

Realistic Texture Synthesis for Point-based Fruitage Phenotype

Journal:	<i>IEEE/ACM Transactions on Computational Biology and Bioinformatics</i>
Manuscript ID	TCBB-2016-03-0101
Manuscript Type:	Regular Paper
Keywords:	Fruitage morphology, Texton division, Texture synthesis, Texture sampling

SCHOLARONE™
Manuscripts

Peer Review Only

Realistic Texture Synthesis for Point-based Fruitage Phenotype

Hui Jun YANG, Ehtzaz Chaudhry, Jian Chang, Jian Jun Zhang, Nan Geng

Abstract— Although current 3D scanner technology can acquire textural images from a point model, visible seams in the image, inconvenient data acquisition and occupancy of a large space during use are points of concern for outdoor fruit models. In this paper, an SPSDW (simplification and perception based subdivision followed by down-sampling weighted average) method is proposed to balance memory usage and texture synthesis quality using a crop fruit, such as apples, as a research subject for a point-based fruit model. First, the quadtree method is improved to make splitting more efficient, and a reasonable texton descriptor is defined to promote query efficiency. Then, the color perception feature is extracted from the image for all of the pixels. Next, an advanced sub-division scheme and down-sampling strategy are designed to optimize memory space. Finally, a weighted oversampling method is proposed for high-quality texture mixing. This experiment demonstrates that the SPSDW method preserves the mixed texture more realistically and smoothly and preserves color memory up to 94%, 92.3% and 85.7% better than direct 2D mapping, multi-resolution quantization and the color vision model method, respectively.

Index Terms— Fruitage morphology, Texton division, Texture synthesis, Texture sampling

1 INTRODUCTION

TO date, the current fruit modeling, realistic rendering and fruit quality analysis mainly depend on photorealistic 2D images [1,2], forward modeling methods [3-5] and triangular mesh-based methods [6]. However, these methods only consider VFX, and the neglected realistic appearance and texture of various fruits consume large amounts of CPU time, storage, or rendering distortion when applied to a high-precision fruit to cope with the anamorphose caused by color, resolution, accuracy, pose, coverage, lighting, scaling and background [7,8]. Therefore, these methods cannot satisfy the higher continuity requirements to accurately reproduce the texture morphology of complex crop fruits.

1.1 Motivation

With the development of agricultural informatization and electronic farming, the real appearance traits of plant organs and fruits surfaces, such as smoothness and color, play an ever more important role in the online sales of crops, product display, and the maintenance and representation of growth stages [9,10].

In contrast, with the development and improvement of 3D scanning precision [11,12], the point-based modeling method has made it possible to acquire the cloud points of plants and fruits with rich detail and high density [13-16]. Furthermore, because of non-topologic connection among points, it can present the morphology of plant organs with rich details more flexibly and veritably than other methods.

- Hui Jun YANG is with the College of Information Engineering, Northwest A&F University, Yang'ling, Shaan'xi, China. And National Centre for Computer Animation, Bournemouth University, Poole, Dorset, BH12 5BB, UK. E-mail: hyang@bournemouth.ac.uk.
- Ehtzaz Chaudhry, Jian Chang and Jian Jun Zhang are with National Centre for Computer Animation, Bournemouth University, Poole, Dorset, BH12 5BB, UK. E-mail: JChang@bournemouth.ac.uk, echaudhry@bournemouth.ac.uk, jzhang@bournemouth.ac.uk.

However, real fruits, usually growing in wild trees with a scattering background, present a challenge for the acquisition of both the points and image at the same time, causing the fruit textures to be separated from scan. A high-precision scanner is employed to capture the rich details of the fruits, resulting in high complexity for numerous point clouds, which can be time consuming on a general computer. To process these data effectively, the point clouds were typically simplified into small units in advance [17-19], which can result in seams, gaps or overlap between simplified primitives. Therefore, it is important to research primitive-based fruit texture synthesis.

1.2 Contribution and Organization

The aim of this paper was to synthesize texture in a simplified fruit model based on the scanned point cloud intensity information and the image information from a synchronized external digital camera. The proposed SPSDW (simplification and perception based subdivision followed by down-sampling weighted average) method can compress fruit texture storage space while maintaining texture authenticity, as shown in Fig. 1. The main contributions of this paper are as follows:

- (1) Texture simplification: based on previous geometric division [19], we simplify texture by improving the quadtree algorithm for adaptive segmentation and organizing texture with a new feature descriptor.
- (2) Texture space compression: we develop visual-based subdivision and down sampling to compress texture storage based on the human eye's perception principle.
- (3) High-quality texture synthesis: we improve texture synthesis quality by designing a weighted average and overlapping sampling scheme based on visual perception.

In what follows, we first discuss the related work in Section 2. Then, we acquire and analyze fruit texture materials in Section 3. Next, we provide the methods for texture organization and synthesis in Section 4. Finally, Sections 5 and 6 show the experimental analysis, comparisons and conclusions.

