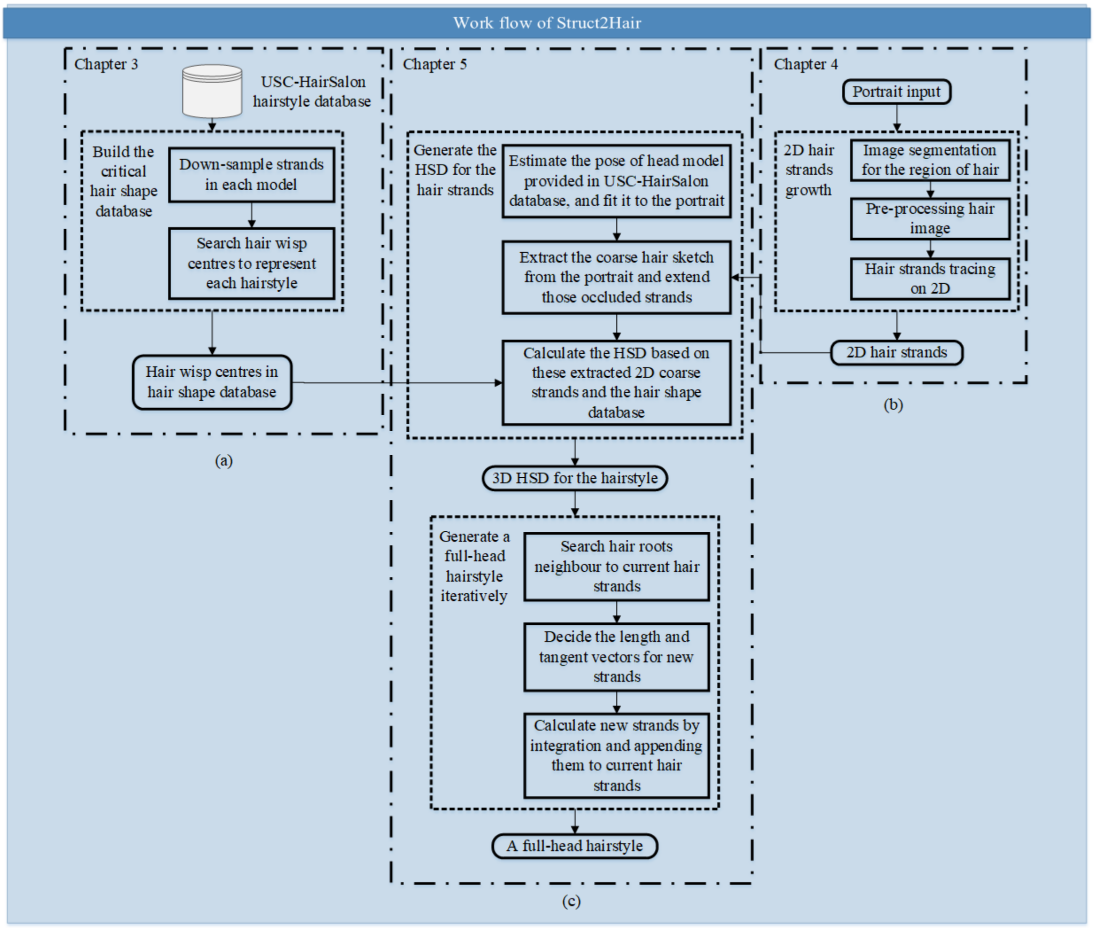

Work flow of Struct2Hair
Chapter 3
USC-HairSalon hairstyle database
Build the critical hair shape database
Down-sample strands in each model
Search hair wisp centres to represent each hairstyle
Hair wisp centres in hair shape database
(a)
Chapter 5
Generate the HSD for the hair strands
Estimate the pose of head model provided in USC-HairSalon database, and fit it to the portrait
Extract the coarse hair sketch from the portrait and extend those occluded strands
Calculate the HSD based on these extracted 2D coarse strands and the hair shape database
3D HSD for the hairstyle
Generate a full-head hairstyle iteratively
Search hair roots neighbour to current hair strands
Decide the length and tangent vectors for new strands
Calculate new strands by integration and appending them to current hair strands
A full-head hairstyle
(c)
Chapter 4
Portrait input
2D hair strands growth
Image segmentation for the region of hair
Pre-processing hair image
Hair strands tracing on 2D
2D hair strands
(b)

Figure 1.5: *Work flow of Struct2Hair. There are four main parts of the Struct2Hair. The first part is to build a critical hair shape database to apply data-driven approach (chapter 3). The second part is to trace 2D hair strands for the user input portrait picture (chapter 4). The key parts of this framework are HSD generation based on the outputs from the first two blocks, and the complete 3D hairstyle model creation from the guidance of HSD (chapter 5).*

Chapter 4 (Figure 1.5(b)) first extracts the region of hair by an image matting algorithm. Then several image processing filters have been implemented and the Gabor filtering has been chosen as the preferred 2D hair feature capturing method according to the comparison of their performances when pre-processing the hair images. The rest of this chapter is the 2D hair strand tracing algorithm.

Chapter 5 (Figure 1.5(c)) demonstrates the key technique of Struct2Hair. The proposed hair shape descriptor is built to represent the hair structure from the processed 2D hair sketch firstly. After that, the dense hair strands are diffused around the scalp by the dedicated orientation field conducted from the HSD. To validate the approach, the modelled hairstyles are compared to the ground-truth hairstyle models and the state-of-the-art single-viewed hair modelling methods respectively. The results show that the reconstructed hair models are closely matched to the portrait input with the original view.

Chapter 6 provides two promising applications based on the framework of Struct2Hair. One is to blend different hairstyles and the other one is an easy implementation of hair editing. A case study of Bas-relief hair modelling is conducted to further examine the image processing results as an additional application of this thesis.

The last chapter concludes the achievements and indicates some future directions from this thesis.

Chapter 2

Related Work

The hair modelling is trying to match the process of real-world hair shape generation. This thesis focuses on generating the geometrical shape of the hairstyle which accounts for the static hair acquisition. This section reviews the related direct hair synthesis methods and image based acquisition techniques. Other works related to this topic like the hair simulation and data-driven approaches are also interesting subjects of extensive study, which has been quickly reviewed here. Other relevant hair modelling topic such as rendering is out of the scope of this thesis. Two surveys Magnenat-Thalmann *et al.* (2000) and Ward *et al.* (2007) have detailed explanations of hair modelling. Ward *et al.* (2007) proposed that three types of techniques to generate global hair shape: geometry-based, physically-based and image-based techniques. This thesis follows its classification, but merges first two techniques into direct hair synthesis method.

2.1 Direct hair synthesis

The direct hair synthesis approach is designed for both generating hairs and simulating their dynamics. The shape description of the hair model also controls its governing dynamic parameters. To accomplish realistic hair modelling purpose, there are two main categories of the hair synthesis, which are the explicit and implicit models. Due to the essential design of this method, the related hair simulations are discussed together with the hair models in this section.

2.1.1 Explicit Approaches

The guide strand is the most popular feature for explicit hair representation, which reduces the size of the large number of hair strands in a typical human hairstyle. More hair strands are interpolated from the guide hairs for adding the density of the full hair, they could be in the form of wisps, clusters, strips or cylinders. The guide strand is a good primitive of manipulating fine hair details to make believable hairstyle. This method also breaks the barrier between the visual effects and computational requirements for the simulation.

The first hair wisp is structured by trigonal prism proposed in year 1986 (Watanabe and Suenaga, 1989). A more efficient hair wisp expression by using of trigonal prism is developed by Chen *et al.* (1999). Within a hair wisp, the continuous trigonal prisms are linked by 3 B-spline curves to indicate the general growth direction and the physics of the wisp. In order to model the cross section shape of the wisp instead of the triangle, a 2D hair strands distribution map is used to wrap the trigonal prism for guiding the hair strands grouping in a wisp. The full hairstyle consists of several sets of hair wisps following their locations on the scalp.

The styling and simulation are never isolated problems in hair modelling field. Lee and Ko (2001) designed a cantilever beam model to solve hairstyling and hair animation in the same equation. The hair strand is connected by the rigid segments, those long strands contains more segments than the short ones. They define that there are two DOFs at each joint, which control the geometric expression of the hair strand. Before the styling, the hair primitives are prepared with the specified length, density and gravity. Then an initial styling force is performed on the hairs to resemble the start hairstyle for animation. The magnitude of the styling force is collaborated with the external force to animate the hair movements, and remains constant during the whole process.

Although researchers have figured out ideas to modelling hairstyle realistically, the digital hair imagery is still too rigid and unsatisfied for photo-realistic applications. This difficulty has been eliminated when Bertails *et al.* (2006) introduced super-helix to capture the natural hair shape. Their work is the remarkable milestone in the history of hair modelling. The hair strand is assembled by integrating the hair length following the material frame built along the hair growth direction. The shape of the strands is created by tuning the geometrical parameters for each helical elements. The Figure 2.1 shows its ability of capturing the natural curliness of the hair strand.

2.1.2 Implicit Methods

Despite the explicit hair modelling offers a convenient way to manipulate the detailed hair shape, it will increase the computational complexity due to the large amount of hair strands. The aim of implicit modelling approach plays the other way around,

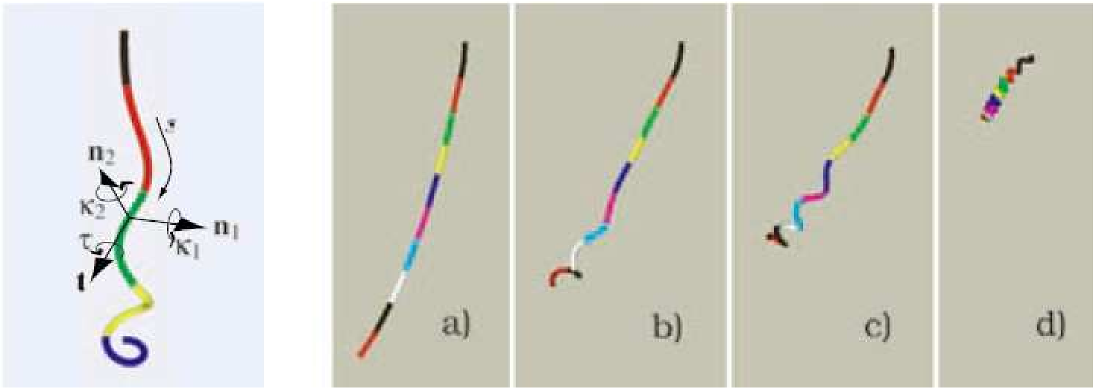


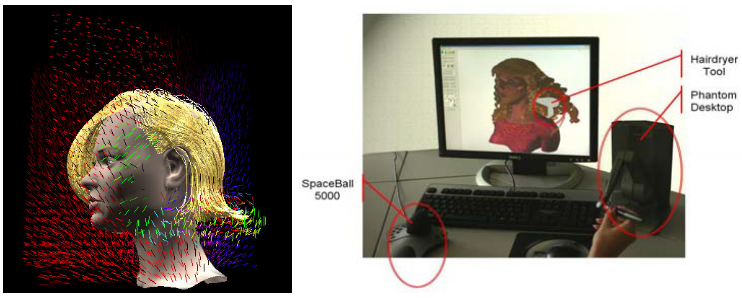
Figure 2.1: *Super-helices model by Bertails* et al. *(2006). Left: geometry of Super-helix. Right, animated different hair types: a) straight, b) wavy, c) curly, d) strongly curly.*

which sacrifices the fine detailed hair information with assembling the globally shape of a hairstyle.

The first and easiest way of modelling hair implicitly is to take hair shape as a texture applied to the polygonal surface aligned with the head model where the hairs locate. Lee *et al.* (1995) pioneered this texture-based method on facial animation with a plausible effect at 20 years ago. Ivanov *et al.* (2003) extruded a mesh surface for attaching hair texture to create the personalized head model, but the hair modelling is not addressed too much in this work. With the development of the imaging techniques and the 3D reconstruction algorithms, Echevarria *et al.* (2014) captured the detailed hair surface for stylisation (including color and geometry) and 3D fabrication.

However, it is difficult to animate the hair when encoded it into the texture map. A generalized cylinder with the embedded volume density model for the hair cluster abstraction is proposed by Yang *et al.* (2000). This model is designed for specifying the envelope shape of the hair wisp, which has been extended to a user interactive tool for hair design (Xu and Yang, 2001). It also enables a feasible rendering process with calculating the intersection of ray and generalized cylinder.

Another implicitly modelling approach is to treat hair as the streamlines within a fluid flow (Hadap and Magnenat-Thalmann, 2000). The linear combination of the stream lines (hair strands) with the ideal flow elements like source and vortex provides so many possibilities of hairstyle expression. To render the fine look of the hairstyle, a large number of hair strands is generated from the vector field. This type of modelling method is easy to develop a user interactive interface of hair styler (Figure 2.2 shows two streamline based hair styler developed by Yu (2001) and Magnenat-Thalmann *et al.* (2007)).



(a) (b)

Figure 2.2: *The implicit modelling approach by treating hair strand as the streamlines within a fluid flow. 2.2(a) is the initial hair model extracted from a vector field by Yu (2001). 2.2(b) is a phantom-assisted virtual hair dressing framework developed by Magnenat-Thalmann* et al. *(2007).*

2.1.3 Multi-resolution editing

Multi-scale modelling approach is popular in many fields such as engineering, mathematics and computer science etc. The aim of it is to detect and analyse the important features in different scales. For hair modelling, the drawbacks of computational complexity and lack of fine details manipulation are mentioned above for both explicit and implicit modelling methods. Therefore, the multi-resolution editing is introduced into hairstyling to make a tradeoff between computation efficiency and visual quality (Ward *et al.*, 2007).

To solve the aforementioned question, Kong and Nakajima (1999) calculate the different level of detailed geometrical shapes depending on the distance from the camera to the hair. They use background hairs to express the inner layer of a hairstyle, which are thick hair strands. The background hairs will be broken into much more thinner strands when they are near the surface and close to the viewer for rendering the fine look of the hairstyle. The visible volume buffer is proposed in this work to determine the surface hairs. The background hairs themselves largely reduce the memory usage when modelling and rendering the hair whilst the thin surface hairs guarantee the plausible visual experience.

Ward and Lin (2003) raise a Level-of-Detail (LoD) representation of hair model by the combination of the hair strands, the hair wisps and the hair strips. The base skeleton of the hairstyle is the highest level of a hair model, which controls the style and the physical behavior of the hair. The hair strands will be grouped into wisps and further into stripes when the LOD transition needed following the different viewing distance. The key ideal of this multi-scale model is to use the high resolution for achieving visual satisfaction, and leverage a low resolution for plausible hair volume visualization without expensive computations. This is a common balanced solution in the CG field.

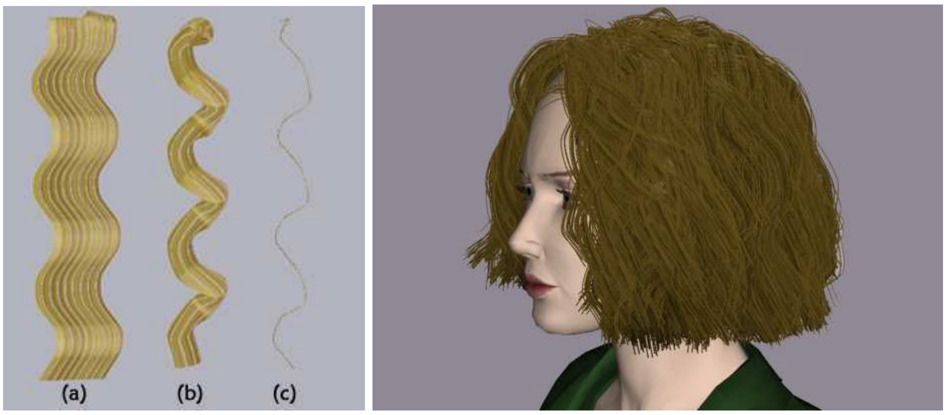


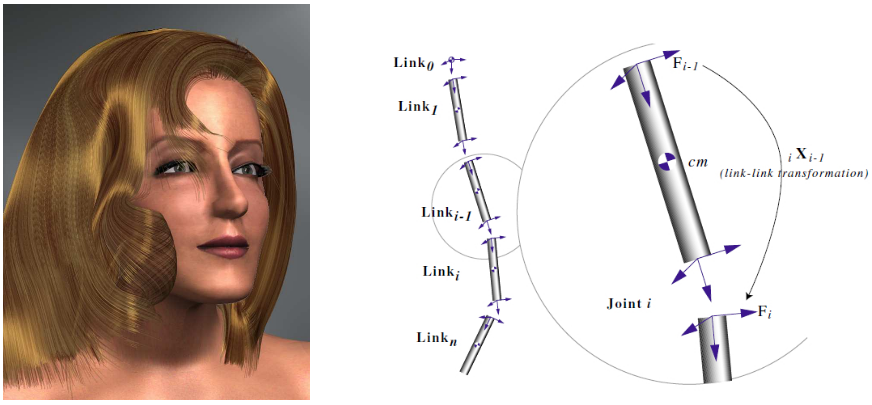
Figure 2.3: *Level-of-Detail representation of hair modelling (Ward and Lin, 2003): (a) strips (b) clusters (c) strand, and a corresponding fine detailed short, wavy brown hairstyle.*

2.1.4 Hair simulation techniques

The main purpose of hair simulation is to perform physically correct motion of hair, which is important for making realistic character's performance. A hair simulation method should be capable of deal with hair movements caused by the character's animation and interactions with objects. More complicated hair dynamics behaviours involve hair-hair and hair-air interactions. Meanwhile, the hair simulation techniques are largely related with the hair models, especially the direct hair synthesis models.

Early researchers attempted to animate hair strands by using simple mass-spring system (Rosenblum *et al.*, 1991). Every single hair strand is modelled as particles connected with springs and hinges. It is easy to implement due to only three degrees of freedom with one translation and two angular rotations. This straightforward mass-spring system obviously cannot reflect hair properties with twisting and non-stretching. Other advanced mass-spring simulation methods aim to solving numerical stability as well as animating realistic interacting hair (Choe *et al.*, 2005; Selle *et al.*, 2008; McAdams *et al.*, 2009).

Hadap and Magnenat-Thalmann (2000, 2001) pioneered to model hair as a continuum by using fluid flow. More precisely, their hair model has duality dynamic properties. Smoothed particle hydrodynamics (SPH) model is the fundamental pattern to handle hair-hair, hair-body and hair-air interactions. But for more realistic close-up observation, individual hair geometry is modelled as a serial rigid multibody chain with controllable parameters to generate bending and twisting. It is akin to immerse hair strands into fluid, in order to decrease duality computation complex, Bando *et al.* (2003) released hair strand structure by sampling hair volume with unordered particles. Reduced SPH model successfully animated hair interactions, but such unorganized particles are not feasible to simulate complicated hairstyle and lack of rendering realistic. To better control particle based hair arrangement and achieve more fast simulation, Müller *et al.* (2012) extended *Follow The Leader* (FTL) method to animate thousands of non-extensible hair strands/furs on virtual characters. They introduced some numerical damping to perform velocity correction which can avoid uneven mass distribution after an FTL projection. Rungjiratananon *et al.* (2012) modelled wetting hair by using Lagrangian method to simulate every single hair strand. They successfully modelled hair water interactions and paved the way for simulating more interesting effects of hair.



(a) Hair as fluid flow (b) Hair strand modelled as rigid multibody serial chain

Figure 2.4: *Model hair as a continuum by Hadap and Magnenat-Thalmann (2000)*

Researchers also considered more physically accurate algorithms to animate hair dynamics, for instance using Cosserat and Kirchhoff theories of rods. Bertails *et al.* (2006) built their Super-helices model by taking individual hair strand into non-stretchable rod. Each guide strand consists of several helical elements with different curvatures and twist to simulate various hair types (see in figure 2.1).

$$n_i^{'}(s,t) = \Omega(s,t) \times n_i(s,t), \text{ for } i = 0, 1, 2 \tag{2.1}$$

$$\Omega(s,t) = \tau(s,t)n_0(s,t) + \kappa_1(s,t)n_1(s,t) + \kappa_2(s,t)n_2(s,t) \tag{2.2}$$

A material frame $n_i(s,t)$ indicating tangential, normal and bi-normal directions, is defined along the centreline of hair strand at every point, attached with two curvatures $\kappa_i(s,t)_{i=1,2}$ and the twist τ respectively (see Equation 2.1 & 2.2 for detail). The Darboux vector $\Omega(s,t)$ is the angular velocity vector of the Frenet frame of a space curve. The generalized coordinates are reconstructed from this material frame for deformation configuration. Full head of hairs are interpolated from those guide strands. They verified this Super-helices model by accurately simulated the motion of hair compared with real sample.

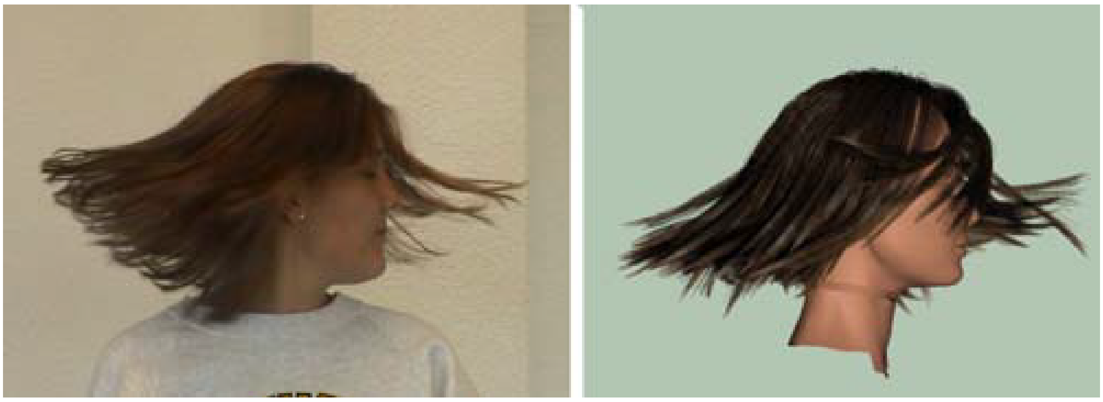
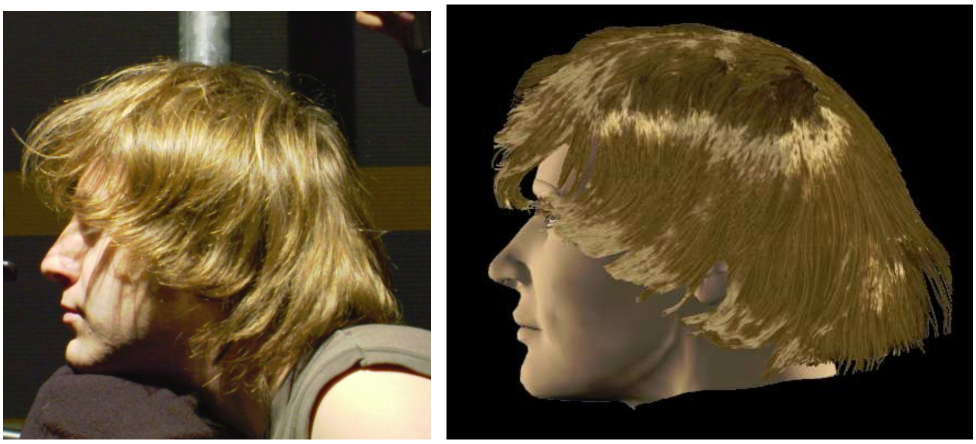


Figure 2.5: *Hair simulation using the super-helices model by Bertails* et al. *(2006). Comparison between a real full head of hair and the super-helices model, on a head shaking motion.*

Since the Super-helices model is related to helical numbers, large number will cause quadratic computation complex, Bertails (2009) proposed a linear time Super-helices to reduce it. For more complex rod assemblies, Kaufman *et al.* (2014) developed an adaptive non-linearity measurement to configure collisions. Their mathematics based algorithm can generate stable simulation even with large number of rods. Meanwhile, Position-based Dynamics (PBD) can also be integrated with elastic rod to do interactive hair design (Umetani *et al.*, 2014).

In order to accelerate hair representation and simulation, Kim and Neumann (2002) introduced multi-resolution methods to divide hair into three layers: individual strands, clusters and strips. Strands are one-dimensional subdivision curves, clusters are subdivision based swept volumes and the stripes are flat two-dimensional subdivision surface. Each level-of-detail (LOD) can transit by grouping or splitting layers. Base skeleton with control points is simulation primitive. When simulation triggered, appropriate LOD representation of hair section is chosen automatically to perform both dynamics accurate and visual realistic. The selection criterion involve visibility, viewing distance and hair motion. Only higher LOD's velocity surpass threshold, the lower LOD will be selected to participant simulation. This LOD representation reduced computations and obtained reasonable visual plausibility at the same time (Ward and Lin, 2003; Bertails *et al.*, 2003).



(a) one of the original images (b) reconstructed model

Figure 2.6: *Captured complete hair geometry by Paris* et al. *(2004)*

2.2 Image-based hair capture techniques

Despite the developments of computational hair model have create many fantastic applications, it still needs the long-time and tedious work to design a desirable virtual hairstyle manually. One of the main purposes of reconstructing hair shape is to save the labour cost in the traditional hairstyle modelling process. As images provide an intuitive sense of the natural hairstyles, most hair reconstruction techniques are capturing hair from image inputs.

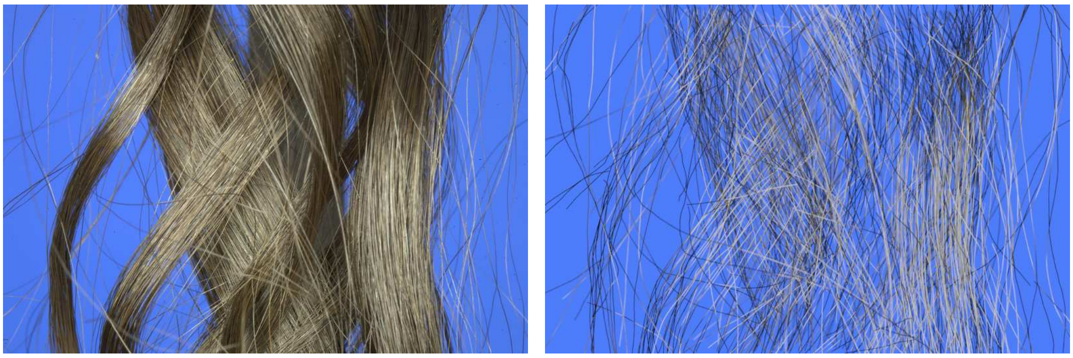
2.2.1 Multi-viewed hair modelling

Kong and Nakajima (1998) first proposed capturing virtual hairstyle by multiple images taken from different views. In this work, only right, back and top viewed images are used to extract hair outlines. These regenerated outlines help estimate hair volume and paint hair strands into this volume region. This method gave an idea about image based hair reconstruction and had the ability of modelling simple and coarse virtual hairstyle. It has limitations on recovering complex hairstyle and memory management at that time.

Grabli *et al.* (2002) extended shape from shading approach to hair geometry acquisition procedure. Their input pictures are taken under controlled light directions for tracing hair growing path. Final 3D hair strands are built by integrating 2D pixel orientation with hair illumination and reflection profile information. Differs from Kong and Nakajima (1998), they have no hair volume information to guide positioning hair strands in space, which leads to lack of accuracy in details. However, this novel method is a significant contribution to image based hair reconstruction question.

Using the similar data capture measurement, Paris *et al.* (2004) examined several image processing filters and captured a complete geometry of real person's hair. They compared their enhanced bilateral filtering technique with the Sobel technique in (Grabli *et al.*, 2002), and validated their method with the convincing results (see figure 2.6). This milestone work reconstruct a dense set of hair strands which paves the way for other applications including high-fidelity animation and rendering. In Paris *et al.* (2008), they benefited from more dedicated equipments to refine the capture setup. Triangulation approach added to retrieve and recover more intricate hair strands compared with their previous results. The authors also presented the rendering of animated hair geometry. In spite of high visual plausible result, this method still has some limitations. Strands grow in hidden part of the hair volume without any physical information may affect reconstruction of complicated hairstyles. Meanwhile, they only consider visible hair at given image pixels will cause error at silhouettes. It can seen from the paper, some bright highlights appear in the results caused by BRDF scheme. In spite of Paris *et al.* (2004) and Paris *et al.* (2008) captured unprecedented hair geometry, their results relied on high price of dedicated devices cannot be prevalent used.

Wei *et al.* (2005) proposed a novel method to capture multi-view hair images by hand-held camera without strict controlled illumination conditions. 2D hair fibres are automatically computed and organized by a sequence of chained line segments while local orientation of each hair pixel on each image is calculated simultaneously. Those image visible hair fibres are directly reconstructed from 2D image lines, otherwise are triangulated from visible views incorporated synthesize and validation scheme. Their multi-view capturing setup guaranteed accurate hair volume to guide positioning hair strands. However, as mentioned in their paper, uncontrolled illumination will raise unreliable local orientation. Meanwhile, multilayered hairstyle cannot be modelled due to inner occlusion.



(a) reference image

(b) reconstructed visible hair fibres

Figure 2.7: *Assembled hair fibres (Jakob* et al.*, 2009)*

Although the previous research works have done a good job, they failed to reconstruct actual individual strands of the input hair. Jakob *et al.* (2009) first proposed a image-based capturing method to assembly hair geometry fibre by fibre. Hair sample is mounted on a turntable to be captured from different directions, incorporates a moving digital camera with macro lens to sweep through the hair volume. Captured images are filtered to detect hair fibres in focus. Multi-viewed capturing setup compensate the orientation uncertainty and recreate the most visible hair strands (see figure 2.7). Finally inner hair volume are filled by artificially generated hair. The regenerated hair model have realistic 3D arrangements, benefiting further rendering and simulation applications.

Recently, Beeler *et al.* (2012) improved 3D face reconstruction technique to recover facial hair at the same time. In this research work, facial hair of the sparse region is individually recreated corresponding to the captured images and those hair fibres heavily overlapped are synthesized to match the reference images. They showed the ability to present the short and sparse hairs because their aim is to capture facial hairs. But for those thick and intricate beard, this approach cannot generate accurate results.

With aforementioned developments, image-driven hair capturing measurement can synthesize hair strands and recover hairstyle realistically. But for further applications, the cutting-edge technique is to capture more accurate hairstyle details. Luo *et al.* (2012) captured hair geometry with structure awareness. They first image hair from different views with a robotic gantry, then filtered those images to compute coarse-to-fine orientation fields under multiple resolutions. Stereo matching criterion applied on nearby views to reconstruct partial hair geometry by aggregating evidence. Full head of hair is recovered by aligning and merging these partial geometry pieces. They also claimed that their reargument is capable of capturing dynamic hair, but with less quality compared to static reconstruction.

The main limitations in this approach indicate their following achievements. Luo *et al.* (2013b) add more cameras of different baselines to best approximate captured hair volume. Meanwhile, they improved 3D orientation fields computation measurement with extracted coarse point cloud by Patch-based Multi-View Stereo(PMVS) algorithm (Luo *et al.*, 2013a). To perform structure-awareness, this novel method first grows local hair strand segments and group coherent strands into ribbons. Then it connects ribbons into hair wisp by restrict small curvature variation between nearby ribbons. After attaching those wisps onto scalp, final hair strands can be synthesized for rendering. Their framework successfully reconstructed different challenging hairstyles with correct topology (including complicated long curly hairstyle in figure 2.8). Based on this approach, Hu *et al.* (2014) modified strands fitting algorithm with simulated examples for more robust capturing result. Similar

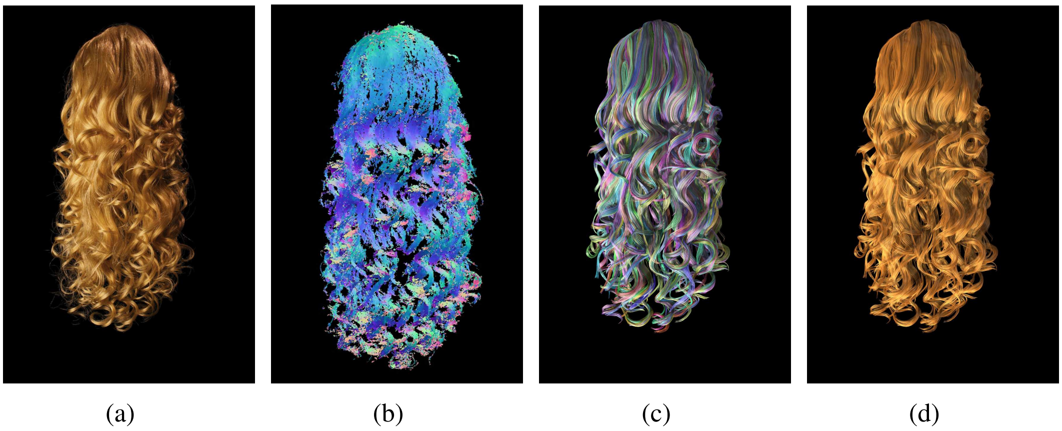

(a) (b) (c) (d)

Figure 2.8: *Structure-Aware Hair Capture by Luo* et al. *(2013a). 2.8(a) A sample of captured hairstyle 2.8(b) Extracted point cloud with 3D orientation field 2.8(c) Color-coded reconstructed hair wisps 2.8(d) Synthesized hair strands*

to previous works, point cloud and 3D orientation fields used to recreate hair strand segments, called cover strands in this measurement. They generate several databases containing examples strands simulated by Super-Helices method (Bertails *et al.*, 2006). By mean-shift approach, example strands could fit grouped cover strands and guide multiple ribbons connecting into hair wisp. Final results verified their robust hair capturing pipeline.

Traditional image-based hair acquisition techniques have limitations on handling shadowing and anisotropy in reflectance, and need manually segment hair region. Herrera *et al.* (2012) avoided these problems by introducing thermal camera for hair capturing. Since thermal camera captures infrared radiation, it's easy and robust to segment hair between skin and background. Hair strand temperature reveal the its distance to head, which integrated with hair volume reconstructed by visual hull, can regenerate boundary of hairstyle. 3D orientation fields are triangulated from a set of sparse thermal images (1000 in this case). Different from pervious hair strand tracing, reconstruction starts from most reliable region, indicated by 3D orientation and temperature to grow and quality improved by curvature constraints. Final recovered hair geometry has promising rendered results.

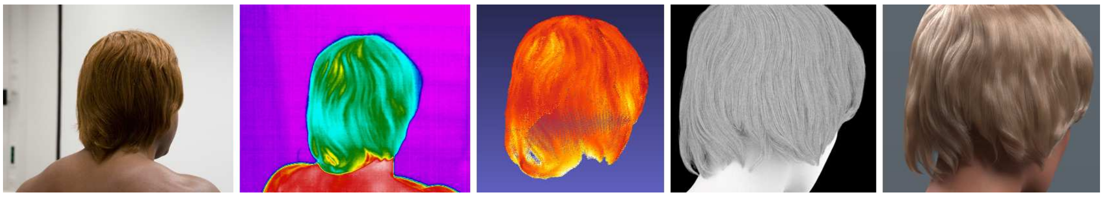


Figure 2.9: *A Thermal Approach To Hair Reconstruction by Herrera* et al. *(2012). From left to right, reference image; captured thermal image; reconstructed hairstyle boundary; rendered hair geometry with ambient occlusion; final rendered hairstyle*

Vanakittistien *et al.* (2016) and Zhang *et al.* (2017) use a small set of images to resemble realistic hair model, which avoid dense inputs acquisition. Both of them construct a rough shape of the hair volume and the 3D hair orientation field in it to generate hair strands. Vanakittistien *et al.* (2016) use several pictures (8 pictures) to build the visual hull for hair generation, the 3D hair orientation filed is calculated from the obtained 2D orientation maps on each input image. The final synthesised hair model reveals the original hairstyle appearance with less detail. Zhang *et al.* (2017) only use four-view images to reconstruct the hair model, which leverage a prior knowledge of the hairstyle shape from a large database. The key component of their framework is to synthesis continuous hair texture on the rough hair shape, from which a 3D hair direction field is computed to guide hair strands generation. The comparison of their methods is shown in Figure 2.10.

2.2.2 Single-viewed hair acquisition

The majority of image-based hair reconstructions have to recover the missing 3D information from the multiple views, which need the tedious capturing set-up preparation. Chai *et al.* (2012) prototyped modelling hair from single-viewed image. 2D hair segments grow along the local orientation detected by a set of Gabor filters inspired by Jakob *et al.* (2009), incorporates confidence constraints. Simple and heuristic depth estimation rules applied to recover hair volume. 3D hair strands traced randomly inside hair volume rather than scalp to confirm original viewed appearance.

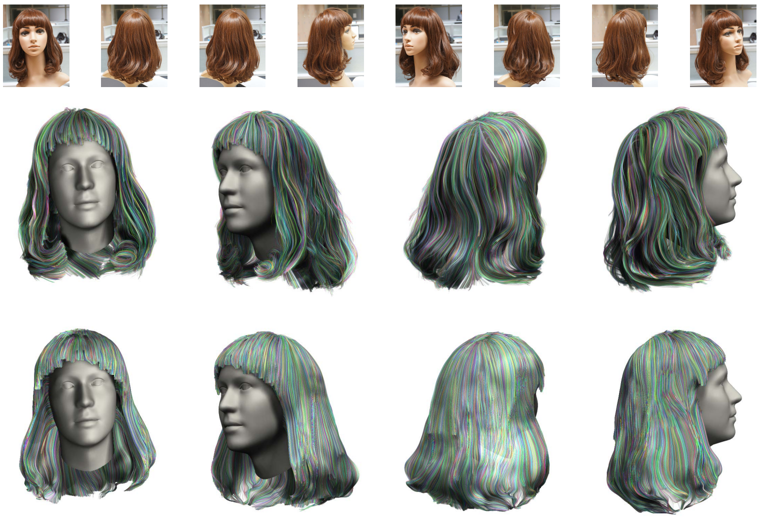


Figure 2.10: *Comparison between Vanakittistien* et al. *(2016) & Zhang* et al. *(2017). Second row is the result using first 4 input photos by Zhang* et al. *(2017) and the bottom row is the result of Vanakittistien* et al. *(2016) using all 8 photos. All the pictures from Zhang* et al. *(2017).*

Their results can be rendered in a slightly different view and transferred into other portrait head model (see in figure 2.11). Because the 3D strands are not traced from scalp, so it is not feasible to animate hair strands together with head movement.

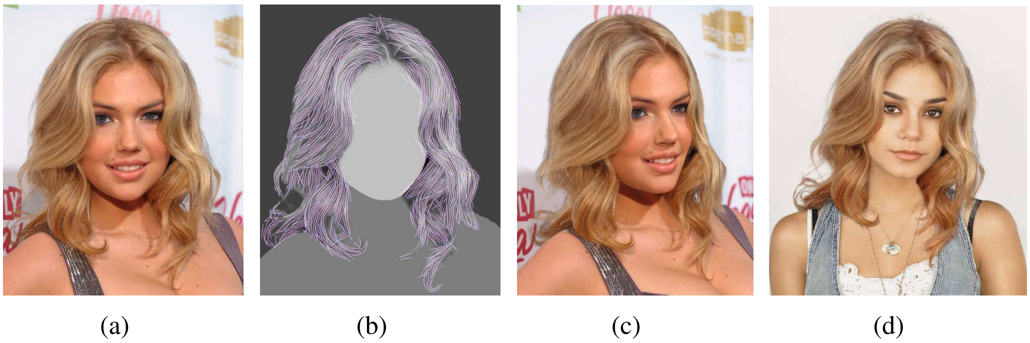


Figure 2.11: *Structure-Aware Hair Capture by Chai* et al. *(2012). 2.11(a) original single-viewed portrait, 2.11(b) regenerated hair strands, 2.11(c) rendered from a novel view and 2.11(d) transferred to another subject.*

Chai *et al.* (2013) extend their single-view hair modelling method to video manipulation. This improved method trace hair strands in 3D direction field from scalp after solving directional ambiguity in 2D space. Recovered physically valid hair strands are capable of being interactive editing and generating plausible dynamic simulation in video frames. In Chai *et al.* (2015), a shape from shading algorithm is used to refine their previous hair strands generation. They also added a helical prior for 3D hair strands optimisation to pop up high-quality portrait relief. However, the aforementioned approaches are incapable of generating full-head hair model due to the missing information in single image.

The two most related researches to this thesis, which model full-head hairstyle model from singe-viewed image by the data-driven method are Hu *et al.* (2015) and Chai *et al.* (2016) (see Figure 2.12). Hu *et al.* (2015) prototyped capturing whole hairstyle model with using a hairstyle database. Several user input strokes needed for guiding retrieving the hair example in the database. To make the hair modelling even more handling, Chai *et al.* (2016) introduced a fully automatic way to generate the hair model. The hair region in the portrait image is detected by a trained convolutional neutral network. They also constructed a database for data-driven hair modelling. The resulted hair model could benefit many future applications including physical-based hair animation and hair space navigation etc.

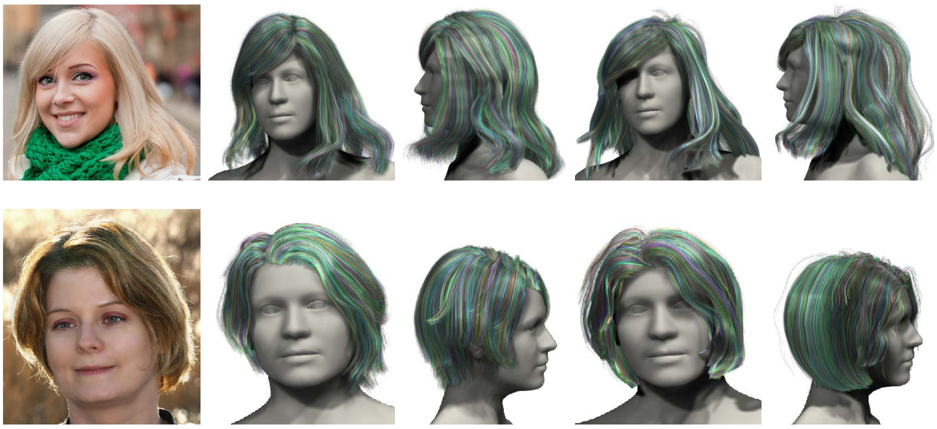


Figure 2.12: *Compare with state-of-the-art hair modelling techniques. From left to right are the input portraits, hairstyle model generated by Chai* et al. *(2016) and Hu* et al. *(2015) respectively. Original portrait courtesy of Bob HARRIS and Chris Zerbes.*

2.3 Data-driven Modelling

The data-driven method has been used in many areas in addition to the hair capturing. The success of data-driven based research shows its ability in many applications including modelling and controlling character animation etc (Blanz and Vetter, 1999; Anguelov *et al.*, 2005; Chaudhuri and Koltun, 2010; Baak *et al.*, 2013; Gao *et al.*, 2016). Among them, there are several data-driven based hair modelling methods that are similar to this thesis in spirit. Wang *et al.* (2009) syntheses new hair geometry from the example hairstyle by incorporating the 2D feature map and transferring the oriented detail of 3D hair example. They hierarchically build the new hairstyle from the coarse layer (global hair flow) to the fine detail (hair strands). This approach designs a solution of remixing hairstyles to generate a novel sample, which paves the way for the later data-driven based hair modelling from single image input. As a single portrait input naturally lacks of depth information, the visible hairs in the back of the head can only be inferred by heuristic method. Leveraging a hairstyle database can generate reliable photo-realistic hair model and keep the global hair shape consistency at the same time. Both Hu *et al.* (2015) and Chai *et al.* (2016) introduce single-viewed hair modelling frameworks benefiting from a hairstyle database.

2.4 Summary

This Chapter reviews the research works related to the hair modelling techniques, which includes modelling hair geometry, simulating hair dynamics and the corresponding data-driven approaches. Compared with the manually hairstyling process, the image-based hair acquisitions offer a trade off between time cost and visual realistic. To the best knowledge of the author, reconstructing hair geometry from single portrait image is state-of-the-art hair modelling method. Inspired by these fantastic works mentioned above, a single-viewed hairstyle generating framework Struct2Hair is proposed in this thesis with the combination of a data-driven method. Different from the current top-down pipelines (Wang *et al.*, 2009; Hu *et al.*, 2015; Chai *et al.*, 2016), the Struct2Hair analyses the key structure of a hairstyle and reconstructs the full-head hair model from the hair shape descriptor, which avoids hairstyle remixing and extra adjustments to match the hairstyle of the original picture. Since the hair shapes are difficult to be analysed by a generative approach, this bottom-to-top method is still quite challenge currently.

Chapter 3

Critical hair shape database

It is easy for human to understand the 3D structure from a still 2D image even if some parts of the scene are unknown by the view of projection. This is based on our understanding from daily life experiences. But it is hard for a computer vision system to infer 3D structure from a single viewed image due to the ambiguity between local features and their corresponding 3D location. In general, the single image based 3D reconstruction is an ill-pose problem. To simplify this problem, one can estimate the 3D shapes of 2D objects by adding depth priors with the help of a large dataset, instead of solving the depth information directly. The arising data-driven method cuts an extremely complicated problem into small feasible tasks, which is successful in 3D object reconstruction (Xu *et al.*, 2011; Baak *et al.*, 2013; Huang *et al.*, 2015).

In the research of hair modelling, data-driven method has dramatically improved the performance of image based hairstyling (Hu *et al.*, 2014, 2015; Chai *et al.*, 2016). With the usage of full-head hairstyle model database, the complete hairstyle corresponding to the original image can be generated by remixing the matched candidate hairstyles. Different with the state-of-the-art image based hairstyling systems, this thesis addresses constructing hairstyle from the basic hair shapes, which is taken as the fundamental structure of a hairstyle. In order to use data-driven method to avoid recovering complicated depth information, chapter 3 introduces a critical hair shape database.

3.1 Hairstyle databases

Motivated by the pioneered research works, this chapter introduces a critical hair shape database. Traditional manually construction of exhaustive hairstyle patterns is a laborious task. With the intricacies of various hairstyles coupled with large numbers of hair strands, grooming and combing virtual hairstyle require both technical manipulation and artistic inspiration for professional designers. Both of the two cutting-edge singe-viewed hair modelling methods (Hu *et al.*, 2014; Chai *et al.*, 2016) benefit from constructing a hairstyle database to avoid directly hair shape recovering from 2D to 3D. They collected hair models from the online repositories with characters from the Sims[1], and manually cleaned up and post-processed for their own projects.

3.1.1 USC-HairSalon database

USC-HairSalon database is introduced by Hu *et al.* (2014) to be used for a data-driven hair modelling prototype. In order to match the target hairstyle, they allow the user draw few strokes to guide candidate examples retrieving from the database. As their work is a sketch-based framework, they need strands based hair model for convenient search of hairstyles. Since the online hair models are designed for different virtual characters, they first manually align each model to a fixed standard head model. The original hair model consists of various triangle meshes, which represent a wisp of hair strands. After the alignment, they extract hair strands from the meshes uniformly around the scalp. Figure 3.1 details the procedure of pre-processing an example hairstyle. There are 343 3D hairstyle models in the USC-HairSalon database, Figure 3.2 previews the entire samples of this database. The collection of them have a wild range of overall shapes.

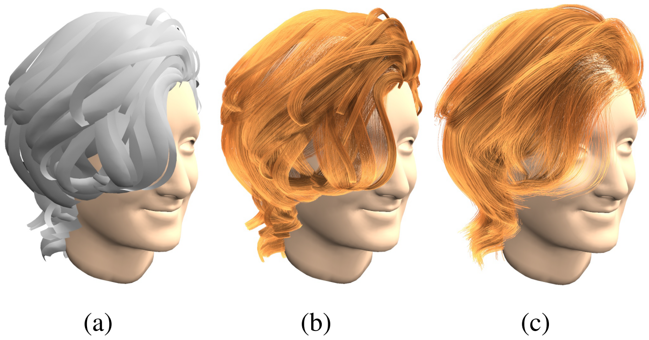


Figure 3.1: *Pre-processing of an example hairstyle by Hu* et al. *(2014). 3.1(a) is the original hair mesh downloaded from online repositories[2], 3.1(b) is the uniformly sampled hair strands on the original mesh, 3.1(c) is the final example hairstyle with evenly distributed hair strands.*

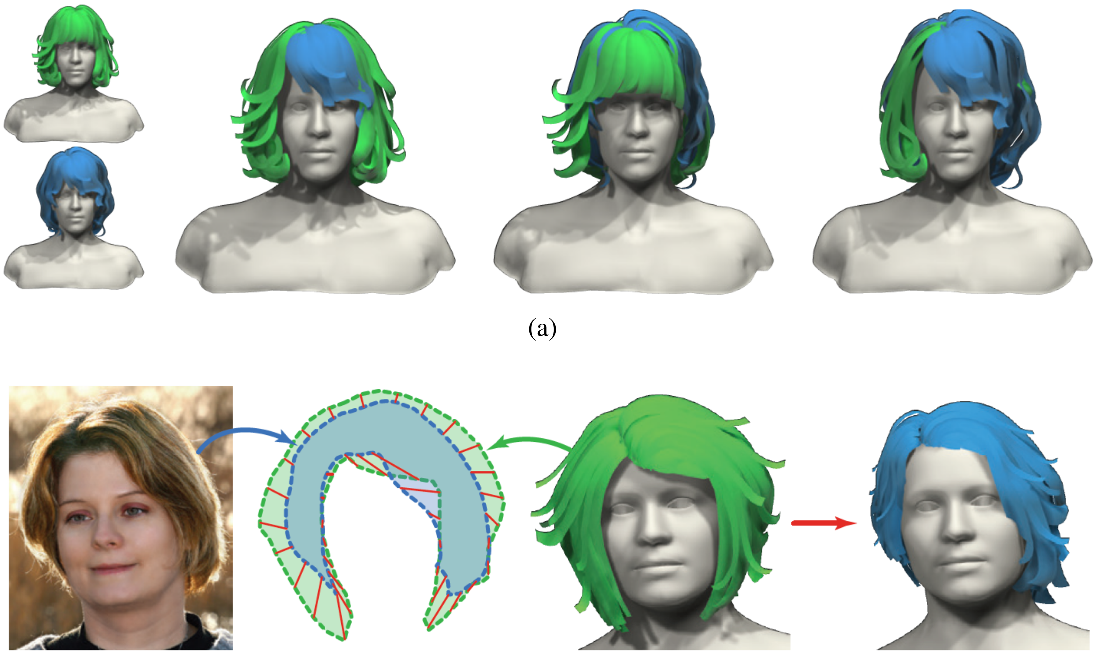
[1] The Sims is a life simulation video game series, developed by EA Maxis and published by Electronic Arts.

[2] The Sims resources and newsea Sims are two online repositories for the characters of the Sims.

[3] Image credit: Liwen Hu, Chongyang Ma, Linjie Luo and Hao Li. Weblink: USC-HairSalon.

Figure 3.2: *Miniature of the USC-HairSalon database*[3]. *The collection of hairstyles ranging between long wavy hairstyles and short straight hairstyles from the upper-left corner to the right bottom.*

3.1.2 3D Hairs in the Wild database



(a)

(b)

Figure 3.3: *Enrich 3DHW database by Chai* et al. *(2016). 3.3(a) remix existing hairstyle exemplars to populate a new one. 3.3(b) generating new hairstyle model from an online portrait picture.*

The 3D Hairs in the Wild (3DHW) is a large 3D hairstyle database, which is a part of the latest research work *AutoHair* conducted by Chai *et al.* (2016). They build this database by generating a large-scale 3D hairstyles for the online portraits. Same as USC-HairSalon database, the original hair exemplars are downloaded from the online repositories for the Sims. Then they keep populating new hairstyle models by two ways: enriching the exemplar set by exhaustively remixing existing pairs and generating the new hairstyle models from online photos (see Figure 3.3 for their approaches of enriching database). The combination measurement first enlarges their database of more than 40K models, then expanding to an even larger size of 50K with the generated hairstyles from online portrait images. There is no public access to the 3DHW database currently, the Figure 3.4 is a visualization of the sub-graph of it in Chai *et al.* (2016).

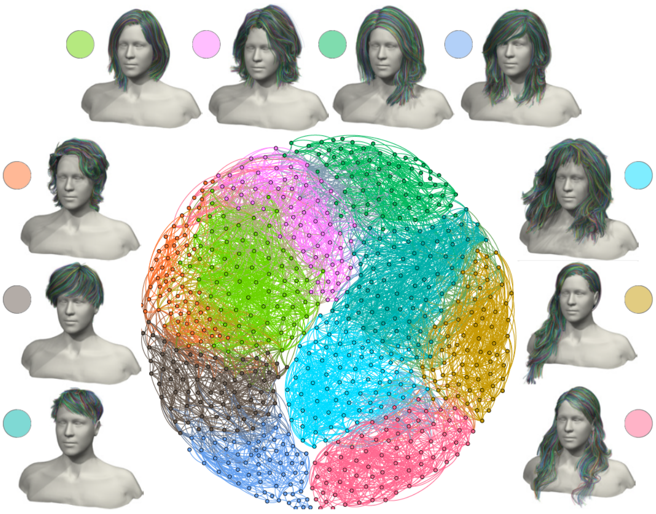


Figure 3.4: *A sub-graph visualization of 3DHW, full database needs request from http://gaps-zju.org/autohair/.*

3.1.3 3D hair model file

As the hair clustering is required for further hair structure analysis, the desirable hair file for a hairstyle is the collection of hair strands. There are two kinds of hair files to organize 3D hair model. One is hair.hair file, the other one is hair.data file, and both of them are formatted as the binary file.

The hair.hair file is proposed by Cem Yuksel in his cyCodeBase[4]. Compared to hair.data file, the hair.hair format contains more properties of the hair, including hair strands position, predefined hair thickness, the transparency and color information. The Hair file begins with a 128-byte long header followed by the geometry and topology data. The Hair file is designed for better hair rendering and simulation as Cem

[4] cyCodeBase by Cem Yuksel.

Yuksel hosts a series of research works on the subject of hair rendering, and Figure 3.5 shows an example of rendering result by Hair file. His webpage is recommended for further exploring.

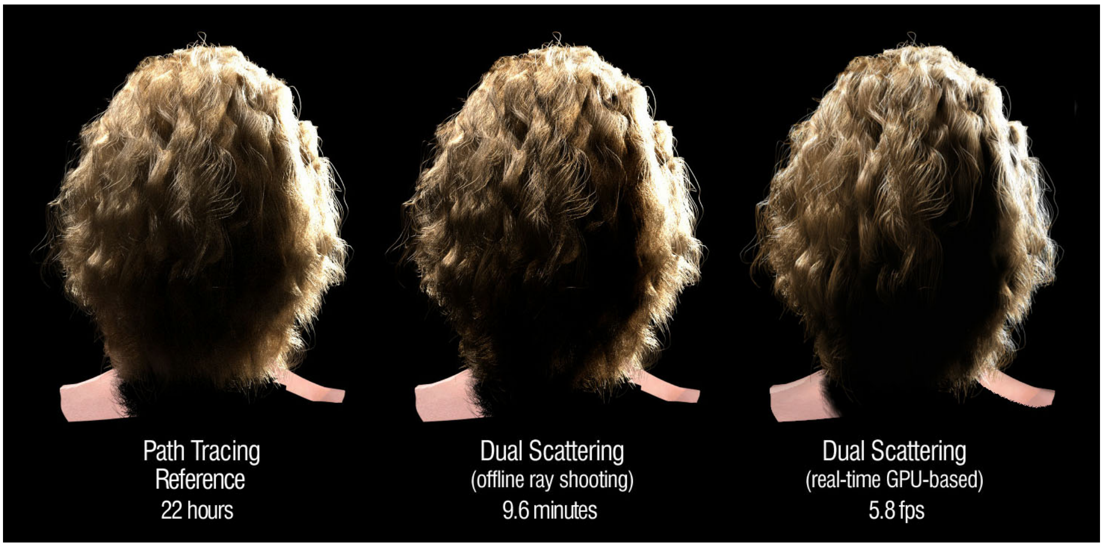


Figure 3.5: *An example of rendered hair with dual scattering proposed in Zinke* et al. *(2008). The hair model is saved as a hair.hair file*

The hair.data file consists of the number of hair strands, followed by points position starting with the number of points for each hair strand. This hair data format is compact and easy to understand, which simply profiles the basic geometrical information. It is popularly used by Paris *et al.* (2004, 2008); Hu *et al.* (2014, 2015). The format of the hair.data file shows below:

```
NumofStrands(int)
NumofVertices(int)
Vertex1(3 float) Vertex2(3 float) Vertex3(3 float) ...
NumofVertices(int)
Vertex1(3 float) Vertex2(3 float) Vertex3(3 float) ...
...
```

This thesis constructs the critical hair shape database based on the USC-HairSalon 3D hairstyle database, which organizes the hairstyle by using of .data file format. The hair strand is connected by the 3D points given by their space positions. This straightforward structure helps us focusing on the intrinsic hair shapes. The following sections are discussed the construction of critical hair shape database.

3.2 Build critical hair shape (CHS) database

This thesis aims to develop an approach of hairstyle shape generation from basic hairstyle structure, which is different from the current data-driven single image based hair modelling techniques. Clustering on hair strands aims to divide the hairstyle into hair wisps, where the hair strands in the same wisp possess similar shape to each other with respect to a given similarity measurement, whereas strands in different wisps are dissimilar.

Here, a two-hierarchic clustering method is proposed to search the optimal hair wisp representatives or centres to express the main distinctions of different wisps in distribution, length and shape etc. This clustering method can be divided into two steps: scalp region segmentation and wisp centres searching. The first step is achieved by a fast down sampling. It aims to reduce the calculation burden of the following step without losing too much information of strands' shape and then preliminarily classify strands by their location. The critical hair shape database is built based on the wisp centres.

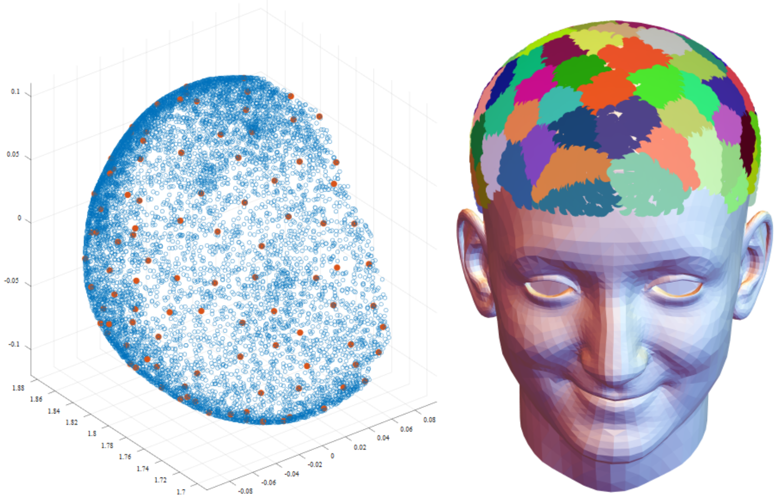
3.2.1 Scalp region segmentation

By assuming hair strands from the same region on the scalp generally have the similar length and shape in this work, the hair roots on the scalp have been segmented into m non-overlapping parts by a down sampling plan named as Minimum and Maximin Distance Design (Johnson *et al.*, 1990; Morris and Mitchell, 1995), which is representative in selecting samples to be evenly distributed in a spatial interval.

The 3D Cartesian coordinates of all the hair roots are denoted as $\boldsymbol{X} = \left\{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, ..., \boldsymbol{x}^{(n)}\right\}^T$, $\boldsymbol{x}^{(i)} \in \mathbf{R}^3, i = 1, 2, ..., n$, where $\boldsymbol{x}^{(i)}$ is the root node of the i th strand, and n is the number of strands. Instead of sampling from a spatial interval, a dataset $\boldsymbol{X}$ is the target. Hence interval sampling problem in Minimum and Maximin Distance Design is transformed to the point sampling by selecting the best m samples $\boldsymbol{X}'$ to minimise (Taheri and Ahmadian, 2016):

$$\min_{\boldsymbol{X}' \subset \boldsymbol{X}} \Phi_b\left(\boldsymbol{X}'\right) = \left(\sum_{j=1}^{k} J_j d_j^{-b}\right)^{1/b}, \tag{3.1}$$

where $\boldsymbol{X}'$ is a m-size subset of $\boldsymbol{X}$. $\{d_j\}$ is a vector of distinct distance values between all point pairs in $\boldsymbol{X}'$ (where the Euclidean distance is used here). J_j is the number of point pairs which can be separated by d_j. k is the size of $\{d_j\}$. b is a positive integer which is generally set as $20 - 50$ if it is a large problem. For how to select a proper



(a) (b)

Figure 3.6: *Scalp region segmentation. 100 optimal samples are selected from 9977 hair roots. 3.6(a) is the scattered hair roots around the scalp with 100 selected samples (red dots). 3.6(b) is the segmented hair regions according to the down sampling process.*

value, the reference Morris and Mitchell (1995) is recommended.

Considering there are nearly 10,000 strands in each hair model, in this work b is set as 50, and adopt the integral programming to heuristically solve the optimisation problem 3.1 with the genetic algorithm (Taheri and Ahmadian, 2016). Then the optimal samples $\boldsymbol{X}'$ are evenly distributed on the scalp which can represent the whole hair roots $\boldsymbol{X}$. After that, for each hairstyle model, the scalp can be segmented into several non-overlapping areas by searching for the neighbour hair roots of each sample in $\boldsymbol{X}'$ by the kNN algorithm (see Figure 3.6).

3.2.2 Hair wisp centres searching

For each hairstyle model in the database, after coarsely dividing the n hair strands into m groups, the second layered clustering method is then applied to ensure strands within same cluster sharing similar hair shape.

K-means++ clustering. With the appealing simplicity and speed in practice, the k-means method is one of the most popular clustering algorithm used in scientific and industrial applications (Arthur and Vassilvitskii, 2007). But in k-means method, the algorithm starts with k arbitrary initial ”centres”, which are typically chosen uniformly from the dataset. However, the randomly seeding process weakens its accuracy. Thus, the k-means++ is introduced with careful seeding to improve both the speed and accuracy of k-means. The algorithm 3.1, 3.2 show how the k-means++ serves searching hair cluster centres.

Algorithm 3.1 The standard k-means algorithm

```
1: Randomly choose an initial $k$ centres $\mathcal{C} = c_1, c_2, \ldots, c_k$ from a set of n data points $\mathcal{S}$
2: For each $i \in 1, \ldots, k$, set the cluster $C_i$ to be the set of points in $\mathcal{S}$ that are closer to $c_i$ than they are to $c_j$ for all $j \neq i$.
3: For each $i \in 1, \ldots, k$, set $c_i$ to be the centre of mass of all points in $C_i$: $c_i = \frac{1}{|C_i|} \sum_{s \in C_i} s$.
4: Repeat Steps 2 & 3 until $\mathcal{C}$ no longer changes.
```

Algorithm 3.2 The k-means++ algorithm

```
1: Take one centre $c_1$, chosen uniformly at random from $\mathcal{S}'$.
2: Take a new centre $c_i$, choosing $s \in \mathcal{S}'$ with probability $\frac{d_s^2}{\sum_{s \in \mathcal{S}'} d_s^2}$.
3: Repeat step 2 until $k$ centres have been taken together.
4: Proceed as with the standard k-means algorithm (Algorithm 3.1).
```

In Algorithm 3.1 and 3.2, $\mathcal{S}$ represents all hair strands within a hairstyle model, whose roots correspond $\boldsymbol{X}$. The $\mathcal{S}'$ is the hair strands with roots in $\boldsymbol{X}'$. k is the user pre-defined number of hair clusters, d_s^2 denote the shortest distance from a hair strand to the closest centre which has been already chosen. Here, k centres are chosen from the optimal samples mentioned at section 3.2.1 to improve the efficiency.

3.2.3 Distance between hair strands

Hausdorff distance. Unlike measuring the distance between two points in the space, the distance between two curves is much more complicated since they could have different shapes and lengths. As a popular approach in measuring the distance between two curves in a metric space (Chen *et al.*, 2010), the Hausdorff distance $d_H(\boldsymbol{s}_i, \boldsymbol{s}_j)$ is introduced to express the distance between two hair strands. It is presented as:

$$d_H(\boldsymbol{s}_i, \boldsymbol{s}_j) = \max \left\{ \sup_{q \in \boldsymbol{s}_j} \inf_{p \in \boldsymbol{s}_i} d(p, q), \sup_{p \in \boldsymbol{s}_i} \inf_{q \in \boldsymbol{s}_j} d(p, q) \right\} \tag{3.2}$$

In equation 3.2, s_i, s_j are denoted as the ith and jth strand of a hair model, p and q are the nodes distributed on them respectively, where $p, q \in \mathbf{R}^3$. The $d(p, q)$ adopts the Euclidean distance (see Figure 3.7 for details). However, the hair strands in USC-HairSalon database have the same number of points, no matter long hair strand or short. This will cause problem when calculating one-sided Hausdorff distance from short strand to long strand. Therefore, a cubic spine is used to fit the original hair

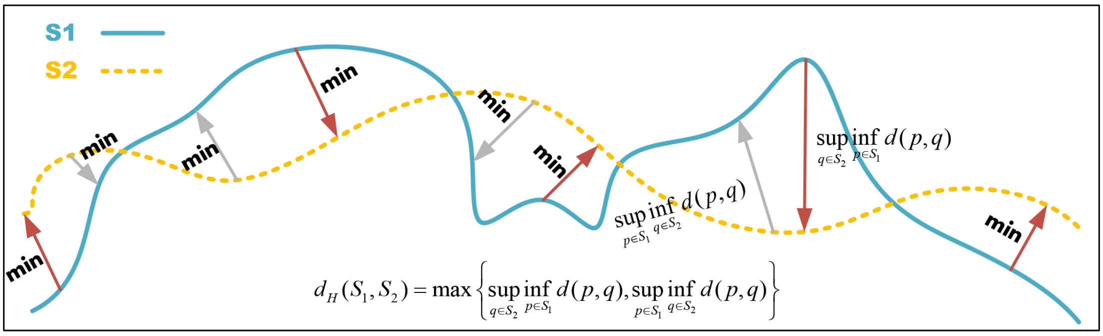

S1
S2
min
min
min
min
min
min
min
min
$\sup_{q\in S_2}\inf_{p\in S_1} d(p,q)$
$\sup_{p\in S_1}\inf_{q\in S_2} d(p,q)$
$d_H(S_1,S_2)=\max\left\{\sup_{q\in S_2}\inf_{p\in S_1} d(p,q), \sup_{p\in S_1}\inf_{q\in S_2} d(p,q)\right\}$

Figure 3.7: *Hausdorff distance between strands $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$. p and q are the points located on $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$ respectively. The Hausdorff distance is $\sup_{q\in\boldsymbol{s}_2}\inf_{p\in\boldsymbol{s}_1} d(p,q)$ in this case.*

strand, which will ensure the same length between two consecutive points (see Figure 3.8).

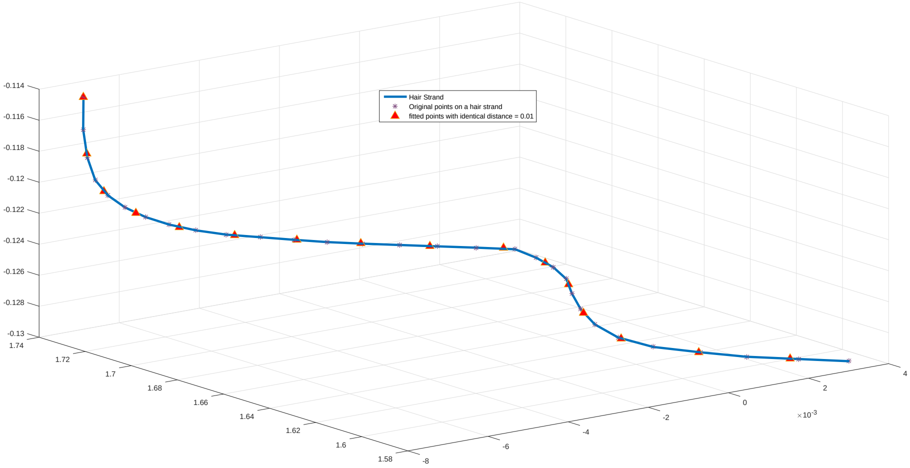


Figure 3.8: *Fit hair strand by cubic spine with identical distance between consecutive points, the curve segment cuts from the original hair strand.*

Gradient coefficient. The $d_H(\boldsymbol{s}_i, \boldsymbol{s}_j)$ explains two strands with similar shapes will share a relatively smaller distance. But two strands with different shapes still have chance to hold a small distance if they are close enough in the space. To distinguish strands with different shapes, a metric is provided where three factors are mainly considered: strand length, spatial position and gradient value:

$$d_s(\boldsymbol{s}_i, \boldsymbol{s}_j) = \frac{d_H(\boldsymbol{s}_i, \boldsymbol{s}_j)}{g_{i,j}}, l \geq 2. \tag{3.3}$$

$g_{i,j}$ is the variable to represent the gradient information between two hair strands. It is computed as inner product between the gradient vectors of the two strands, which is regarded as a coefficient added in the Hausdorff distance. Then the metric to measure the distance between two strands is given as:

$$g_{i,j} = \frac{1}{l-1} \sum_{k=1}^{l-1} \langle \text{grad}\left(p_k\right), \text{grad}\left(q_k\right) \rangle \tag{3.4}$$

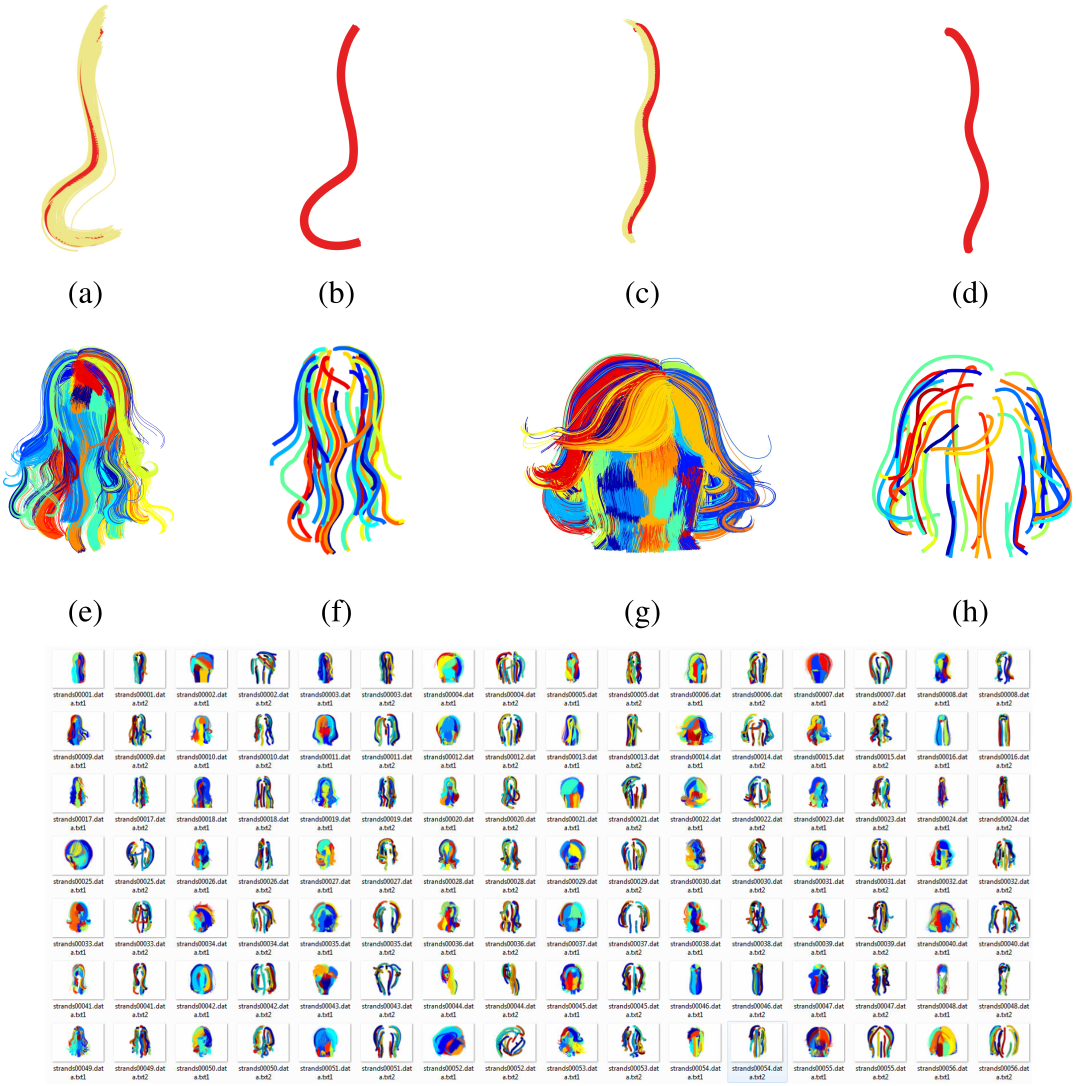
grad (p_k) and grad (q_k) respectively denote the unit gradient vector at point p_k (on s_i) and q_k (on s_j). l is the number of points on the shorter strand, wherein, to make points on all strands distributed equally in distance, the cubic spline interpolation is used to re-sample these points beforehand. The metric in equation 3.3 shows that two strands which have a similar length, location and gradients will contribute to a smaller distance.

In section 3.2.1, the down-sampled strands have been obtained. Each strand there represents an individual spatial area on the scalp. Based on the metric, the K-means ++ method is applied to gather these strands into c clusters. Then there yields c centre strands, each of which can largely reflect the overall style of strands in a cluster. The clustering result of the hair models are shown in Figure 3.9.

The critical hair shape database is developed based on USC-HairSalon hairstyle database (Hu *et al.*, 2015). By doing the clustering on 343 models in the hairstyle database in section 3.2.2, a large set of hair wisps (see Figure 3.10) are collected. Those hair wisps exhaustively demonstrate natural hair shapes ranging from short straight hairstyles to long curly hairstyles. Therefore, this thesis calls such hair wisps as critical hair shapes and collect them as a database of critical hair shape. The critical hair shape centres have been stored for fast database retrieving. They also generate an interesting application in chapter 6.

3.3 Summary

This chapter demonstrates constructing a database composed of critical hair shape extracted from the USC-HairSalon dataset. The main purpose of building such a database is to choose a minority of hair strands from each full-head hair model, which can capture the representative features of their overall hairstyle. To improve computation efficiency, 10000 hair strands roots have been down sized to 100 optimal samples, which also divide scalp into 100 regions. Then the k-means++ is performed on the processed hairstyle model to cluster hair strands and locate the cluster centres at the same time. The Hausdorff distance combined with gradient is used during the clustering to better measure the similarity of geometrical shapes between hair strands. After the clustering processed on all the hairstyle models, the dataset with a collection



(a) (b) (c) (d)

(e) (f) (g) (h)

(i)

Figure 3.9: *Hair strand clustering. 3.9(a) and 3.9(c) are two clusters of hair wisps, 3.9(b) and 3.9(d) are the corresponding cluster centres. 3.9(e) and 3.9(g) are two hairstyle models from Hu* et al. *(2015). Their hair strands have been grouped into 50 clusters, using different colours to show hair clusters. 3.9(f) and 3.9(h) are the corresponding 50 cluster centres. 3.9(i) is the miniature of the clustering done on the hairstyle database.*

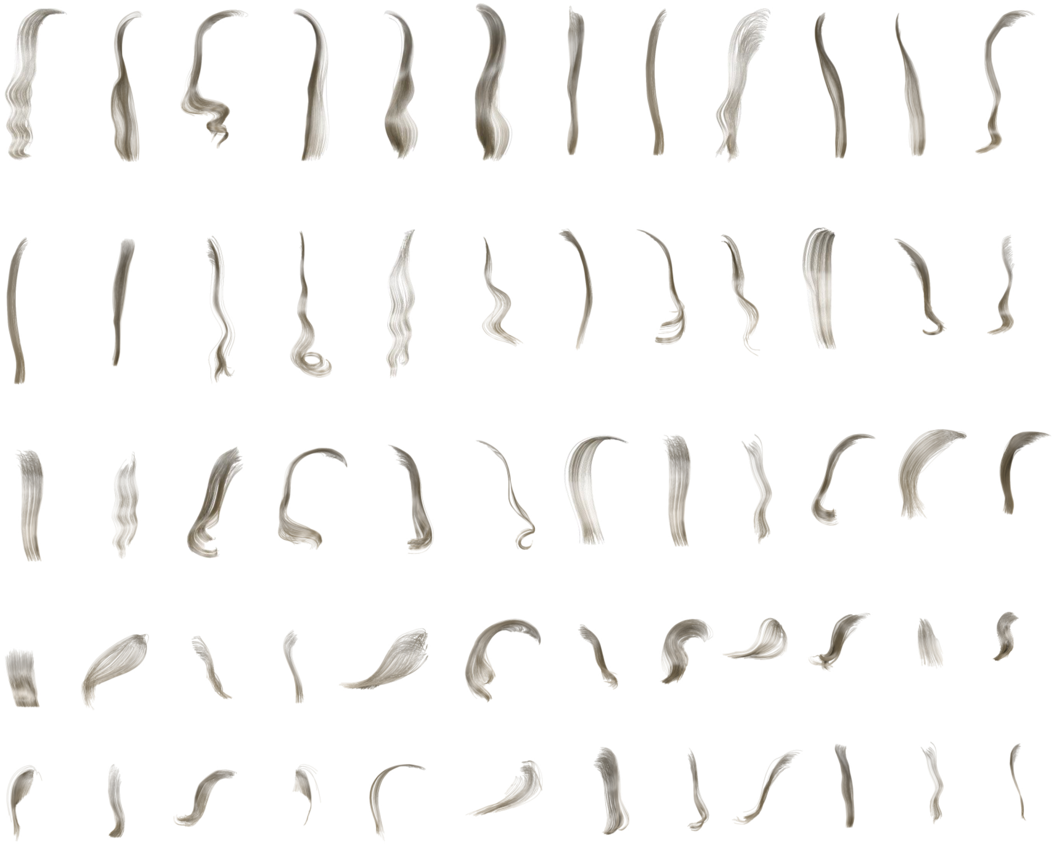


Figure 3.10: *Preview of the Critical Hair Shape database. Those samples are chosen from the CHS database, there are more available samples under clean-up and post-process.*

of 17,150 critical hair shapes is constructed. The critical hair shape database enables a data-driven approach for image based hairstyle reconstruction in the next chapter.

Chapter 4

2D hair sketch extraction

Inspired by the aforementioned hair modelling researches, this thesis proposed a single image based hair modelling method which focuses on hairstyle structure analysis and 3D photo realistic hairstyle generation. To better capture the shapes as well as the rich dynamics of hair, image based modelling techniques have been developed for reconstructing their 3D geometry and important visual features. Due to illumination condition and hair shape intricacies, most hair images contain inevitable noise which impair extracted hair models. To avoid this situation, pre-processing image has usually been used in advance to improve the performance of 2D hair strands extraction. Meanwhile, the pre-processing results have enhanced the 2D hair features, which is applied for bas-relief stylisation as a case study in chapter 6.

In this chapter, image pre-processing including image matting and image filtering are introduced first. Section 4.1.2 provides a theoretical analysis of image filters applied to 2D hair orientation map detection. Afterwards, an algorithm of tracing 2D hair strands on the image plane along the captured orientation map is demonstrated. In the end, the traced 2D hair strands have been clustered into groups with similar hair shape. Each cluster centre is a hair shape curve. We collect all of them as the extracted 2D hair sketch, where the centres are treated as the strokes in 2D hair sketch. The hair sketch is the connection between prior knowledge (Critical hair shape database in Chapter 3) of hair shape and reconstructed 3D hairstyle model (HSD based hair model in Chapter 5).

4.1 Image pre-processing

4.1.1 Region of hair segmentation

Since the reference hairstyle comes from portrait, extracting the region of hair from input image is the first task. Image matting aims to extract foreground objects from images or video sequences, which contributes to image and video editing applications. Mathematically, image matting problem can be described as following equation:

$$I = \alpha F + (1 - \alpha)B, \tag{4.1}$$

where I represents observed image, F and B are the foreground and background respectively, and α is the matte (Wang and Cohen, 2008). For a given pixel $p(x, y)$, $I(p)$ is a 3D color vertex. For the right hand, the unknown $F(p)$ and $B(p)$ have same size with $I(p)$, and scalar factor α ranges between [0,1]. For those definitely background pixels, α is assigned as 0, α equals 1 vice versa. Otherwise, α is between (0,1) as a mixed pixel. Figure 4.1 shows an example of trimap. It is infeasible to solve 7 unknown variables from 3 known variables. So most of image matting methods require user interactions to decrease the solution space. For instance, a few user scribbles or a corresponding trimap is enough to roughly segment different layers of the image. Based on user assistant, image matting problem is reduced as solving α values for mixed pixels.

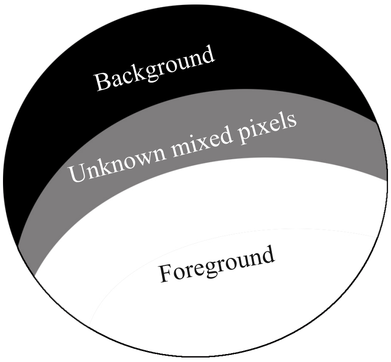


Figure 4.1: *Trimap*

In this research, it is easy to distinguish hair part from the background. So the user specified trimap is used in Figure 4.2(b) to mark majority hair as the definitely foreground. For those hair wisps mixed with background are marked as the unknown pixels. The recovered matte only have 1 and 0 which means the extraction is a binary segmentation problem. KNN-matting algorithm helps us generate matte under user specified trimap and split hair from the background (Chen *et al.*, 2012). The computed region of hair and α matte (see Figure 4.2(c) and 4.2(d)) will be used in next step.

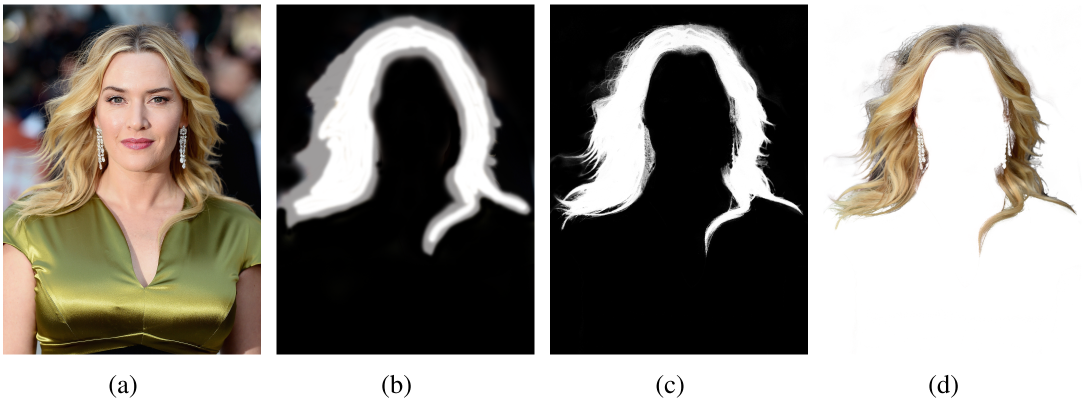


Figure 4.2: *User marked trimap 4.2(b) on original portrait 4.2(a). 4.2(c) is the computed α matte. 4.2(d) is the extracted region of hair. The portrait is chosen from the IMDB-WIKI dataset (Rothe* et al.*, 2016).*

4.1.2 Image filtering

The image provides an essential geometric clue of hair strands like their position and orientation, which contributes to recent success of image-based hairstyle reconstruction. Most pre-processing of hair images (Paris *et al.*, 2004; Jakob *et al.*, 2009; Grabli *et al.*, 2002; Wei *et al.*, 2005) included de-noising and orientation detection which relates to edge detection algorithms. A selection of image filters can fit this role well. There are many edge detection algorithms, like Sobel operator (Kanopoulos *et al.*, 1988) and Canny edge detector (Canny, 1986), which locate edges in a given image. The edge is obtained by thinning and thresholding process. Because, most edge detection algorithms do not hold the edge alignment information which indicates the orientations of hair strands, therefore additional computation is required to grow and connect hair strands. In the image plane, the projected hair fibre growth direction is perpendicular to a particular contrast direction (Grabli *et al.*, 2002). By doing so, most visible strands can be extracted.

Little attention has been paid to investigate the pre-processing methods for hair modelling or reconstruction from image. In this section, four different filters which can be used to extract the hair strand orientation are investigated and analysed.

Sobel Filter In an image, the gradient describes the directional change of intensity or colour. On a captured hair fibre, it is assumed that the pixel's gradient direction is perpendicular to its growth orientation at this point in the image projection plane.

Therefore, researchers used gradient-type filters to estimate hair's 2D orientation field in an early stage (Grabli *et al.*, 2002). The gradient of an image I is denoted as:

$$\nabla I = \begin{bmatrix} I_x \\ I_y \end{bmatrix} = \begin{bmatrix} \frac{\partial I}{\partial x} \\ \frac{\partial I}{\partial y} \end{bmatrix}. \tag{4.2}$$

The gradient direction θ_S can be calculated as

$$\theta_S(x, y) = \tan^{-1}\left(\frac{I_y}{I_x}\right). \tag{4.3}$$

The partial derivatives need to be calculated on every pixel to obtain an image orientation map. Popular operators like Sobel Operator directly compute intensity difference between neighbour pixels to get an approximation of the partial derivative by convolving operation. The Sobel operator (Kanopoulos *et al.*, 1988) (figure 4.3) is represented as two convolving masks along y direction and along x direction for differential operation along y and x direction respectively.

-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

Figure 4.3: *Sobel 2D masks of size 3*3*

Gaussian Derivative Filter As the Sobel operator is sensitive to noise and edge discontinuity, a low-pass filter can be convolved with hair image to reduce the impact of noise, like Gaussian smoothing.

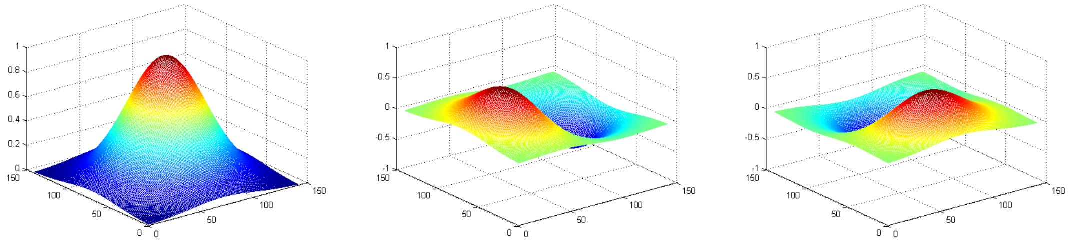
$$G_\sigma(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}. \tag{4.4}$$

The standard deviation σ controls the behaviour of the Gaussian filter, bigger σ removes more noise/details and produce smoothed results after filtering. By calculating derivatives on smoothed hair image, more reliable gradient orientation θ_{DOG} is computed (Kennedy and Basu, 1999).

$$I_x = \frac{\partial G_\sigma(x, y)}{\partial x} * I, \tag{4.5}$$

$$I_y = \frac{\partial G_\sigma(x, y)}{\partial y} * I, \tag{4.6}$$

$$\theta_{DOG} = tan^{-1}\left(\frac{I_y}{I_x}\right). \tag{4.7}$$



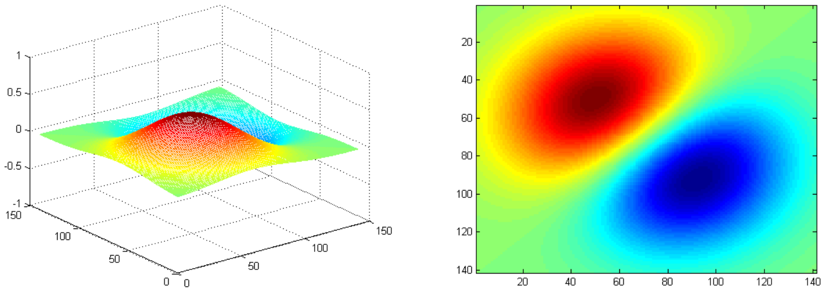
(a) (b) (c)

Figure 4.4: *Traditional Gaussian filter 4.4(a) and the corresponding Gaussian X derivative filter 4.4(b) and Gaussian Y derivative filter 4.4(c).*

Steerable Filter This thesis assumes that the growth direction of a captured hair fibre varies between [0°,180°). But the conventional gradient-type filtering methods are constrained within x and y directions when moving the masks, which dose not calculate gradient orientation directly. Flexible filter like a set of Gaussian derivative based Steerable filters (Freeman and Adelson, 1991) can obtain gradient magnitude along arbitrary direction. The Steerable filter is a linear combination of Gaussian derivatives in x direction and y direction. Gradient orientation is corresponding to direction with the maximum response, where I_x, I_y use the denotation of equation 4.5 & 4.6.

$$Gradient : G_I(\theta) = cos(\theta)I_x + sin(\theta)I_y \tag{4.8}$$

$$\theta = argmax_{\theta}(G_I(\theta)) \tag{4.9}$$



(a) (b)

Figure 4.5: *Steerable filter with orientation 45°. 3D mesh of the steerable filter 4.5(a) and its 2D projection 4.5(b).*

Gabor Filter Another directional filter called Gabor filter (Mehrotra *et al.*, 1992) is largely used in recent hair capturing research. This biological inspired filter is a combination of parameters comprising spectral bandwidth, frequency and orientation, which supports both spatial and spectral analysis. A batch of Gabor filters with different orientations are designed for estimating 2D orientation field. Gabor filter is denoted in Equation 4.10. According to (Petkov, 1995), standard deviation σ and wavelength λ of cosine factor are linked by the half-response spatial frequency bandwidth b (in octaves), see in Equation 4.13. The value of σ changes through bandwidth, which cannot be specified directly. The value of $b = 1$, $\sigma = 0.56\lambda$ are set in all the experiments, and λ is specified in pixels. Different with the gradient-type filters, the maximum response of Gabor filter convolution directly estimate the hair growth direction.

$$K_\theta = exp\left(-\frac{x_\theta^2 + \gamma y_\theta^2}{2\sigma^2}\right) cos\left(\frac{2\pi x_\theta}{\lambda} + \varphi\right) \tag{4.10}$$

$$x_\theta = cos(\theta)x + sin(\theta)y \tag{4.11}$$

$$y_\theta = -sin(\theta)x + cos(\theta)y \tag{4.12}$$

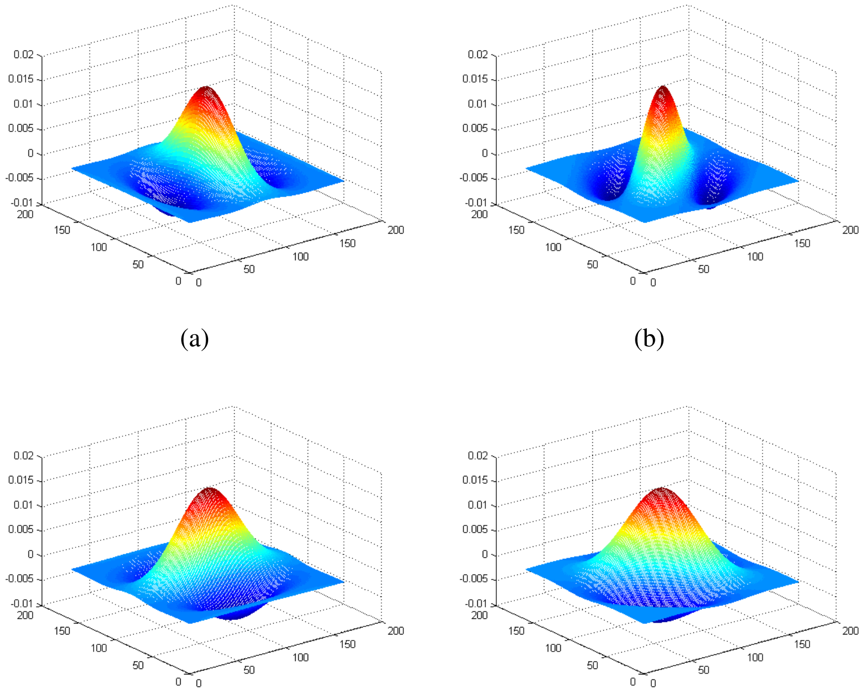
$$b = \log_2 \frac{\frac{\sigma}{\lambda}\pi + \sqrt{\frac{\ln 2}{2}}}{\frac{\sigma}{\lambda}\pi - \sqrt{\frac{\ln 2}{2}}} \tag{4.13}$$

$$\theta_{GB} = argmax_\theta(I_{(x,y)} * K_\theta) \tag{4.14}$$

4.1.3 Evaluation of the image filters

According to the filter kernels described above, the gradient-type filters are easy to implement but need an additional step to calculate edge orientation (the cross angle between edge and horizontal axis of an image in this thesis) from gradient. One of the obvious advantages of directional filters is the orientation can be inferred directly from the response results. Our aim is to select a image filter with reliable 2D orientation detection and reasonable computation complexity. This section designed several tasks for choosing a preferred image filter applied to hair detection purpose from the aforementioned candidates.

To evaluate pre-processing performance, the calculation results on both synthetic image and real hair images are compared. Four filters mentioned above will firstly convolve with a synthetic image with a known edge direction to test the accuracy. Those filtering methods on real hair images are also tested.



(a)

(b)

(c)

(d)

Figure 4.6: *Gabor filter with different θ: 4.6(a) θ = 0°, 4.6(b) θ = 45°, 4.6(c) θ = 90°, 4.6(d) θ = 135°.*

Test with synthetic image The synthetic image consists of quarter concentric circles rendered under a preset illumination directions from the top. The known edge direction is perpendicular to the radius of those concentric circles within $\left[0, \frac{\pi}{2}\right]$ (the angle with horizontal axis is used to demonstrate edge direction).

The gradient orientation matrices obtained by Sobel filtering, Gaussian derivative filtering and steerable filtering are transferred into edge orientation to compare with the ground truth. According to the filtering results in figure 4.7, Sobel filter only captured the pixel with high contrast, failed to extract detail information. As it can be seen from the zoomed in pictures, Gaussian derivative filter, Steerable filter and Gabor filter are all successfully highlighted the possible edge location.

A vector map is used to indicate edge direction (figure 4.8). The vector map shows that Gaussian derivative filter has the best results of orientation estimation from gradient direction calculation. Steerable filter and Gabor filter has small errors due to the choose of orientation range. This experiment sets $\theta \in \left[0, \frac{\pi}{2}\right]$. The Gabor filtering results are very sensitive to the parameters, the selection of them has been explained in Petkov (1995).

Test with hair image Both filtering results and colour mapped hair orientation are presented in Figure 4.9. Color mapped orientation field provides the directly intuitive

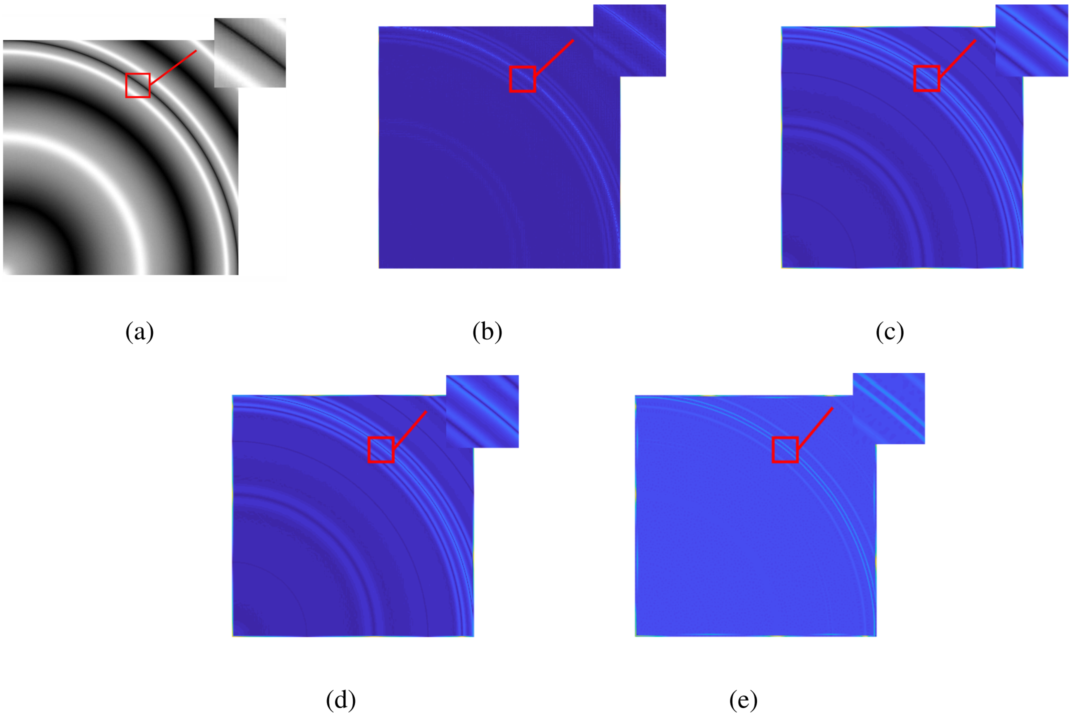

(a)
(b)
(c)
(d)
(e)

Figure 4.7: *Synthesised test image 4.7(a), the Sobel filtering result 4.7(b), the Gausssian derivative filtering result 4.7(c), the Steerable filtering result 4.7(d) and the Gabor filtering result 4.7(e).*

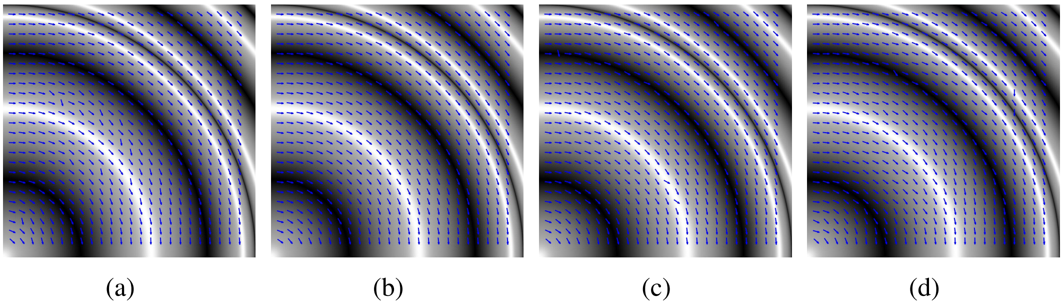

(a)
(b)
(c)
(d)

Figure 4.8: *Calculated vector map by different image filtering results, Sobel operator 4.8(a), Gaussian derivative filter 4.8(b), Steerable filter 4.8(c) and Gabor filter 4.8(d).*

feelings. All the colour plates vary between $[0, \pi]$ in this experiment. The shadowing areas are also zoomed in the original image and filtering results.

An image as a taken photo has inevitable noise due to the lighting condition or motion blur. Sobel is a simple gradient operator, which is sensitive to noise. Although it performs well on the synthetic picture, it is not recommended to use it to process hair image with a low resolution and the results may prone to noise incurred error.

The Gaussian derivative filter and Steerable filter are both Gaussian based filters, they get more smooth and continuous filtering results, which are similar in the patterns(as seen in Fig. 4.9).

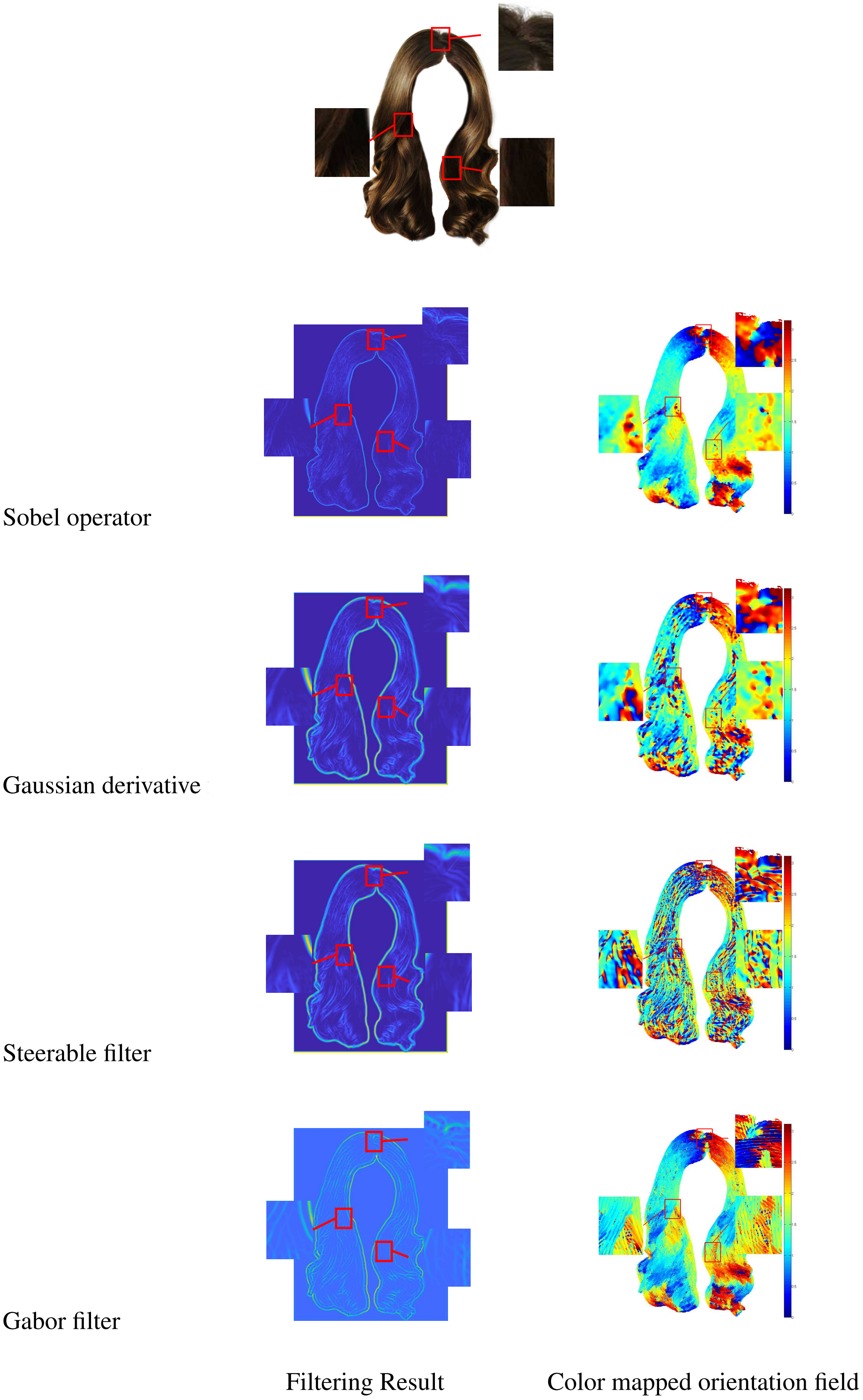


Figure 4.9: *Real hair image filtering results and color mapped hair orientation*

In a picture with a relatively low resolution, a hair fibre is much smaller than a pixel width with introducing unwanted aliasing. The Gaussion filter may over smooth the results. The standard deviation σ is set as 3.0 for both Gaussian derivative filter and Steerable filter. Because Gabor filter is a multi scaled filter, the hair width can be fitted by adjusting the wavelength λ. This case assigns $\lambda = 10$, $\varphi = 0$, $\gamma = 0.5$ and bandwidth $b = 1$. Although λ value is specified in pixels, it still needs to keep the balance between details and noise.

On the contrary, Gabor filter has the advantages of the invariant properties of the extracted features (Kamarainen *et al.*, 2006). It recovers the missing details while other approaches failed. The Gabor filter can even deal with extremely curved hairstyle, see Fig 4.10.

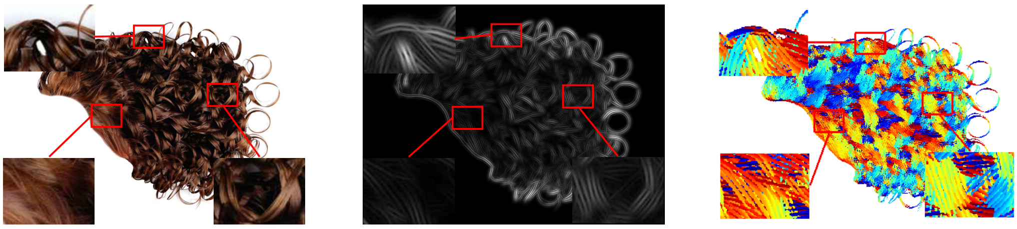


Figure 4.10: *Gabor filtered curve hairstyle and color mapped orientation field*

Performance comparison Table 4.1 and table 4.2 compares their running time and presents performance scores respectively. In table 4.2, 4 factors: easy of implementation and use, less sensitive to noise, computing speed, and direct estimate hair orientation are used to evaluate the applicability of hair orientation detection for the selected filters. The performance score is based on the experiment results in this thesis.

Sobel filter is a super fast operator with no question. Its running time keeps within millisecond even in processing a large image. The Gaussian derivative filter also performs well on computing speed. But both of them are sensitive to noise. For some realtime interactive applications, they might be good choices as long as introducing some result enhancements. Steerable filter and Gabor filter need calculate all the directions of the user defined orientation range to select optimal response. Therefore, their computing speeds depend on the number of directions. In this case, 90 directions evenly ranged from 0 to 180°. For those off-line applications with high requirements and directly orientation estimation, Gabor filter is satisfied overall. In chapter 6, the pre-processing results are applied for bas-relief stylisation of hair from image input.

Image quality	Sobel Filter	Gaussian derivative filter	Steerable filter	Gabor filter
128*128	0.347ms	3ms	82.6ms	168.2ms
512*512	3.9ms	13ms	1.1856s	2.6503s
1024*1024	17.4ms	40.6ms	4.5036s	11.7213s

Table 4.1: *Running time on different image qualities*

	Sobel Filter	Gaussian derivative filter	Steerable filter	Gabor filter
Easy of implementation and use	⋆⋆⋆⋆⋆	⋆⋆⋆⋆	⋆⋆⋆⋆	⋆⋆
Less sensitive to noise	⋆⋆	⋆⋆⋆	⋆⋆⋆	⋆⋆⋆⋆⋆
Computing speed	⋆⋆⋆⋆⋆	⋆⋆⋆⋆	⋆⋆⋆	⋆⋆
Direct estimate hair orientation	-	-	-	√

Table 4.2: *Filter performance score*

4.2 2D hair strands generation

With aforementioned image pre-processing methods, an estimated 2D orientation map by Gabor filtering is obtained. Next the seed points will be selected to grow hair strands according to the estimated orientation map.

In order to reduce the effects of noise, a confidence map has been calculated to iteratively refine orientation map by following existing methods (Paris *et al.*, 2004; Chai *et al.*, 2012). The definition of the confidence map is:

$$conf(x,y) = \sqrt{\sum_{i=1}^{N}(D(\theta_i,\tilde{\theta}) \cdot (R(\theta_i) - R(\tilde{\theta}))^2)}, \tag{4.15}$$

where N equals the numbers of Gabor filters, $D(\theta_i,\tilde{\theta})$ is the minimum distance between current orientation and the orientation with corresponding max filtering response, $R(\theta_i)$ is the image filtering response with orientation i and $R(\tilde{\theta})$ is the max response. Sometimes, unreliable pixel might have a false high confidence due to imperfection inputs. As suggested in Chai *et al.* (2012), iteratively filtering confidence map can effectively generate an accurate orientation map.

Next seed points are sampled regularly within the region of hair to cover the whole hairstyle. Similar to previously methods, each time tracing hair strand on both possible directions along the estimated orientation from a picked seed point. Thus the orientation map is converted into a set of individual hair strands. The pseudo code of the tracing process is described as algorithm 4.1, where C is the confidence map, c_{low}

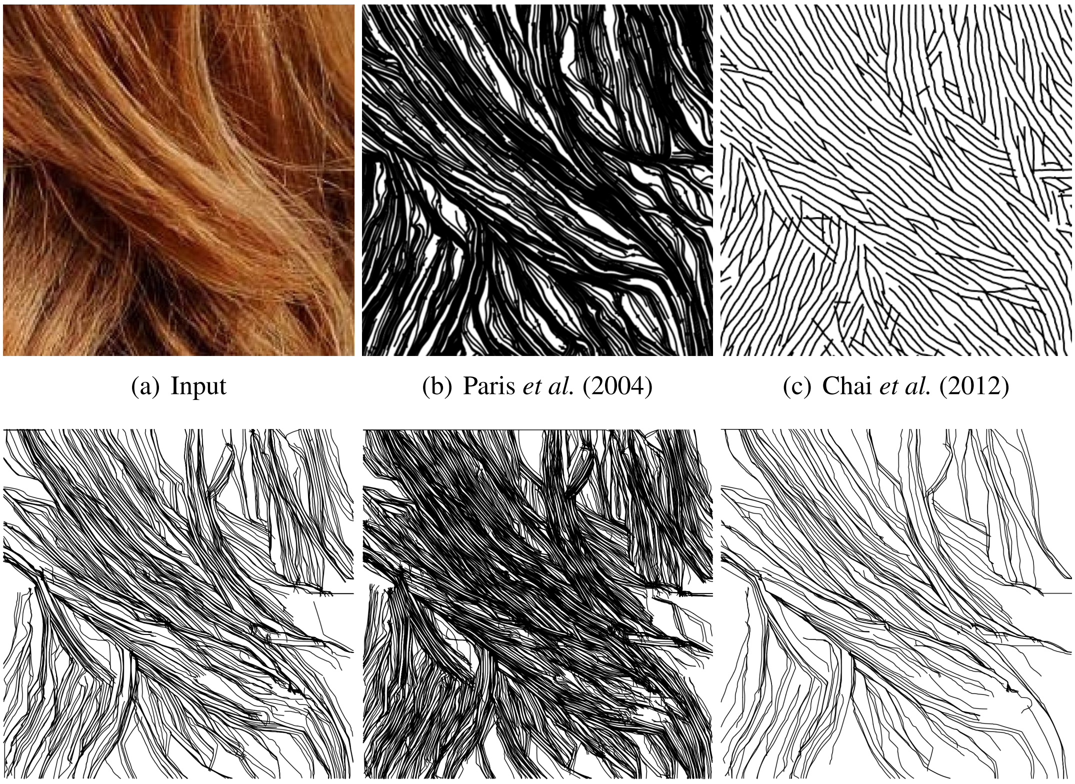
is the confidence thresholding value and $v(\vec{p_i})$ is the hair growth direction of point p_i.

Algorithm 4.1 Tracing hair strand from seed point p_i

```
procedure HAIR_TRACING($p_i$, $C$, $step$)
    $status \leftarrow certain$
    while $safe\_point > 0$ do
        if $status = certain$ then
            if $C(p_i) < c_{low}$ then
                $status \leftarrow uncertain$
                $v(\vec{p_i}) \leftarrow v(\vec{p_{i-1}})$
                $safe\_point - 1$
            end if
            $p_{i+1} = p_i + step * v(\vec{p_i})$
        else
            if $C(p_i) > c_{low} \parallel arccos(v(\vec{p_i}) \cdot v(\vec{p_{i-1}})) < \theta_{max}$ then
                $status \leftarrow certain$
                $safe\_point \leftarrow 5$
            else
                $v(\vec{p_i}) \leftarrow v(\vec{p_{i-1}})$
                $safe\_point - 1$
            end if
            $p_{i+1} = p_i + step * v(\vec{p_i})$
        end if
    end while
end procedure
```

To tailor this tracing algorithm for the structure-aware framework, the c_{low} in Algorithm 4.1 is adjusted to emphasise the possible hair wisps. With carefully observation, there is a tendency of forming a small dark region when occlusion happens between two wisps due to lack of illumination of the back layer. Increasing the value of c_{low} is able to remove the unreliable hair strands traced within the region and highlight the wisps. This thesis sets a relatively high value of 1.0 for c_{low} compared to the settings of Chai *et al.* (2012). Correspondingly, the traced 2D hair strands are slightly different from Chai *et al.* (2012); Paris *et al.* (2004) for the structure analysis purpose. Figure 4.11 illustrates the traced 2D curves comparison between this thesis and others.

Here the hair strands generation methods on a set of different hairstyles ranging from long straight hair to extremely curve hair have been tested. The results from Figure 4.12 demonstrates the ability of Gabor filter in building reliable orientation map. Referring to the generated hair strands in the second row, most of them represented their positions in the original picture.



(a) Input

(b) Paris *et al.* (2004)

(c) Chai *et al.* (2012)

(d) Evenly sampling with average seeds

(e) Evenly sampling with dense seeds

(f) Evenly sampling with sparse seeds

Figure 4.11: *Traced 2D curves comparison of Paris* et al. *(2004), Chai* et al. *(2012) and this theis. 4.11(a): input hair image, 4.11(b) and 4.11(c): traced 2D curves by Paris* et al. *(2004) and Chai* et al. *(2012) respectively, 4.11(d), 4.11(e) and 4.11(f): traced 2D curves by the method proposed in this thesis with average seeds, dense seeds and sparse seeds. Most of the experiments conducted in this thesis using average seeds, which means placing seed point among every 5 pixels.*

4.3 2D hair sketch extraction

With the hair tracing method, the corresponding 2D hair strands have been obtained for the input portrait. Although the traced hair strands reveal the hairstyle appearance of the original image, there are still inevitable artifacts caused by the noises. In addition, the current shape of hair strands is expressed by 2D line segments, which is not an idea structure for us to analyse the geometric information of hairstyle. Thus, a higher layered structure is required for extracting meaningful hairstyle elements.

Most of the state-of-the-art image based hair modelling methods aim to recover realistic 3D shape of hair. However, some important issues have been ignored somehow, such as reconstructing reliable hairstyle from the basic shape structure is one of the topic. 2D hair sketch from the observed hair strands is extracted, which will be used to form the fundamental structure to generate 3D hairstyle in chapter 5.

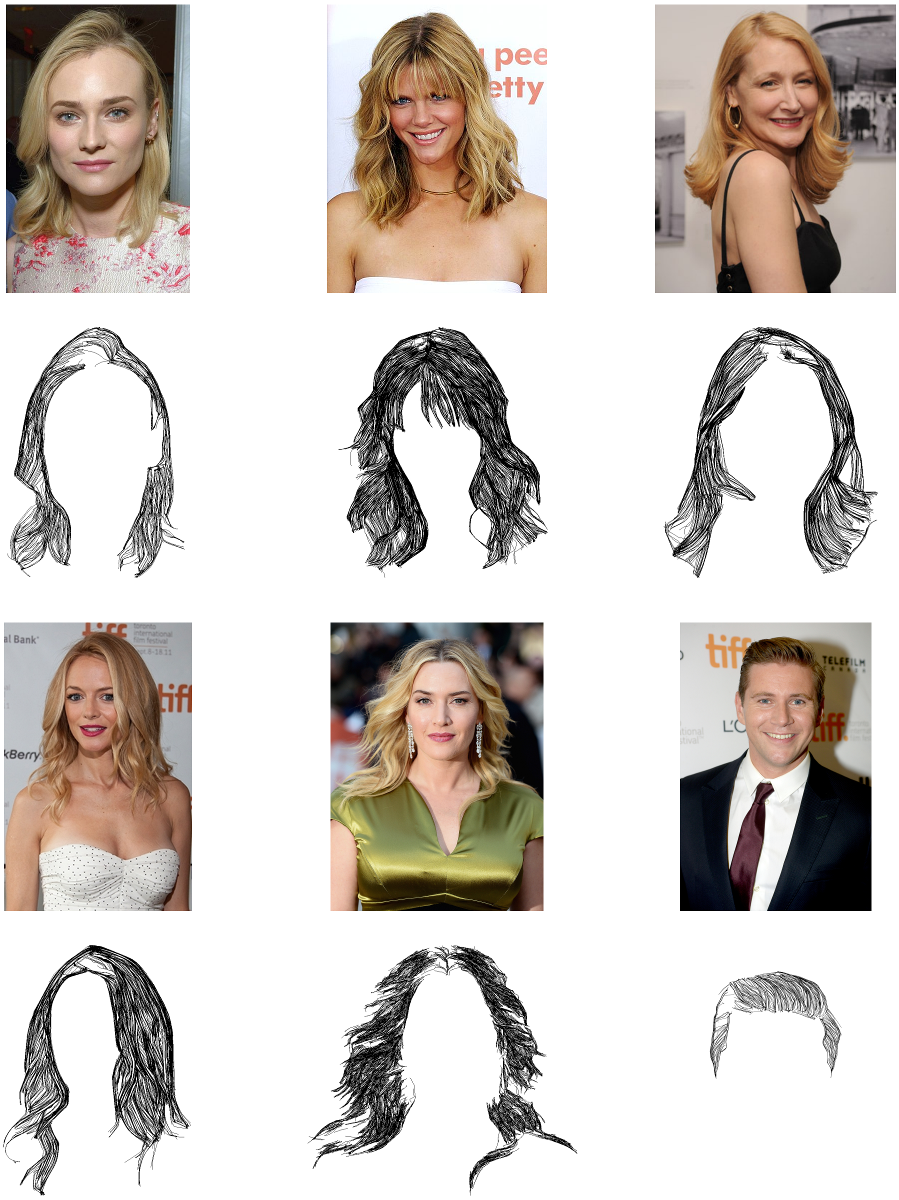
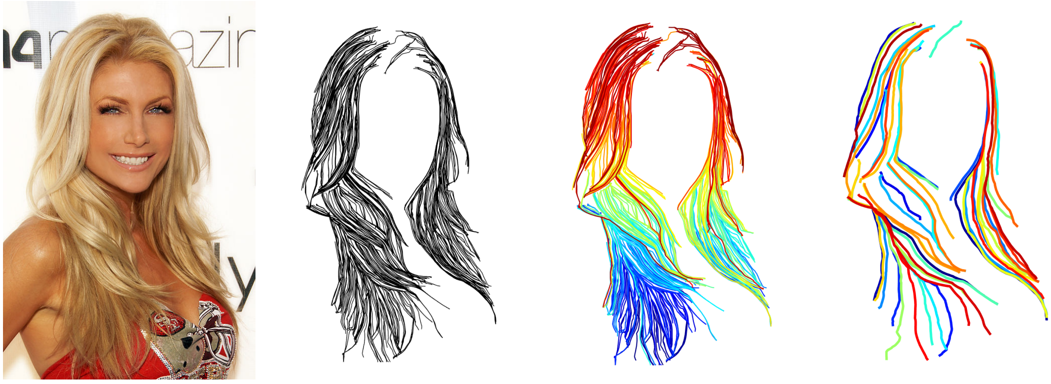


Figure 4.12: *Traced 2D hair strands*

4.3.1 Hair strands connection

In the case of bad illumination or where hair wisps occlusion occurred, a whole hair strand will be cut into several line segments when doing hair tracing on the image plane. Apparently, these isolated line segments should be connected together to form a complete hair strand. Here, an asymmetric distance between any two segments is denoted as:

$$d_c(s_i, s_j) = \begin{cases} d\left(p_i^t, q_j^h\right) & \text{if } d\left(p_i^t, q_j^h\right) < \epsilon \;\&\; \sigma < \frac{\pi}{6} \\ \infty & \text{otherwise} \end{cases}, \sigma = |\theta_{p_i^t} - \theta_{q_j^h}| \tag{4.16}$$



(a) (b) (c) (d)

Figure 4.13: *2D hair strands connection*

where p_i^t, q_j^h denote the tail and head nodes of segment s_i and s_j respectively. $d\left(p_i^t, q_j^h\right)$ is the Euclidean distance, which means the distance cost if attaching the head of s_j to the tail of s_i. The threshold value ϵ is used to filter out those pairs that are too far to be connected together. σ is the orientation difference between p_i^t and q_j^h to enforce a hair strand holding the natural bending angle.

With equation 4.16, the asymmetric distance matrix $\mathbf{D} = (d_c(s_i, s_j))$ can be obtained over the head and tail nodes. These segment pairs are connected by searching the order from the minimal distance value to the higher. Once meeting the end of this searching, such that no extra segment can be connected to this strand, then delete the indexes of these involved segments in matrix $\mathbf{D}$ and start a new searching until it turns to be empty.

4.3.2 Sketch extraction

The sketch is enormously used in 3D modelling to infer the object with the knowledge from database (Ding and Liu, 2016). In hair modelling, the user input strokes commonly play as the sketch of the hairstyle to infer its geometry shape (Wither *et al.*, 2007; Hu *et al.*, 2014, 2015). To automatically outline the hair sketch from 2D image, Chen and Zhu (2006) treat the hair image as 2D piecewise smooth vector fields. The hair sketch is the representation of the discontinuities in the vector field. Although the synthesised sketch is very plausible, it fails to highlight the inside hair shape details, as many of them have been smoothed out. This thesis requires that the 2D hair sketch reveals the hair structure without loosing too much shape details. To achieve this goal, the same K-means++ algorithm in section 3.2.2 is applied on the refined 2D hair strands to cluster them into groups with similar shape. The collection of the cluster centres is the 2D hair sketch. 10 is set as the initial number of clusters to investigate the sketch. The result shows that 10 is not enough for full-head hairstyle recover. Therefore 25, 50 and 75 clusters have been tested and 50 came out as a reasonable trade-off between reconstruction satisfaction and computation efficiency.

4.4 Summary

In order to capture the 2D hair strands of the input image, this chapter first introduced the image matting method to segment the region of hair from a portrait image. Afterwards, four image filters have been examined, all of them are designed for the edge detection purpose. According to their performances on 2D hair orientation map extraction, the Gabor filter is believed as a reliable pre-processing tool for 2D hair strands capturing. To decrease the affection of the image noise, a confidence map is calculated to iteratively refine the orientation map. Thus the 2D hair strands have been traced for the structure analysis, which benefit from an enhanced orientation map. Instead of directly manipulating the 2D hair strands to pop-up an incomplete 3D plausible hairstyle (Chai *et al.*, 2012), this thesis cluster the hair strands into groups with similar shapes and extract the centres of them as the 2D hair sketch. This 2D hair sketch plays as the coarse initial hairstyle structure which participants into a hair shape descriptor based hair modelling method in following chapters.

Chapter 5

The Struct2Hair: HSD based 3D hairstyle generation

Unlike modelling human bodies with fixed anatomical features, to generate a realistic hairstyle model from single-viewed image is complicated. Due to the large numbers of hair strands with intrinsic shapes, there is almost no rules to describe the structure of a hairstyle. In order to solve this problem, Hu *et al.* (2015) and Chai *et al.* (2016) applied data-driven approach to realize hair model generation from single image. Different with their research, our project aims to investigate the fundamental structure of a hairstyle, by proposing a Hair Shape Descriptor to capture the hairstyle's skeleton. In this Chapter, the Critical Hair Shape database is used to guide the transition from the extracted 2D hair sketch to the HSD. The complete hair model is reconstructed from the HSD.

5.1 2D hair sketch extension

In last Chapter, the generated 2D hair strands didn't reveal the missing information from the original image. Consider the hairstyle is a combination of different hair wisps, hair strands grown from the back side of the head also participant to add variation to the visible hair. Thus, the extracted hair sketch is extended to be attached to the scalp of the head aligned to the portrait.

Figure 4.13(c) shows that some extracted strands are occluded by the front hair, which partly appear in the current view. To recover the occluded part for these strands, the extracted hair sketch is extended with the following steps to let them grow from hair roots in 2D. In USC-HairSalon database, all the hairstyles are aligned with a head model provided by Hu *et al.* (2015). In the following work, in order to do the 2D shape centres matching, the critical hair shape centres are projected to the 2D portrait plane through fitting the head model to the face of the portrait by the pose estimation and shape fitting method in Huber *et al.* (2016).

1. Pose estimation & Head fitting
   (a) Detect 2D facial landmarks of input portrait, and manually label their correspondences on the given standard head model.
   (b) Estimate the pose of the standard head model by solving camera matrix using the Gold Standard Algorithm (Hartley and Zisserman, 2004).
   (c) Rotate and scale the standard head model by obtained camera matrix to make its face aligned to the portrait.
2. Attach 2D hair sketch
   (a) Project the 3D hair roots on the standard head model to the 2D image plane.
   (b) Search the nearest hair root for each strand in the 2D hair sketch, and use polynomials to fit these strands along with their nearest hair root, which ensures that the occluded strands are attached to the hair roots on the back head.

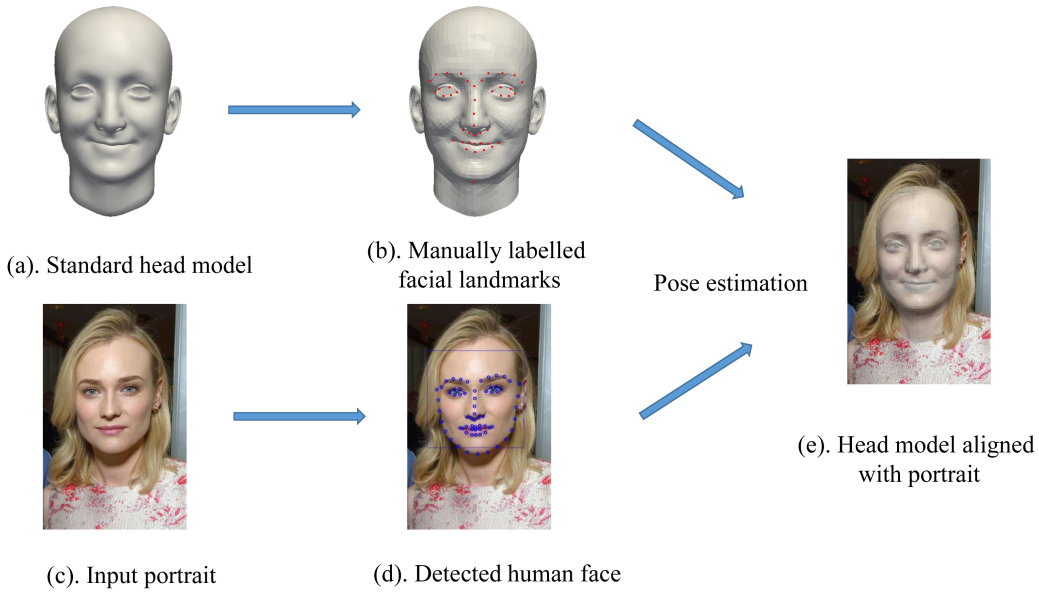


Figure 5.1: *Pose estimation & Head fitting. The input portrait is chosen from the IMDB-WIKI dataset (Rothe* et al.*, 2016).*

Pose estimation & Head fitting. To accomplish the estimation of head pose or camera position, the prior task is to find the approximation of camera matrix by given a number of 2D-3D point pairs. The 3D facial landmarks are manually labelled on the head model first. The corresponding 2D landmark points of the portrait are detected by the OpenCV face detector. In Huber *et al.* (2016), they provide a C++ library to calculate the camera matrix **C**. When the prepared 2D-3D point pairs are ready, one can just load and run the function in their library to obtain the **C**:

```
vector<cv::Vec2f> portrait_points = ...;
vector<cv::Vec3f> head_model_points = ...;
Mat affine_camera = estimate_affine_camera(
    portrait_points, head_model_points);
```

This process is shown in Figure 5.1. In this project, the pose estimation is used to project critical hair shape centres onto the image plane. Therefore, only rotation, transition and scale have been considered when doing pose estimation without shape deformation. The detected silhouette points in Figure 5.1(d) has been removed to avoid face deformation. Our **C** contains a rotation matrix R and a translation vector **t**. There is a scale factor S to set the head to a proper size.

Attach 2D hair sketch. In order to attach 2D hair sketch onto hair roots, the 3D roots have been projected onto the image plane by camera matrix first. Then for each stroke of the 2D hair sketch, add the closest root as the first point to the stroke. To avoid simply attaching the sketch to form rigid stroke, a piecewise polynomial is used to fit and smooth the strand. The connected line segment from the nearest hair root to the first point of the hair sketch is fitted by a second-order polynomial. For the curly hairstyle, a higher ordered polynomial is applied to preserve the original curve shape instead of over smoothing. For an extreme case, a fifth-order polynomial is enough to ensure a reasonable performance. Figure 5.3(a) displays that those occluded hair strands in Figure 4.13(d) are connected to the hair roots, which are used to add the variation of the hair in following steps.

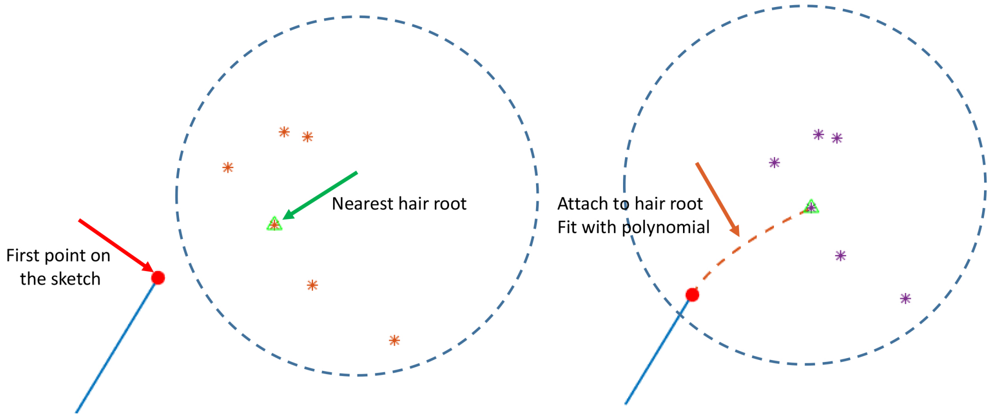
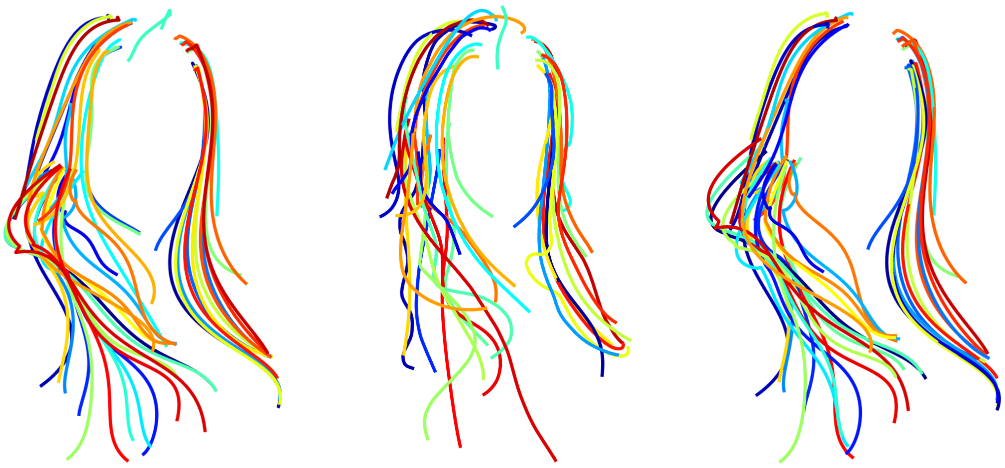


Figure 5.2: *Connect the sketch to its nearest hair root. Fit with polynomial to avoid rigid line segment.*



(a) Attached 2D hair sketch (b) Matched CHS centres (c) Generated HSD

Figure 5.3: *Attached 2D hair sketch to generate HSD.*

5.2 HSD generation

In this thesis, the hair shape descriptor (HSD) is the representative structure of a hairstyle model. Once the 2D hair sketch has been attached to the projected hair roots, there will need a method to recover its depth for HSD generation. This section will build the HSD from 2D sketch under the guidance of critical hair shape database.

For each 2D cluster centre, search the closest critical hair shape from the table maintained for the hair shape database according to the shape and distance constraints. Therein, this thesis firstly project every entry of the table to the 2D image plane based on the pose estimation process. Then the distance metric in equation 3.3 is used for matching the closet 2D critical hair shape centre to each strand of hair sketch. The matched 3D critical hair shape centre is shown in Figure 5.3(b).

This section builds frenet-serret frame for both 2D strand in hair sketch and its corresponding 3D critical hair shape centre. The depth of 3D critical hair shape centre is then applied to guide the generation of 3D hair sketch. This 3D hair sketch is our hair shape descriptor (HSD), which plays as an important structure feature to reconstruct full-head hairstyle generation in section 5.3. For each pair of 2D strand and matched hair shape centre, the corresponding parametric curve functions can be built as follows.

$$\left\{ \begin{array}{l} x = x_1\left(s_1\right) \\ y = y_1\left(s_1\right) \end{array} \right. , s_1 \in \left[0, L_1\right], \tag{5.1}$$

$$\begin{cases} x = x_2\left(s_2\right) \\ y = y_2\left(s_2\right) \\ z = z_2\left(s_2\right) \end{cases}, s_2 \in \left[0, L_2\right], \tag{5.2}$$

where $s_{i,\ i\in[1,2]}$ represents the curvilinear abscissa along the curve i. x, y and z are the Cartesian coordinates of the nodes on a curve. the x-y plane is coplanar with the portrait plane. For 2D hair sketch, the coordinates along z-axis (depth direction) are zeros. x_1, y_1, x_2, y_2, z_2 are the parametric functions which are approximated by the cubic spline functions. L_1 and L_2 are denoted as the total arc length of the two curves.

Afterwards a linear mapping from s_1 to s_2 is taken as:

$$s_2 = \frac{L_2 s_1}{L_1}, s_2 \in \left[0, L_2\right], \tag{5.3}$$

and transfer the depth information to the 2D hair sketch, then each 3D strand of the HSD can be represented by

$$\begin{cases} x = x_1\left(s_1\right) \\ y = y_1\left(s_1\right) \\ z = z_2\left(\frac{L_2 s_1}{L_1}\right) \end{cases}, s_1 \in \left[0, L_1\right]. \tag{5.4}$$

A generated HSD is shown in Figure 5.3(c).

5.3 Full-head hairstyle generation

Based on the HSD obtained above, full-head hairstyle is generated by treating the whole hairstyle as a 3D orientation field. All the hair strands are grown on a predefined region on the scalp with the evenly distributed hair roots (10000 hair strands have been generated for the full-head hairstyle in this thesis).

Before that, two priori rules are proposed for generating hair from HSD.

1. For strands whose hair roots are located within a close distance, their strand length are similar, except some special areas (e.g. the fringe hair).
2. The orientation vectors of neighboring nodes on two strands (attached together) are similar in common cases.

When doing full-head hair generation, the HSD represents the shape information as a collection of local hair details. Each strand of HSD is treated as an initial hair cluster centre to diffuse the entire hairstyle around the scalp. It adopts an iterative hair generation method to keep local hair details when generating new hairs. New hair strands with close distance to HSD will be grown first, then expanded until full-head strands generation is completed. The main steps of the hair generation algorithm are shown as following:

Algorithm 5.1 Full head hair strands diffusion

```
1: Build a dataset $\boldsymbol{X} = \{\boldsymbol{x}_i\}, \boldsymbol{x}_i \in \mathbf{R}^3$ to store all the hair roots and a dataset $\boldsymbol{S} = \{S_i\}$
   to store the current strands, which is initiated by strands in HSD.
2: For each strand $S_i \in \boldsymbol{S}$, search $m$ nearest hair roots in $\boldsymbol{X}$.
3: Collect all the $m$ nearest hair roots, and generate strands based on their neighbour
   strands in $\boldsymbol{S}$.
4: Update the current strands set $\boldsymbol{S}$ by appending the generated strands, and remove
   their hair roots from $\boldsymbol{X}$.
5: Repeat step 2 - 4 until $\boldsymbol{X}$ is empty, $\boldsymbol{S}$ is the generated full-head hair.
```

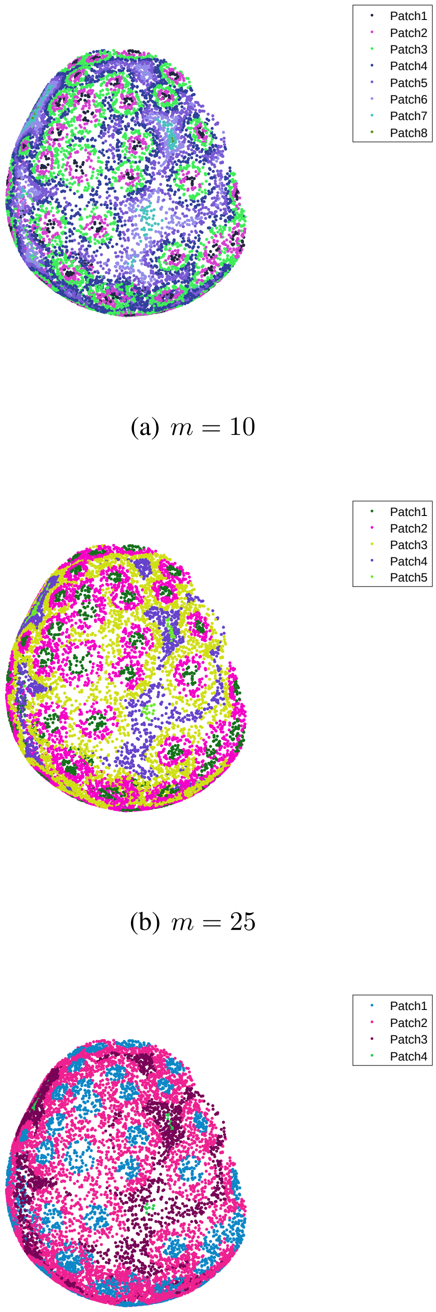
The number of new generated strands in step two is increasing with the expanding of $\boldsymbol{S}$, which ensures that the local shape information of the original centres is kept while the shape of new generated strands between these centres varies smoothly. Each time, m nearest hair roots are searched for the current hair strands in $\boldsymbol{S}$ for new hair strands generating, this patch hair growth process is demonstrated in Figure 5.4. This will also be reflected in the local generation step, please see Figure 5.5 for the details of hair strands diffusion.

Taking the first hair generation step as an example, given a set of hair roots and initial hair strands from HSD, there are two phases in generating the new strands: deciding the strand length and orientation vectors.

5.3.1 Strand length

Based on the first priori rule, for each hair root, the k-NN algorithm is used to search m neighboring hair roots among these centres, then adopt the Multi-quadric (M-Q) radial basis function (Cheng, 2012) to interpolate its strand length based on the length data of these centres. k-NN method used here is aimed at keeping the local length and avoiding a large coefficient matrix in M-Q interpolation (see the matrix $(\phi(r_{i,j}))$ in 5.7). The form of M-Q radial function is:

$$\phi(r) = (-1)^{[\beta]} \left(r^2 + c^2\right)^{\beta}, c \geq 0, \beta > 0, \beta \notin \mathbf{N}, \tag{5.5}$$

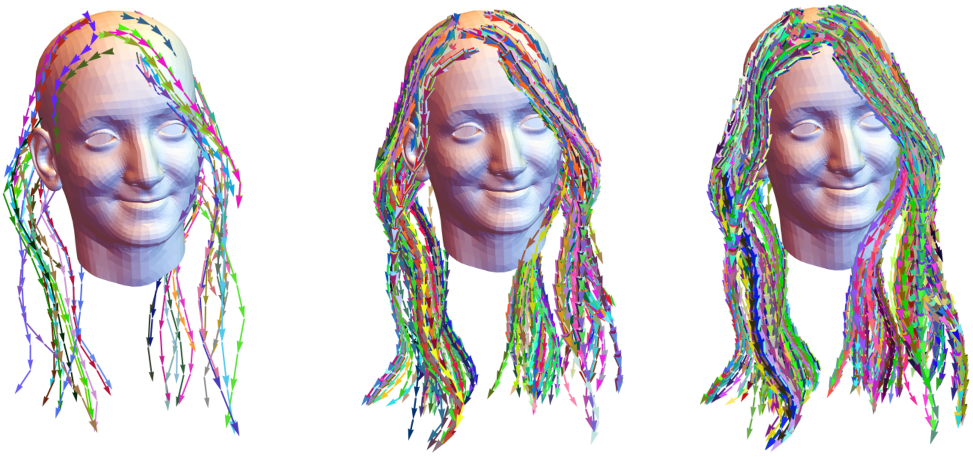


(a) $m = 10$

(b) $m = 25$

(c) $m = 50$

Figure 5.4: *Choose hair roots for full head hair generation by patch with different* m. *5.4(a) full head hair generated by 8 patches with* $m = 10$, *5.4(b) is full head hair generated by 5 patches with* $m = 25$, *5.4(c) is full head hair generated by 4 patches with* $m = 50$. *Color labelled hair roots to illustrate different patches.*



(a) Initial 50 strands (HSD)

(b) 425 strands

(c) 1218 strands

Figure 5.5: *Increasing hair strands around scalp*

and the interpolated strand length function is written as:

$$l\left(\boldsymbol{x}\right)=\sum_{i=1}^{m}a_i\phi(r),\ r=||\boldsymbol{x}-\boldsymbol{x}_i||. \tag{5.6}$$

In equation 5.5, the parameters β and c are used to control the shape of M-Q function where Cheng (2012) is recommended for their definitions. r is the variable of the radial Euclidean distance between two points. $[\beta]$ denotes the ceiling function. **N** is the natural number. In equation 5.6, a_i is the interpolation coefficient. $\boldsymbol{x}$ is the 3D position of the hair root for interpolating, and $\{\boldsymbol{x}_i\}$ is the closest m neighbour centres. Since the 3D position of every centre's hair root $\{\boldsymbol{x}_1,\boldsymbol{x}_2,...,\boldsymbol{x}_m\}$ and their strand length $\{l\left(\boldsymbol{x}_1\right),l\left(\boldsymbol{x}_2\right),...,l\left(\boldsymbol{x}_m\right)\}$ are calulated, the coefficient $\{a_i\}$ can be obtained by solving

$$\begin{pmatrix} l\left(\boldsymbol{x}_1\right) \\ l\left(\boldsymbol{x}_2\right) \\ ... \\ l\left(\boldsymbol{x}_m\right) \end{pmatrix}=\begin{pmatrix} \phi\left(r_{1,1}\right) & \phi\left(r_{1,2}\right) & ... & \phi\left(r_{1,m}\right) \\ \phi\left(r_{2,1}\right) & \phi\left(r_{2,2}\right) & ... & \phi\left(r_{2,m}\right) \\ ... & ... & ... & ... \\ \phi\left(r_{m,1}\right) & \phi\left(r_{m,2}\right) & ... & \phi\left(r_{m,m}\right) \end{pmatrix}\begin{pmatrix} a_1 \\ a_2 \\ ... \\ a_m \end{pmatrix}, \tag{5.7}$$

where $r_{i,j}=||\boldsymbol{x}_i-\boldsymbol{x}_j||$.

Fringe hairs

The strand length of the new strand can be figured out by giving the 3D position of its hair root in equation 5.6. However in some special scenarios, especially the fringe hairs, there is a length gap between them and their neighbours. Since the M-Q function is continuous, it will make the hair at the boundary of the gap a little longer than it suppose to be. In this case, a pre-defined area for fringe hair should be given in advance to ensure the length of new hair strands located in it can only be affected by the hair strands within this area. In this thesis, the shape of the fringe hairs are controlled by two parameters, the ratio of fringe hair roots and the proportion of front hair length. The ration indicates how much hairs will account for the fringe area. The proportion adjusts the front hairs length to make them have a natural look. For the hairstyles with a fringe hair, the special area is set as 20% of the whole scalp. A hairstyle with a choppy full fringe hairs is used to demonstrate how these two parameters work, see Table 5.1 for a detailed discussion with fringe hair editing.

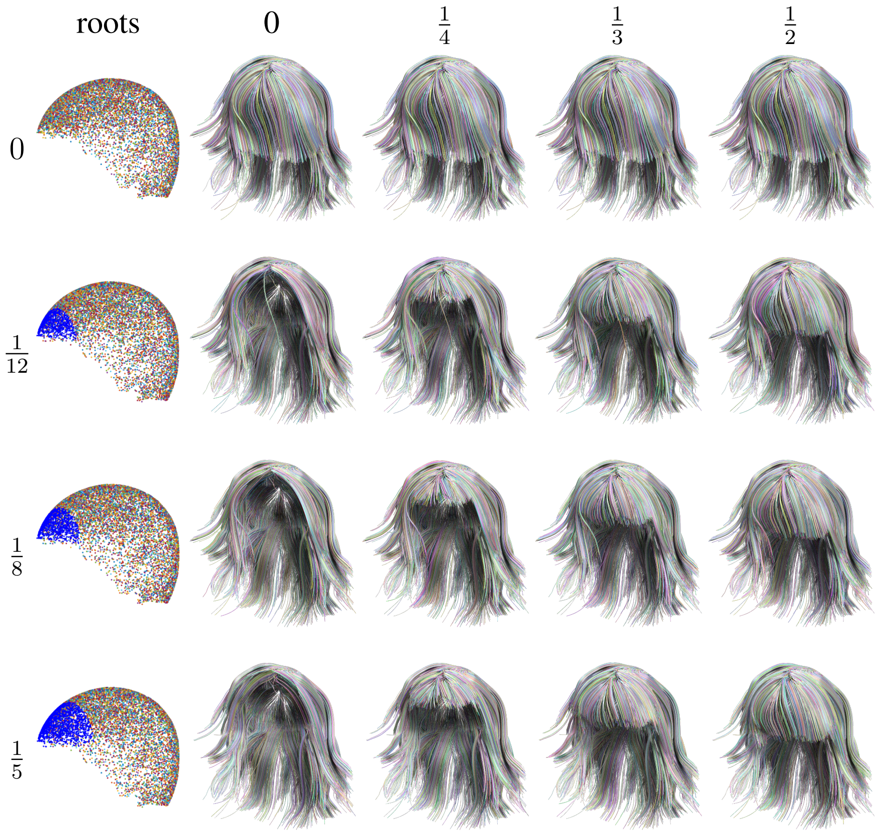


Table 5.1: *Parameters to edit the shape of fringe hairs. The parameters in top row are the proportions of the front hair length, which is used to cut the fringe hair. There is no fringe hair when length proportion equals 0. The parameters in the first column is the ratios, which indicate how much front hairs will account for fringe area. When both ratio and length proportion equal 0, the hairstyle is the original generated from HSD. For ensure the natural look of the hairstyle, ratio is set to* $\frac{1}{5}$ *and the length proportion is set between* $\frac{1}{3}$ *and* $\frac{1}{2}$*.*

5.3.2 Orientation vectors

After figuring out the length of this new strand, based on the second priori rule, the entire curve is constructed by integrating along its orientation vectors. Beginning with its hair root, by using the weighted k-NN regression, the orientation vector of a strand node is interpolated from the u neighbour nodes on other centre strands.

Denote the node on a strand as $\boldsymbol{r}\left(s\right)$, and its u neighbours from different centre strands as $\boldsymbol{r}\left(s_1\right), \boldsymbol{r}\left(s_2\right), ..., \boldsymbol{r}\left(s_u\right)$. $\boldsymbol{r}\left(s\right)$, $\boldsymbol{r}\left(s_i\right) \in \mathbf{R}^3, i = 1, 2, ..., u$. Then the orientation vector $\boldsymbol{r}'\left(s\right)$ can be solved by the weighted regression as :

$$\boldsymbol{r}'\left(s\right) = \sum_{i=1}^{u} \frac{\omega_i^2 \boldsymbol{r}'\left(s_i\right)}{\sum_{i=1}^{u} \omega_i^2}, \tag{5.8}$$

where $\omega_i = \left(\sum_{i=1}^{u} d_i\right) - d_i$,$d_i = ||\boldsymbol{r}\left(s\right) - \boldsymbol{r}\left(s_i\right)||$, d_i is the Euclidean distance between $\boldsymbol{r}\left(s\right)$ and $\boldsymbol{r}\left(s_i\right)$. $\{\omega_i^2\}$ are the weight values.

Starting from the hair root, iteratively calculating its orientation vector and updating the new node by $\boldsymbol{r}\left(s+\delta s\right) = \boldsymbol{r}\left(s\right) + \boldsymbol{r}'\left(s\right)\delta s$ until meeting the end of the strand length. Besides, to make the shape of strands vary smoothly between centre strands, this thesis adaptively change the searching range u in a new hair generation step by $u_{i+1} = 1.5u_i$, $u_0 = 1.5$.

5.4 Results

This section evaluates the hair modelling results from the following five aspects:

1. Import modelling results into computer modelling softwares to test the ease of portability.
2. Recover the hairstyle model in USC-HairSalon database to validate the framework.
3. Compare the generated hairstyle model with input portrait image to evaluate the modelling result.
4. Compare the modelling results between Struct2Hair framework and other state-of-the-art image-based hair modelling techniques to verify the reliability.
5. Indicate limitations.

There is no benchmark to examine the precise accuracy of single-view based hair modelling due to lack of the ground-truth data. This thesis involves original hairstyle model from a database as the ground-truth data to check the validation of the proposed framework. 3, 4 investigate the visual convincingness of modelling hairstyles, which are followed the latest measurements by Hu *et al.* (2014, 2015); Chai *et al.* (2016); Zhang *et al.* (2017). Detailed examinations are discussed below.

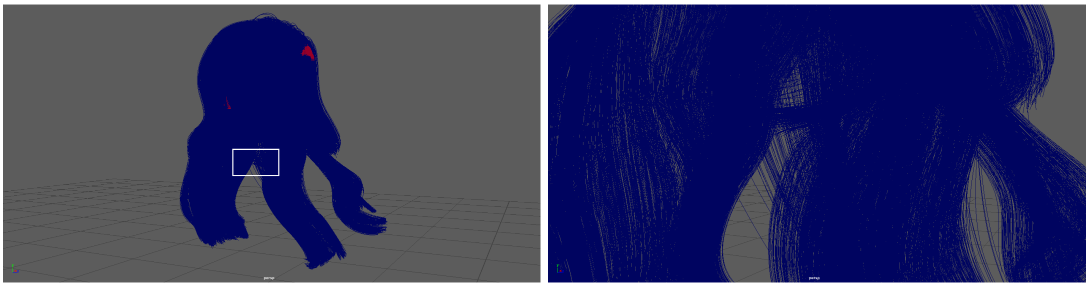
5.4.1 Import hair geometry into the computer modelling software

Maya and Houdini are selected as the example softwares to show the compatibility of the model. The reconstructed hairstyle model is a collection of hair curves, which are easily to import into the Maya or Houdini as a group of curves. For both Maya and Houdini, a hair curve is placed when all the hair points are appended as the control vertices to create a new curve.

In Maya, the hair curves are imported by using of the following MEL[1] commands:

```
curve -p hair_i(1,1) hair_i(1,2) hair_i(1,3)
curve -p hair_i(2,1) hair_i(2,2) hair_i(2,3)
......
curve -p hair_i(n,1) hair_i(n,2) hair_i(n,3)
```

i is the i^{th} hair curve. The imported hairstyle model is shown in Figure 5.6. For the animators or game designers, they can do further hair combing and styling on the basis of the reconstructed model. Or for common users, one can simply load the hairstyle model and attach default Maya hair material for photo-realistic rendering.



(a) Import a hairstyle model into Maya

(b) A close-up detail

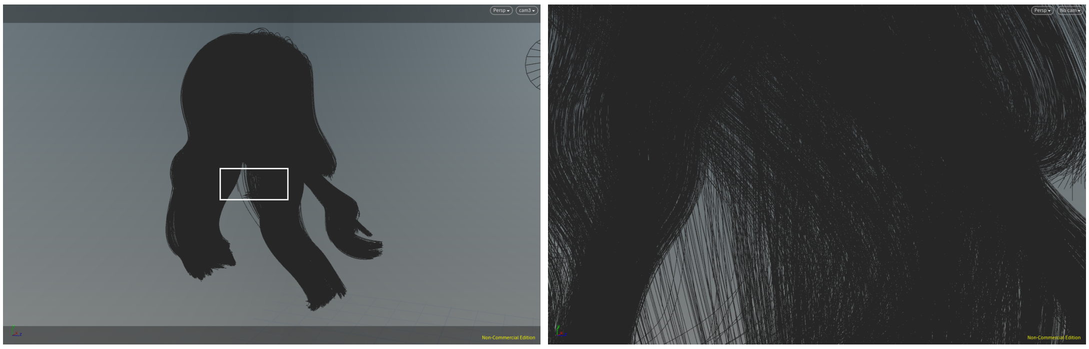
Figure 5.6: *Import a hairstyle model into Maya. 5.6(a) imported hairstyle and a close-up detail of it 5.6(b).*

In Houdini, the hair curves are added by the following pseudo Python scripting:

```
curve = geo.createNURBScurve(i)
for vertex in curve.vertices():
    vertex.point.setPosition((hair_i[0],hair_i[1],hair_i[2]))
```

[1] Maya commands, please see Beginning MEL scripting: exploring commands for further reading.

i is the i^{th} hair curve, same as the MEL script above. The imported hairstyle model in Houdini scene is demonstrated in Figure 5.7. In this thesis, Houdini is chosen as the rendering software for all the experiment results.



(a) Import a hairstyle model into Houdini

(b) A close-up detail

Figure 5.7: *Import a hairstyle model into Houdini. 5.7(a) imported hairstyle and a close-up detail of it 5.7(b).*

5.4.2 Comparison with ground-truth hairstyle model

The hairstyle models (as ground truth data) from USC-HairSalon database are used to evaluate the feasibility of the HSD based hair generation algorithm. The HSD of the ground truth hairstyle model in the database is the collected critical hair shape centres, which haven been described in Chapter 3. A straight short hairstyle model and a long wavy hairstyle model are selected to examine the algorithm performance.

Figure 5.8 demonstrates the HSD based hairstyle reconstruction results. The recovered short hairstyle model is almost the same as the original hairstyle model, with some slightly hair lengths differences compared to the ground truth data. For the long wavy hairstyle model, there are some over traced front hair strands. The length of a hair strand is decided by its nearest m neighbouring centres, which is the reason of causing undesired hair length. It can be improved by reducing the number of the neighbouring centres.

Additionally, as the full-head hair is fully automatically generated from HSD, which is found as a compact feature to save and reload the original hairstyles of the USC-HairSalon database. The results in Figure 5.8 proves its capability of recovering the ground-truth data. As a result, the USC-HairSalon database has been downsized from 3.53GB to 41.2MB by representing hairstyles of HSD. The downsized database is stored within a single Matlab .mat file.

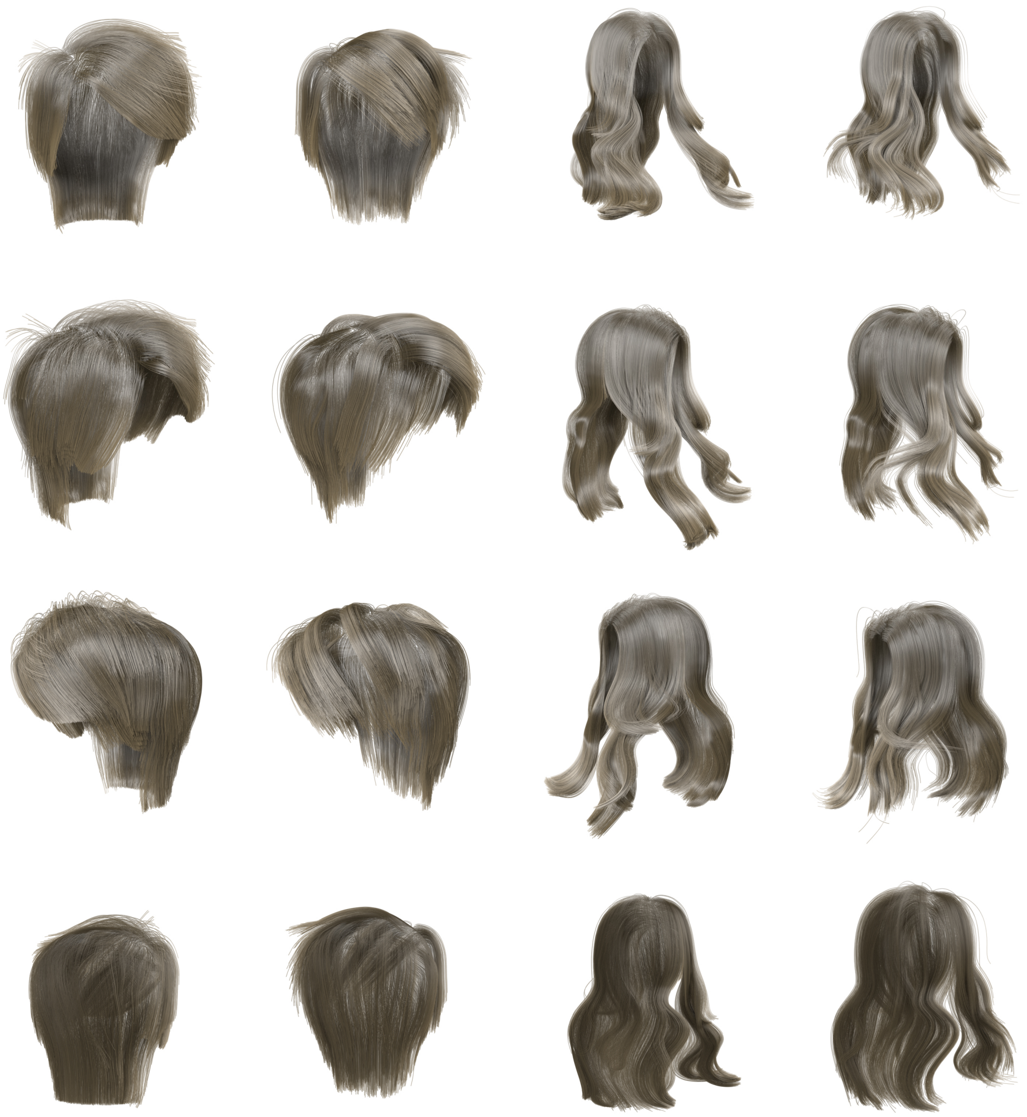


Figure 5.8: *Hairstyle synthesized by HSD approach compared with ground truth hairstyle. From left to right: the first column is a short straight hairstyle from USC-HairSalon, the second column is the corresponding synthesized short straight hairstyle by HSD, the third column is a long curly hairstyle from USC-HairSalon, the last column is the corresponding synthesized hairstyle by HSD.*

5.4.3 Single-viewed full head hairstyle reconstruction

Next, the novel HSD approach is tested on portraits with different hairstyles to validate the framework of Struct2Hair. For the purpose of proving its capability of recovering various hairstyles from single portrait image, five portrait pictures of different hairstyles have been selected as the input. The recovered hairstyle models are illustrated in Figure 5.9 with the same viewpoint compared to the original hair. Moreover, another picture has been rendered with a rotated angle of view to show the global hair consistency. The regenerated hairstyle models follow the natural way of

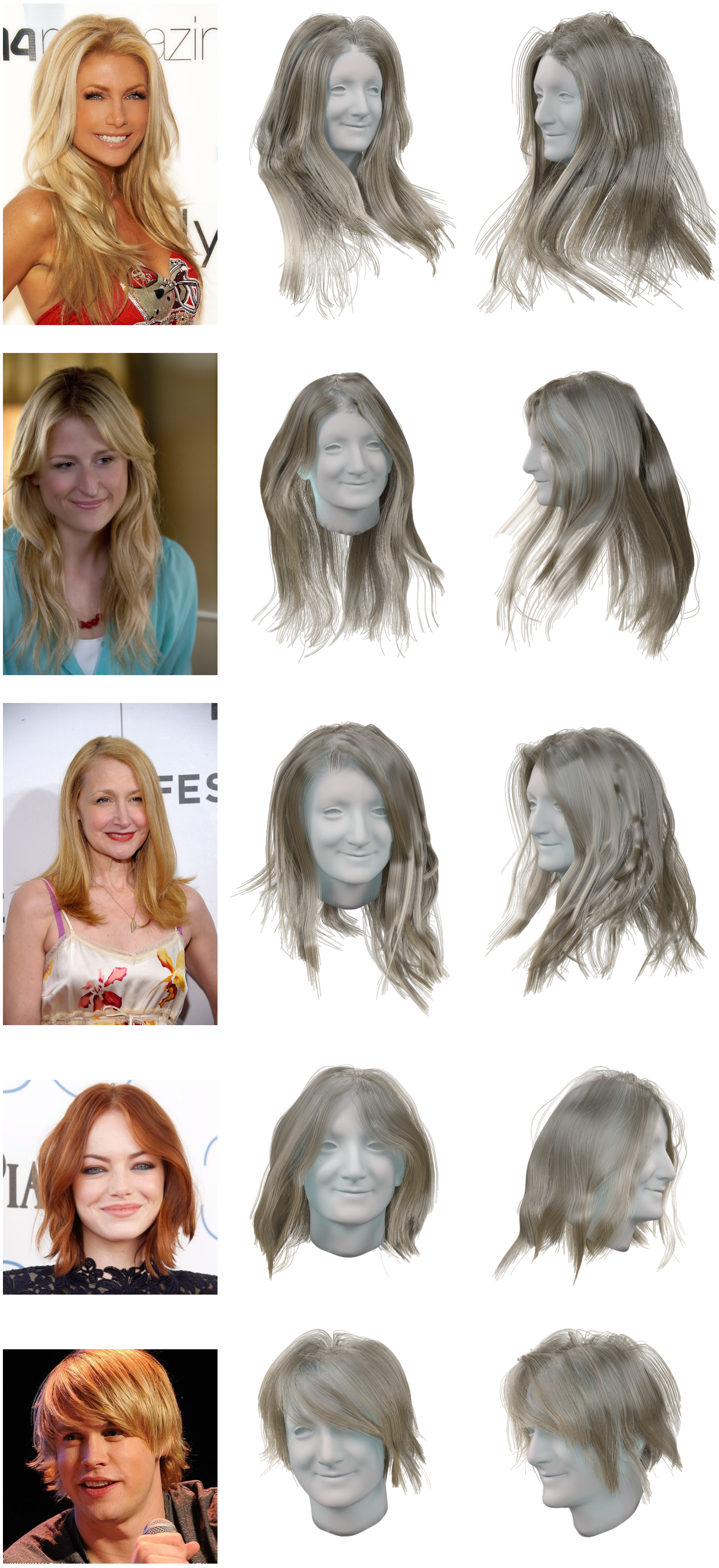
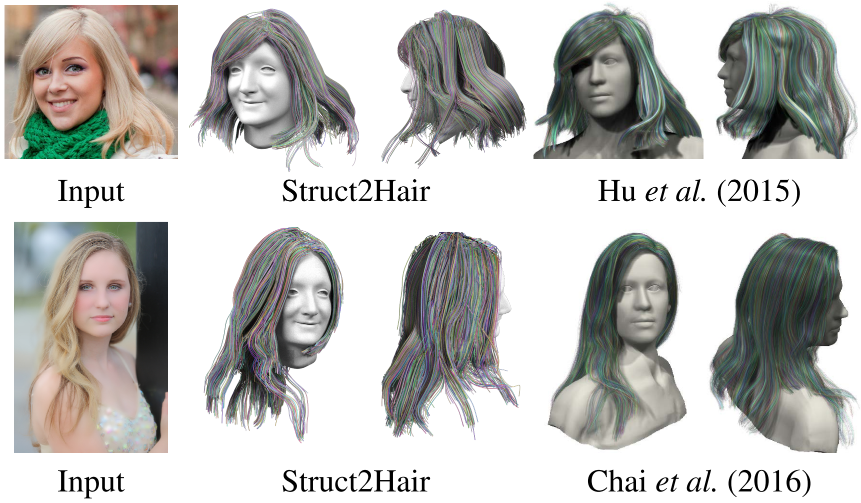


Figure 5.9: *Hairstyle models generated by HSD approach from single image. From left to right, 1 column is the original portrait, 2 column is the generated hairstyle model by HSD approach, 3 column is the corresponding side view. Portraits originate from Rothe* et al. *(2016).*

hair growth, which closely match the input portraits with the original angle of view.

5.4.4 Comparison with the state-of-the-art methods

The hair models generated from the Struct2Hair framework are compared with the state-of-the-art single image-based hair modelling algorithms (Hu *et al.*, 2015; Chai *et al.*, 2016) here. Figure 5.10 shows that our framework obtains similar quality of hairstyle models compared to the previous methods, which closely match the input portrait. However, different from their methods, our framework targets reconstruction of the reference hairstyle from its intrinsic structure, which benefits a presence of rich local details of hair shape.


Input Struct2Hair Hu *et al.* (2015)

Input Struct2Hair Chai *et al.* (2016)

Figure 5.10: *Compare with the state-of-the-art single-viewed hair modelling techniques. From left to right are the input portraits, hairstyle model generated by Struct2Hair, Hu* et al. *(2015) and Chai* et al. *(2016) respectively. Original image courtesy of Chris Zerbes and Bob HARRIS.*

Furthermore, the Struct2Hair framework is also compared with the cutting-edge multi-viewed hair modelling technique by Zhang *et al.* (2017). They use four-view images to reconstruct a target hairstyle. The result from our Struct2Hair is rendered under the four reference views to compare with their model in Figure 5.11. It presents the capability of our framework to generate reliable hairstyle model with only a single view reference image. Our result shows realistic details even for the side view.

5.4.5 Limitation

The HSD extraction is based on image pre-processing. In some cases, portrait images are lack of clear hair information due to high reflection or bad illumination will cause

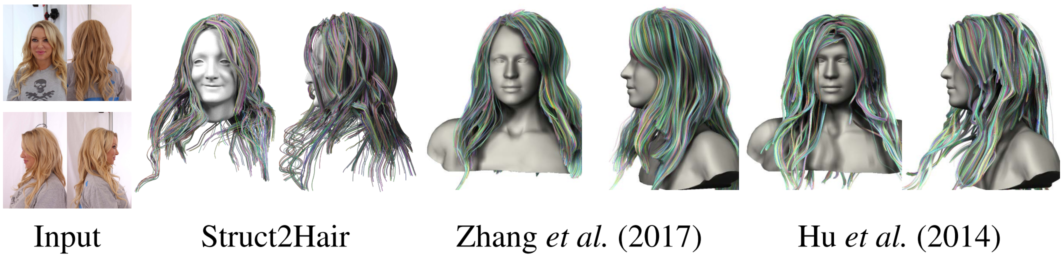


Figure 5.11: *Comparison with the state-of-the-art multi-viewed hair modelling techniques. From left to right are the input portraits, hairstyle model generated by Struct2Hair using from the front view input, Zhang* et al. *(2017) and Hu* et al. *(2014) respectively.*

unreliable HSD generation, which result in unrealistic hairstyle capturing. Our HSD method is incapable of dealing with the fuzzy hairstyle at the current stage. The intrinsic fuzzy hairstyle will effect the consistency of 2D hair strands extraction. In that case, a better performance 2D hair strands extraction approach should be considered. As the HSD descriptor is designed for capturing hairstyle from single image, the generated hair model can only keep the hairstyle matched the original view, and preserve a plausible visual effect when rotating the head due to the unknown ground-truth of the back side.

Moreover, similar to Chai *et al.* (2016), the head pose estimation method fails when extreme side-viewed face or tilted head appear in the input portrait. This will cause unreliable critical hair shape manipulation and unsatisfied hair model generation. The author believes this problem will be solved by the improvement of face alignment method. Also, like the previous data-driven based single-view hair modelling methods, our Struct2Hair is relied on the critical hair shape dataset constructed from the USC-HairSalon database. When there is no desired critical hair shape matched to the HSD, the quality of the reconstructed hair model is less satisfactory compared to Hu *et al.* (2015); Chai *et al.* (2016). This requires us to build a large critical hair shape database in the future work.

5.5 Summary

The Struct2Hair, a hair shape descriptor (HSD) approach for hair modeling is introduced in this paper. As far as our knowledge, HSD is the first direct hairstyle structure description. Compared to the state-of-the-art single-viewed hair modelling methods, HSD based hair modelling provides a bottom-up pipeline, which focuses the basic structure of a hairstyle. The generated hairstyle models show the ability of capturing local feature. The full-head hair grown around the scalp follows the natural behaviour which retain the whole shape of the hairstyle globally. Our compact HSD structure also enables a way of hairstyle management like saving and reloading the hairstyle, as well as enlarge the hairstyle database by appending new extracted HSD. Promising hairstyle editing and simulation applications can be developed by manipulating the HSD in future.

Chapter 6

Application

The purpose of single image based hair modelling is to help the animators and game designers save labour costs and improve the realism of virtual hair model. To better leverage the prior knowledge of hair shapes in the CHS database, the Struct2Hair framework enables enlarge the current hairstyle database by two ways: populating new hairstyle models and adding generated HSD from image inputs. As generating hairstyle models from image inputs have been described in chapter 5, this chapter will focus on how to obtain novel hairstyle exemplars by morphing hairstyles. Meanwhile, the Struct2Hair framework can be developed into a useful hair editing tool for further combing and styling hair model, which could be an extension of the current hair system in existing computer animation & modelling softwares. Apart from these, the image pre-processing methods have enhanced hair appearance in image, whose results have been used as a case study of Bas-relief hair modelling to investigate possible application of art creation.

6.1 Populating new hairstyles

The aforementioned USC-HairSalon database only contain 343 hairstyle models, it limits the hair reconstruction performance with lack of enough hairstyle exemplars. This section provides a method to populate new hairstyle models by blending hairstyles. As mentioned in the previous chapter, the full head hair model is generated under the guidance of HSD. To achieve the hairstyles blending, the author starts to remix the HSD to build a orientation field for the new hairstyle following the algorithm described in Section 5.3. The selected candidates are not only from the examples within the current CHS, the reconstructed HSDs from the images will join the blending process to increase the diversity and abundance of the database in future.

The first task of diffusing hair strands around the scalp is to decide where to place the hair follicles. Suppose there are $\boldsymbol{K}$ hairstyles used to build a new hair model, each 3D strand from the corresponding $\boldsymbol{K}$ HSD is denoted as $h_{i,k}$, $i = 1, 2, \ldots, 50$ and $k \in \boldsymbol{K}$. Then for each root point of the $h_{i,k}$, search its e nearest neighbouring follicles and collect them as an union of sets $\boldsymbol{F}$. These chosen points are the starting points to initiate hair strands generation iteratively.

The second task is to add the HSD coefficients to control the orientation vector at each step when integrating along the hair strand length. For a follicle point $\mathbf{f} \in \boldsymbol{F}$, its u nearest root points of the candidate HSD are $\mathbf{r}_1(h_{i,k})$, $\mathbf{r}_2(h_{i,k})$, $\ldots$, $\mathbf{r}_j(h_{i,k})$, $j = 1, 2, \ldots, u$. The defined orientation vector is:

$$\mathbf{f}'(s) = \sum_{j=1}^{u} \frac{\gamma_k \omega_j^2 \mathbf{r}'(h_{i,k})}{\sum_{j=1}^{u} \omega_j^2}, \sum_{k=1}^{K} \gamma_k = 1, \tag{6.1}$$

where γ_k is the coefficients for the k^{th} HSD, the sum of them equals 1. The distance weight ω is specified in section 5.3.2.

With the rules above, hairstyles blending is tested to investigate the synthesis of various hairstyles. To start the case study of blending two hairstyles, the coefficients $\gamma_1 : \gamma_2$ are set from $1 : 0$ to $0 : 1$. The blending results show that the hairstyle one morphing into the hairstyle two with changing the coefficients, see Figure 6.1 for details.

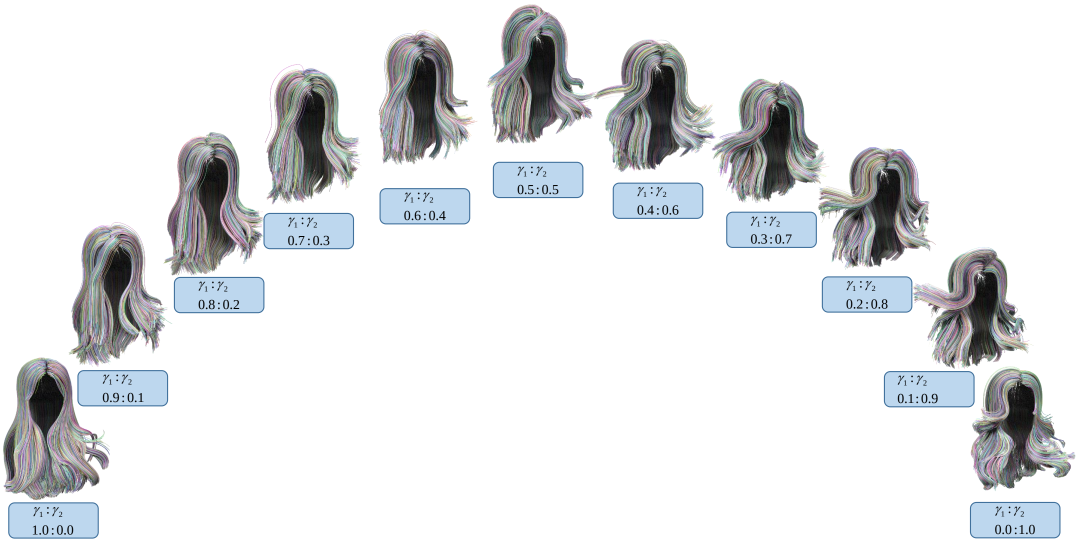

$\gamma_1 : \gamma_2$ 1.0:0.0
$\gamma_1 : \gamma_2$ 0.9:0.1
$\gamma_1 : \gamma_2$ 0.8:0.2
$\gamma_1 : \gamma_2$ 0.7:0.3
$\gamma_1 : \gamma_2$ 0.6:0.4
$\gamma_1 : \gamma_2$ 0.5:0.5
$\gamma_1 : \gamma_2$ 0.4:0.6
$\gamma_1 : \gamma_2$ 0.3:0.7
$\gamma_1 : \gamma_2$ 0.2:0.8
$\gamma_1 : \gamma_2$ 0.1:0.9
$\gamma_1 : \gamma_2$ 0.0:1.0

Figure 6.1: *Blend hairstyle one with hairstyle two by setting different coefficients γ_k. From left to right, the hairstyle is morphing from pure hairstyle one to pure hairstyle two with the $\gamma_1 : \gamma_2$ changing from $[1 : 0]$ to $[0 : 1]$.*

6.2 Hairstyle editing

The hairstyle is one of most important features to enhance your personality. When one's hairstyle is cooperating, it frames his/her face, gives a distinct and even younger look. One hairstyle editing application is built upon the Struct2Hair framework. It enables a possible virtual hair try-on application by cutting or extending hair, adding fringe hair and curling hair, with the HSD based full head hair generation technique specified in section 5.3.1. All the operations could be done on the HSD to imitate the virtual hair styling, which are illustrated in Figure 6.2.

For cutting manipulation, one can cut the length of the HSD to regenerate a corresponding short hairstyle model. Other processes such as adding fringe hair, curling hair and extending hair are accomplished by appending structure strokes to the current HSD. Those structure strokes are selected from the centres in CHS database. In Figure 6.2, hair extension, curling and adding fringe hair are the combinations of original HSD and long strokes, curly strokes and fringe strokes respectively. The results show the validation of this method.

6.3 Bas-relief Modelling

Figure 6.3 shows a piece of bas-relief artwork of ancient Egypt. It can be seen that the hair plays an important role in the bas-relief model as it resembles the feminine beauty and attraction. The pre-processing techniques discussed in chapter 4 make images ready as inputs for Bas-relief like hair modelling. The author follows the main technique of Bas-relief modelling developed in Meili *et al.* (2010); Wang *et al.* (2012). After obtaining the pre-processed hair images, unsharp masking is applied to enhance the details of hair, then the height of a 3D bas-relief hair model can be generated by solving a poisson equation:

$$\nabla^2 h(x, y) - g = 0, \tag{6.2}$$

$$g = \nabla \cdot G(\nabla I_d) \tag{6.3}$$

where h is the height, I_d is the input image, and a compress function G nonlinearly scales the gradient of the input image.

Reliefs of different hairstyles are shown in Figure 6.4. Two main pre-processing techniques are compared here. One is the Gabor filter, which mimics the function of human eyes and extracts features to represent hair orientations. The other is the steerable filter which uses Gaussian convolution to estimate the gradient information

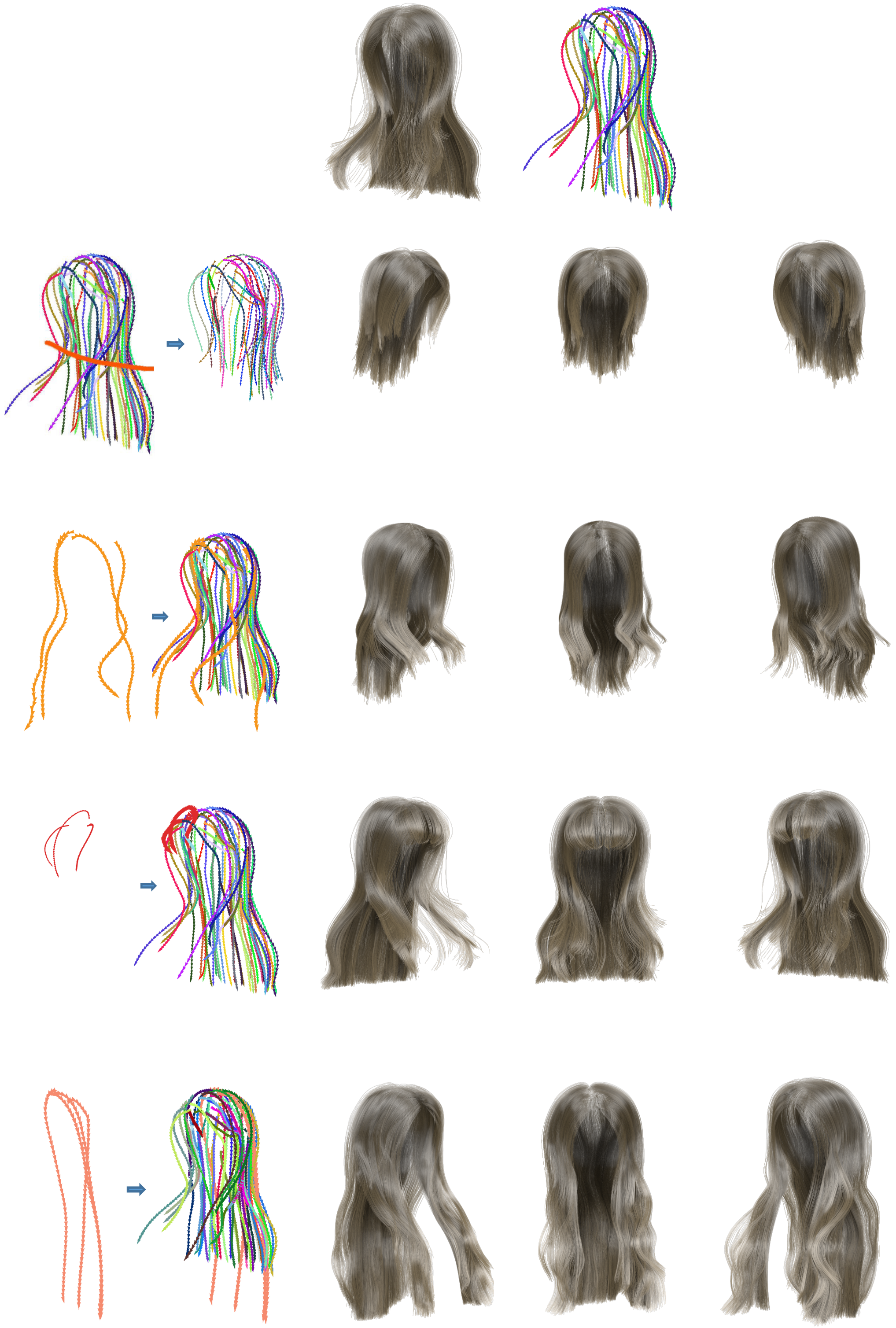


Figure 6.2: *Hair editing by HSD manipulation. First row is the original hairstyle and its corresponding HSD. From the second row to bottom row, left column is HSD manipulation, middle left column to right column are different views of the new hairstyles. First row, hair cut by cutting long HSD to short HSD. Second row, hair curl by adding curly hair shape to current HSD to change 3D orientation field. Third row, add fringe hair by appending fringe hair shape to current HSD. Last row, hair extension by adding long hair shape to current HSD.*

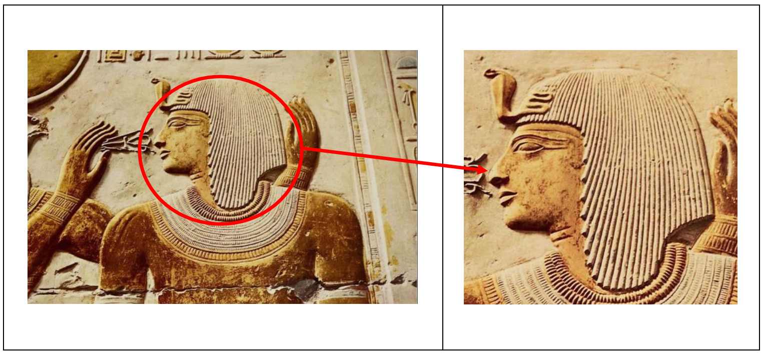


Figure 6.3: *A piece of bas-relief artwork of ancient Egypt.*

which helps to locate the hair growth direction.

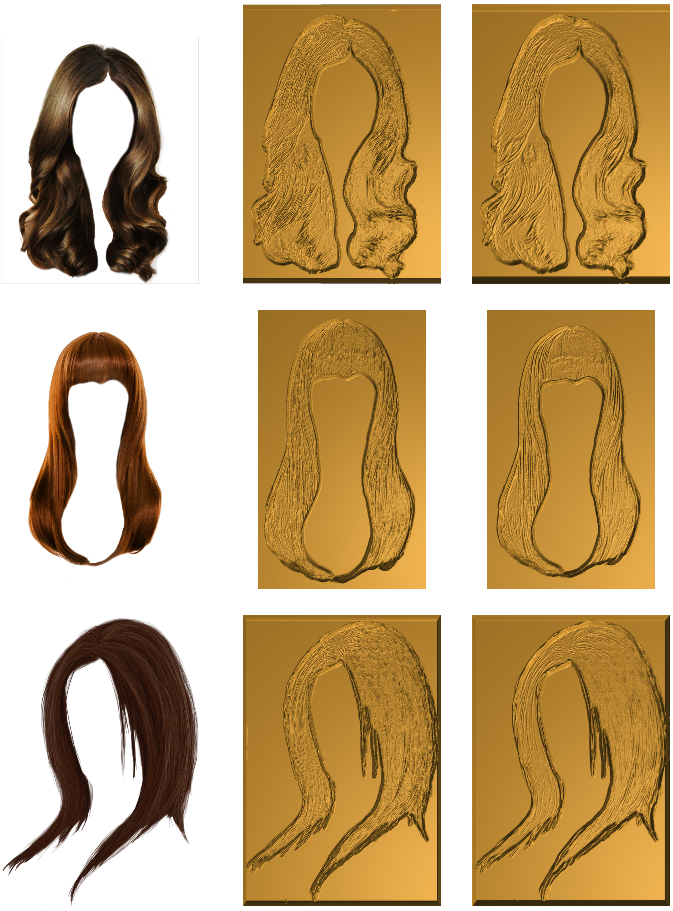


Figure 6.4: *Bas-relief like hair modeling. Left column: Relief by Gabor filtering results. Right column: Relief by Steerable filtering results*

In Figure 6.4, the left column contains reliefs based on Gabor filtering results. Clear hair traces were captured especially near the hair partition line. The Steerable filtering results in the right column smoothed out some small features, but preserve the main structure and hair orientation well. In the relief modelling, a level of simplification is often preferred, that is artists tending to use fewer lines or less shape variation in the art form to abstract the main features and characteristics of the original references. It is noticed that the steerable filtering offer better image pre-processing for relief modelling than Gabor filter, providing a good balance of de-noising, capturing hair structure and preserving shape features. This is somehow on the contrary to the case of 3D reconstruction of hair model from an image, where the Gabor filter is preferred (Chai *et al.*, 2012) to take the advantage of its being robust to noise and reliable detection of local features.

The author also generates the relief models for a curvy hairstyle in Figure 6.5 which is hard to model and deal with using traditional modelling techniques due to the complex hair shape and orientations. Four image filters are used and results generated from them are compared here. In the original image, the space between overlaid hair wisps possess low contrast. Thus, Sobel filtering based relief model failed to enhance overlaid hair structure. The Gaussian derivative filter and the steerable filter smoothed the image to suit the relief production. They removed some details of high frequency but preserved the main features. Gabor filter successfully kept original hairstyle to satisfy the relief modelling.

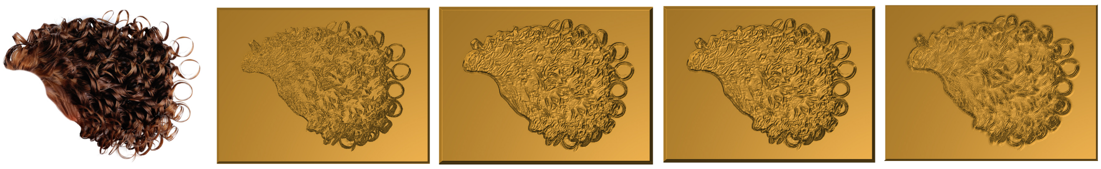


Figure 6.5: *Bas-relief curved hairstyle. From left to right: Sobel filtering based relief; Gaussian derivative filtering based relief; Steerable filtering based relief; Gabor filtering based relief*

6.4 Summary

This Chapter discussed the applications related to the Struct2Hair framework, including populating new hairstyle models by hairstyles blending, manipulating HSD to edit a hairstyle and generating hair art creation.

The hairstyle blending technique proposed in this Chapter provides a solution for further enlarging the hairstyle model database, which blends different hairstyles by creating a orientation vector field from the corresponding HSDs to construct the new hairstyle. The morphing results show the power of remixing different hairstyles.

Guaranteed by the HSD based hair modelling technique, various interesting manipulations could be operated on the Struct2Hair output. They formed a hair editing application that is capable of changing hairstyles without complicated computations. This is a promising attempt for virtual hairstyling, leading a future direction of designing a virtual hair Salon.

By applying pre-processing results calculated from chapter 4, this chapter also creates bas-relief stylisation of the hair images from the orientation maps. The results show that the steerable filter provides a good balance of de-noising and feature preserving, which works well for relief modelling as shown in this Chapter. Except for the relief models looking a bit rigid, the Gabor filter is proved as a reliable image pre-processing method for this purpose.

Chapter 7

Conclusions & Future Work

7.1 Conclusions

The realistic virtual hairstyle modelling is not an easy task. The diversity of the individual hair shapes and the large amount of them make the hair modelling even more difficult to accomplish. The traditional design of a virtual hair model costs hours to achieve a desirable sample. It impairs devising the personalized virtual hair for common users. Furthermore, manually modelled hair models are lack of naturalness compared to the hair modelled from the real-world data. To solve this problem, researchers have changed their focuses to capture virtual hairstyle model with natural looking.

The image based hair model acquisition techniques have attracted increasing attentions from the computer graphics community. It offers a photo-realistic hairstyle model without traditional demands of tedious and time consuming works. Most of the image based hair modelling methods require expensive set-ups or intensive user interactions. Among them, the single-viewed hair modelling is believed as the state-of-the-art technology, which prevents cost image processing and human intervention. It also provides a trade-off between the computation and the visual realism. However, the existing single-viewed hair modelling researches are top-down approaches, which show less interesting in the fundamental hair structure.

This thesis addresses this challenge by introducing an image based 3D hairstyle modelling framework *Strut2Hair*. The basic concept of this framework is to extract the hair structure from the single image input and reconstruct the 3D hairstyle model with the help of a data-driven approach. The proposed hair shape descriptor to encode the hair structure is the foundation of this framework. It reveals that a hairstyle can be represented by a skeleton form (HSD). The hair model generated from the HSD preserves both the local geometric shape and the global hairstyle consistency.

The Struct2Hair framework enables naturalness hair model generation for various hairstyles, ranging from the short straight hairstyles to the long wavy ones. The outputs of the framework verify the hypothesis that the hair shape descriptor is the skeleton of a hairstyle and capable of guiding full head hairstyle reconstruction. During this exploration, some findings have been conducted in addition to this framework, which lead to the main contributions and interesting applications of this thesis. The following paragraphs will discuss and summarize them in details.

This thesis has introduced the importance of hairstyling for modelling the realistic virtual character in computer graphics. For many years, the hair modelling researchers were focused on developing the high fidelity hair simulation with a simple hairstyle. The hair styling work has been limited to the professional artist. But the recent developments in capturing geometry from the real-world data have led to a great demand of the image based hair modelling. With reviewing the past-to-present literatures of hair geometry modelling works, this thesis built a single image based hair modelling framework to extend the current hairstyling technique.

This thesis has devised a strategy of extracting 2D hair sketch directly from image for the purpose of capturing structure. The sketch is the representation of hairstyle skeleton, this contributes to the pre-processing part of the Struct2Hair framework. 2D hair features extraction is a tough mission as the existence of unavoidable noises in an image. Hence related image processing filters have been introduced, compared and tested on the hair images to calculate an orientation map. The 2D hair strands are traced from the obtained orientation map. A clustering algorithm was then applied on the 2D hair strands to draw the hair sketch.

This thesis introduced a data-driven technique for solving ill-posed single-view 3D reconstruction problem. To leverage the power of data-driven measurement, this thesis constructed a critical hair shape database from an existing 3D hairstyle database, which was used to recover the depth of the 2D hair sketch. This thesis verified that 3D hair model can be generated under the guidance of a skeleton of the hairstyle, where the skeleton consists of the basic hair shape centres. The critical hair shape is a wisp containing hair strands with similar geometrical shape, length and located closely on the scalp as well. A tailored hierarchic clustering algorithm has obtained a satisfactory critical hair shape database. This CHS database will further benefit the sketch based hair modelling research in the future.

The proposed Struct2Hair framework bridges the gap between the input image and the 3D hairstyle model, with the designed hair shape descriptor. It is the core algorithm of the Struct2Hair framework. The hair shape descriptor is a 3D skeleton of a hairstyle.

A heuristic method is used to generate a HSD by matching candidate centres from the CHS database and mapping their depth information to the 2D hair sketch. The full head hair model then has been recreated under the orientation vector field generated by the HSD. The experiment results emphasize the validity of the Struct2Hair framework. The outputs were assessed by comparing to the ground-truth hairstyle models from the original database, the input portraits and the state-of-the-art image based hair modelling techniques. It is believed that this solution has a feasible observation of the hairstyle structure.

The findings of this thesis support several promising applications. One is populating new hairstyles with blending different hairstyles. It paves the way for enlarging the size of the existing hairstyle database. Meanwhile, a hairstyle editing application has been developed based on the HSD guided hair model generation. This straightforward application indicates some future directions of the Struct2Hair. The other application is a case study of Bas-relief modelling hairstyle, which is tested on the image pre-processing results. It explores the potential application of art creation on the image processing results.

Although this thesis found an innovative solution for modelling hairstyle based on a single image in a structure-aware manner, it is limited in several ways. First, the 2D hair strands tracing algorithm fails when input with extreme lighting condition, where only part of the 2D hair strands can be traced. Second, the Struct2Hair is currently unable to model extremely curly or fuzzy hairstyle due to only unreliable HSD extracted. Third, the modelling results depends on the diversity of the CHS database. Last, matching CHS centres for 2D hair sketch relies on the existing head pose estimation algorithm. It cannot model hairstyle for the case of extreme side-viewed face or tilted head. The next section will discuss some future directions to extend the work carried out by this thesis.

7.2 Future Work

7.2.1 Real-time framework design

The purpose of the Struct2Hair framework is to save manually hairstyling time for animators and assist the common user to do customised modelling. Lack of efficiency is the main downside of this framework. Current Struct2Hair is only capable of capturing 3D hairstyle off-line on a HP Z440 Workstation with 3.7GHz Interl(R) Xeon(R) CPU and 32GB memory. The hair diffusion and subdivision process are calculated on the CPU, where the number of the points increases exponentially.

It can be dramatically improved by a GPU-based coding, which has a impressive performance on computing intensive problems (Lee *et al.*, 2010). Additionally, the current code is developed with MATLAB, which is less efficiency with dealing iterative calculations. There are iterative problems when doing database retrieving and selecting hair roots for a patch ruled growing manner. Thus, one of the important future directions is to covert current code to GPU-based with a more recursion powerful programming language. The next version of the Struct2Hair should be a real-time interactive system and compatible with the industrial softwares like Maya, Houdini and Unity etc.

7.2.2 Layer optimisation

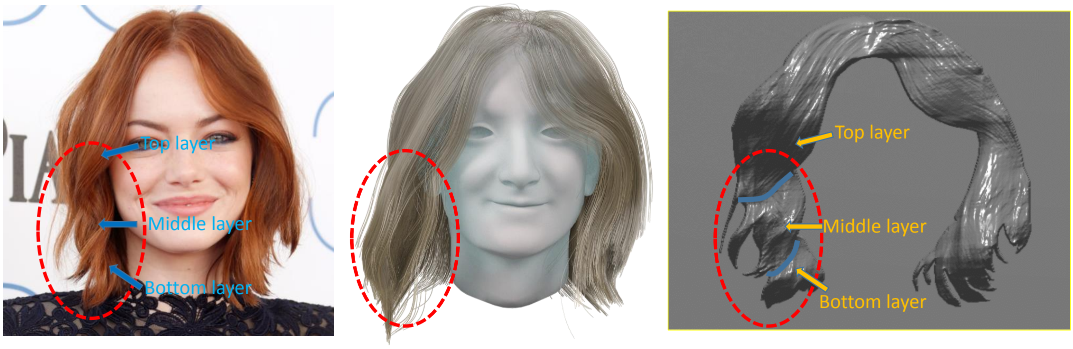


Figure 7.1: *Shape from shading based layer optimisation. From left to right, original portrait, current reconstructed hair model and shape from shading base shape. On the coarse hair surface, the different height areas indicate three different hair layers respectively, which could be the constraints added to the energy function when doing optimisation.*

Although the regenerated hair model are closely matched the original image as the initial expectation, there are many possibilities to improve the framework. One possible direction will be refining the hair layering by a layer optimisation. The current output hair model keeps the fine geometry details and the global shape consistency. However there is no middle layer to reveal the accurate hair alignment, which causes disappointed hair layering. A feasible method to solve this problem is to use a coarse hair surface shape to optimise the inaccurate layering. This base shape could be calculated by the classic shape from shading method, where different height areas will be the guide layering (see Figure 7.1 for detail). The ultimate layer optimisation should achieve more precisely hair layering.

7.2.3 Virtual hair salon development

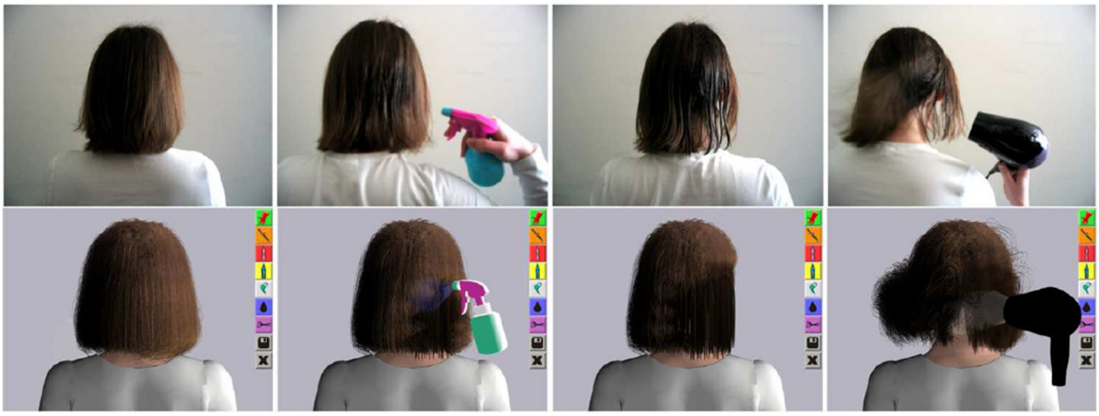


Figure 7.2: *The interactive virtual hair salon by Ward* et al. *(2006).*

As reported previously in section 6.2, a hairstyle editing application has been developed with some basic functions. It is a promising application for further development as there was a research pioneered the virtual hair salon application before (see Figure 7.2), but it was not designed as a customised tool (Ward *et al.*, 2006). The principle advantage of single image based hair modelling framework is that it could easily build personalized product. Especially as soon as the real-time framework has been done, the virtual hair salon application can be tailored onto the portable devices like mobile phone and tablet. For virtual games, the customised hairstyles will enhance the personality of the virtual character and improve the social engagement. For cosmetic industry, this type of application could improve the current online virtual hair try-on applications. Trying cosmetic production with the customised hair shape will boost highly virtual reality.

References

Aloimonos J., 1988. Shape from texture. *Biological cybernetics*, **58**(5), 345–360.

Anguelov D., Srinivasan P., Koller D., Thrun S., Rodgers J. and Davis J., 2005. Scape: shape completion and animation of people. In *ACM Transactions on Graphics (TOG)*, volume 24. ACM, 408–416.

Arthur D. and Vassilvitskii S., 2007. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 1027–1035.

Baak A., Müller M., Bharaj G., Seidel H.-P. and Theobalt C. 2013. 71–98. A data-driven approach for real-time full body pose reconstruction from a depth camera. In *Consumer Depth Cameras for Computer Vision*, Springer.

Bando Y., Chen B.-Y. and Nishita T., 2003. Animating hair with loosely connected particles. In *Computer Graphics Forum*, volume 22. Wiley Online Library, 411–418.

Beeler T., Bickel B., Noris G., Beardsley P., Marschner S., Sumner R. W. and Gross M., 2012. Coupled 3d reconstruction of sparse facial hair and skin. *ACM Transactions on Graphics (TOG)*, **31**(4), 117.

Bertails F., 2009. Linear time super-helices. In *Computer Graphics Forum*, volume 28. Wiley Online Library, 417–426.

Bertails F., Audoly B., Cani M.-P., Querleux B., Leroy F. and Lévêque J.-L., 2006. Super-helices for predicting the dynamics of natural hair. In *ACM Transactions on Graphics (TOG)*, volume 25. ACM, 1180–1187.

Bertails F., Kim T.-Y., Cani M.-P. and Neumann U., 2003. Adaptive wisp tree: a multiresolution control structure for simulating dynamic clustering in hair motion. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Eurographics Association, 207–213.

Blanz V. and Vetter T., 1999. A morphable model for the synthesis of 3d faces. In

Proceedings of the 26th annual conference on Computer graphics and interactive techniques. ACM Press/Addison-Wesley Publishing Co., 187–194.

Canny J., 1986. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (6), 679–698.

Chai M., Luo L., Sunkavalli K., Carr N., Hadap S. and Zhou K., 2015. High-quality hair modeling from a single portrait photo. *ACM Transactions on Graphics (TOG)*, **34**(6), 204.

Chai M., Shao T., Wu H., Weng Y. and Zhou K., 2016. Autohair: Fully automatic hair modeling from a single image. *ACM Transactions on Graphics (TOG)*, **35**(4), 116.

Chai M., Wang L., Weng Y., Jin X. and Zhou K., 2013. Dynamic hair manipulation in images and videos. *ACM Trans. Graph*, **32**, 4.

Chai M., Wang L., Weng Y., Yu Y., Guo B. and Zhou K., 2012. Single-view hair modeling for portrait manipulation. *ACM Transactions on Graphics (TOG)*, **31**(4), 116.

Chaudhuri S. and Koltun V., 2010. Data-driven suggestions for creativity support in 3d modeling. *ACM Transactions on Graphics (TOG)*, **29**(6), 183.

Cheang S., 2008. *Hair: Styling, culture and fashion*. Berg.

Chen H. and Zhu S.-C., 2006. A generative sketch model for human hair analysis and synthesis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **28**(7), 1025–1040.

Chen K., Lai Y., Wu Y.-X., Martin R. R. and Hu S.-M., 2014. Automatic semantic modeling of indoor scenes from low-quality rgb-d data using contextual information. *ACM Transactions on Graphics*, **33**(6).

Chen L.-H., Saeyor S., Dohi H. and Ishizuka M., 1999. A system of 3d hair style synthesis based on the wisp model. *The Visual Computer*, **15**(4), 159–170.

Chen Q., Li D. and Tang C.-K., 2012. Knn matting. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 869–876.

Chen X.-D., Ma W., Xu G. and Paul J.-C., 2010. Computing the hausdorff distance between two b-spline curves. *Computer-Aided Design*, **42**(12), 1197 – 1206.

Cheng A.-D., 2012. Multiquadric and its shape parametera numerical investigation of error estimate, condition number, and round-off error by arbitrary precision computation. *Engineering Analysis with Boundary Elements*, **36**(2), 220–239.

Choe B., Choi M. G. and Ko H.-S., 2005. Simulating complex hair with robust collision handling. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*. ACM, 153–160.

Choe B. and Ko H.-S., 2005. A statistical wisp model and pseudophysical approaches for interactive hairstyle generation. *IEEE Transactions on Visualization and Computer Graphics*, **11**(2), 160–170.

Ding C. and Liu L., 2016. A survey of sketch based modeling systems. *Frontiers of Computer Science*, **10**(6), 985–999.

Echevarria J. I., Bradley D., Gutierrez D. and Beeler T., July 2014. Capturing and stylizing hair for 3d fabrication. *ACM Trans. Graph.*, **33**(4), 125:1–125:11.

Favaro P. and Soatto S., 2005. A geometric approach to shape from defocus. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **27**(3), 406–417.

Freeman W. T. and Adelson E. H., 1991. The design and use of steerable filters. *IEEE Transactions on Pattern analysis and machine intelligence*, **13**(9), 891–906.

Fu H., Wei Y., Tai C.-L. and Quan L., 2007. Sketching hairstyles. In *Proceedings of the 4th Eurographics Workshop on Sketch-based Interfaces and Modeling*, SBIM '07, New York, NY, USA. ACM, 31–36.

Gao L., Lai Y.-K., Liang D., Chen S.-Y. and Xia S., 2016. Efficient and flexible deformation representation for data-driven surface modeling. *ACM Transactions on Graphics (TOG)*, **35**(5), 158.

Grabli S., Sillion F. X., Marschner S. R., Lengyel J. E. and others , 2002. Image-based hair capture by inverse lighting. In *Proceedings of Graphics Interface (GI)*, 51–58.

Hadap S. and Magnenat-Thalmann N., 2000. *Interactive hair styler based on fluid flow*. Springer.

Hadap S. and Magnenat-Thalmann N., 2001. Modeling dynamic hair as a continuum. In *Computer Graphics Forum*, volume 20. Wiley Online Library, 329–338.

Hartley R. I. and Zisserman A., 2004. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition.

Herrera T. L., Zinke A. and Weber A., 2012. Lighting hair from the inside: A thermal approach to hair reconstruction. *ACM Transactions on Graphics (TOG)*, **31**(6), 146.

Hu L., Ma C., Luo L. and Li H., 2014. Robust hair capture using simulated examples. *ACM Transactions on Graphics (TOG)*, **33**(4), 126.

Hu L., Ma C., Luo L. and Li H., 2015. Single-view hair modeling using a hairstyle database. *ACM Transactions on Graphics (TOG)*, **34**(4), 125.

Huang Q., Wang H. and Koltun V., 2015. Single-view reconstruction via joint analysis of image and shape collections. *ACM Transactions on Graphics (TOG)*, **34**(4), 87.

Huber P., Hu G., Tena R., Mortazavian P., Koppen W. P., Christmas W., Rätsch M. and Kittler J., 2016. A multiresolution 3d morphable face model and fitting framework. In *International Conference on Computer Vision Theory and Applications (VISAPP)*, 1–8.

Ivanov D., Lempitskii V., Shokurov A., Khropov A. and Kuzmin Y., 2003. Creating personalized head models from image series. In *Proc. of the 13th Int. Conf. on Comput. Graphics GraphiCon*.

Jakob W., Moon J. T. and Marschner S., 2009. Capturing hair assemblies fiber by fiber. In *ACM Transactions on Graphics (TOG)*, volume 28. ACM, 164.

Johnson M., Moore L. and Ylvisaker D., 1990. Minimax and maximin distance designs. *Journal of Statistical Planning and Inference*, **26**(2), 131 – 148.

Kamarainen J.-K., Kyrki V. and Kalviainen H., 2006. Invariance properties of gabor filter-based features-overview and applications. *Image Processing, IEEE Transactions on*, **15**(5), 1088–1099.

Kanopoulos N., Vasanthavada N. and Baker R. L., 1988. Design of an image edge detection filter using the sobel operator. *Solid-State Circuits, IEEE Journal of*, **23** (2), 358–367.

Kaufman D. M., Tamstorf R., Smith B., Aubry J.-M. and Grinspun E., 2014. Adaptive nonlinearity for collisions in complex rod assemblies. *ACM Transactions on Graphics (TOG)*, **33**(4), 123.

Kennedy L. M. and Basu M., 1999. A gaussian derivative operator for authentic edge detection and accurate edge localization. *International journal of pattern recognition and artificial intelligence*, **13**(03), 367–380.

Kim T.-Y. and Neumann U., 2002. Interactive multiresolution hair modeling and editing. In *ACM Transactions on Graphics (TOG)*, volume 21. ACM, 620–629.

Kong W. and Nakajima M., 1998. Generation of 3d hair model from multiple pictures. *The Journal of the Institute of Image Information and Television Engineers*, **52**(9), 1351–1356.

Kong W. and Nakajima M., 1999. Visible volume buffer for efficient hair expression and shadow generation. In *Computer Animation, 1999. Proceedings*. IEEE, 58–65.

Lee D.-W. and Ko H.-S., 2001. Natural hairstyle modeling and animation. *Graphical Models*, **63**(2), 67–85.

Lee V. W., Kim C., Chhugani J., Deisher M., Kim D., Nguyen A. D., Satish N., Smelyanskiy M., Chennupaty S., Hammarlund P. and others , 2010. Debunking the 100x gpu vs. cpu myth: an evaluation of throughput computing on cpu and gpu. *ACM SIGARCH computer architecture news*, **38**(3), 451–460.

Lee Y., Terzopoulos D. and Waters K., 1995. Realistic modeling for facial animation. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. ACM, 55–62.

Luo L., Li H., Paris S., Weise T., Pauly M. and Rusinkiewicz S., 2012. Multi-view hair capture using orientation fields. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 1490–1497.

Luo L., Li H. and Rusinkiewicz S., 2013a. Structure-aware hair capture. *ACM Transactions on Graphics (TOG)*, **32**(4), 76.

Luo L., Zhang C., Zhang Z. and Rusinkiewicz S., 2013b. Wide-baseline hair capture using strand-based refinement. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 265–272.

MacDorman K. F. and Ishiguro H., 2006. The uncanny advantage of using androids in cognitive and social science research. *Interaction Studies*, **7**(3), 297–337.

Magnenat-Thalmann N., Hadap S. and Kalra P., 2000. State of the art in hair simulation. In *International Workshop on Human Modeling and Animation, Korea Computer Graphics Society*, 3–9.

Magnenat-Thalmann N., Montagnol M., Bonanni U. and Gupta R., 2007. Visuo-haptic interface for hair. In *Cyberworlds, 2007. CW'07. International Conference on*. IEEE, 3–12.

McAdams A., Selle A., Ward K., Sifakis E. and Teran J., 2009. Detail preserving continuum simulation of straight hair. In *ACM Transactions on Graphics (TOG)*, volume 28. ACM, 62.

Mehrotra R., Namuduri K. R. and Ranganathan N., 1992. Gabor filter-based edge detection. *Pattern Recognition*, **25**(12), 1479–1494.

Meili W., Jian C., Junjun P. and Zhang J. J., 2010. Image-based bas-relief generation with gradient operation. In *Proceedings of the 11th IASTED International Conference*, volume 679, 33.

Morris M. D. and Mitchell T. J., 1995. Exploratory designs for computational experiments. *Journal of Statistical Planning and Inference*, **43**(3), 381 – 402.

Müller M., Kim T.-Y. and Chentanez N., 2012. Fast simulation of inextensible hair and fur. In *Workshop on Virtual Reality Interaction and Physical Simulation*. The Eurographics Association, 39–44.

Paris S., Briceño H. M. and Sillion F. X., 2004. Capture of hair geometry from multiple images. In *ACM Transactions on Graphics (TOG)*, volume 23. ACM, 712–719.

Paris S., Chang W., Kozhushnyan O. I., Jarosz W., Matusik W., Zwicker M. and Durand F., 2008. Hair photobooth: geometric and photometric acquisition of real hairstyles. *ACM Trans. Graph*, **27**(3), 30.

Petkov N., 1995. Biologically motivated computationally intensive approaches to image pattern recognition. *Future Generation Computer Systems*, **11**(4-5), 451–465.

Prados E. and Faugeras O., 2006. Shape from shading. *Handbook of mathematical models in computer vision*, 375–388.

Remil O., Xie Q., Xie X., Xu K. and Wang J., 2017. Surface reconstruction with data-driven exemplar priors. *Computer-Aided Design*, **88**, 31–41.

Robbins C. R. and Robbins C. R., 2002. *Chemical and physical behavior of human hair*, volume 4. Springer.

Rosenblum R. E., Carlson W. E. and Tripp E., 1991. Simulating the structure and dynamics of human hair: modelling, rendering and animation. *The Journal of Visualization and Computer Animation*, **2**(4), 141–148.

Rothe R., Timofte R. and Gool L. V., July 2016. Deep expectation of real and apparent age from a single image without facial landmarks. *International Journal of Computer Vision (IJCV)*.

Rungjiratananon W., Kanamori Y. and Nishita T., 2012. Wetting effects in hair simulation. In *Computer Graphics Forum*, volume 31. Wiley Online Library, 1993–2002.

Selle A., Lentine M. and Fedkiw R., 2008. A mass spring model for hair simulation. In *ACM Transactions on Graphics (TOG)*, volume 27. ACM, 64.

Shao T., Xu W., Zhou K., Wang J., Li D. and Guo B., 2012. An interactive approach to semantic modeling of indoor scenes with an rgbd camera. *ACM Transactions on Graphics (TOG)*, **31**(6), 136.

Shen C.-H., Fu H., Chen K. and Hu S.-M., 2012. Structure recovery by part assembly. *ACM Transactions on Graphics (TOG)*, **31**(6), 180.

Sloan R. J. S., 2015. *Virtual Character Design for Games and Interactive Media*. CRC Press.

Taheri M. and Ahmadian M., 2016. Machine learning from computer simulations with applications in rail vehicle dynamics. *Vehicle System Dynamics*, **54**(5), 653–666.

Umetani N., Schmidt R. and Stam J., 2014. Position-based elastic rods. In *ACM SIGGRAPH 2014 Talks*. ACM, 47.

Vanakittistien N., Sudsang A. and Chentanez N., 2016. 3d hair model from small set of images. In *Proceedings of the 9th International Conference on Motion in Games*, MIG '16, New York, NY, USA. ACM, 85–90.

Vernall D. G., 1961. A study of the size and shape of cross sections of hair from four races of men. *American Journal of Physical Anthropology*, **19**(4), 345–350.

Wang J. and Cohen M. F., 2008. *Image and video matting: a survey*. Now Publishers Inc.

Wang L., Yu Y., Zhou K. and Guo B., 2009. Example-based hair geometry synthesis. In *ACM Transactions on Graphics (TOG)*, volume 28. ACM, 56.

Wang M., Chang J., Kerber J. and Zhang J. J., 2012. A framework for digital sunken relief generation based on 3d geometric models. *The Visual Computer*, **28**(11), 1127–1137.

Ward K., Bertails F., Kim T.-Y., Marschner S., Cani M.-P. and Lin M., March 2007. A survey on hair modeling: Styling, simulation, and rendering. *Visualization and Computer Graphics, IEEE Transactions on*, **13**(2), 213–234.

Ward K., Galoppo N. and Lin M. C., 2006. A simulation-based vr system for interactive hairstyling. In *Virtual Reality Conference, 2006*. IEEE, 257–260.

Ward K. and Lin M. C., 2003. Adaptive grouping and subdivision for simulating hair dynamics. In *Computer Graphics and Applications, 2003. Proceedings. 11th Pacific Conference on*. IEEE, 234–243.

Watanabe Y. and Suenaga Y. 1989. 691–700. Drawing human hair using wisp model. In *New Advances in Computer Graphics*, Springer.

Wei Y., Ofek E., Quan L. and Shum H.-Y., 2005. Modeling hair from multiple views. In *ACM Transactions on Graphics (TOG)*, volume 24. ACM, 816–820.

Wither J., Bertails F. and Cani M.-P., 2007. Realistic hair from a sketch. In *Shape Modeling and Applications, 2007. SMI'07. IEEE International Conference on*. IEEE, 33–42.

Witkin A. P., 1980. Shape from contour.

Xu K., Zheng H., Zhang H., Cohen-Or D., Liu L. and Xiong Y., 2011. Photo-inspired model-driven 3d object modeling. In *ACM Transactions on Graphics (TOG)*, volume 30. ACM, 80.

Xu Z. and Yang X. D., 2001. V-hairstudio: an interactive tool for hair design. *IEEE Computer Graphics and Applications*, **21**(3), 36–43.

Yang X. D., Xu Z., Yang J. and Wang T., 2000. The cluster hair model. *Graphical Models*, **62**(2), 85–103.

Yu Y., 2001. Modeling realistic virtual hairstyles. In *Computer Graphics and Applications, 2001. Proceedings. Ninth Pacific Conference on*. IEEE, 295–304.

Zhang M., Chai M., Wu H., Yang H. and Zhou K., July 2017. A data-driven approach to four-view image-based hair modeling. *ACM Trans. Graph.*, **36**(4), 156:1–156:11.

Zinke A., Yuksel C., Weber A. and Keyser J., 2008. Dual scattering approximation for fast multiple scattering in hair. *ACM Transactions on Graphics (TOG)*, **27**(3), 32.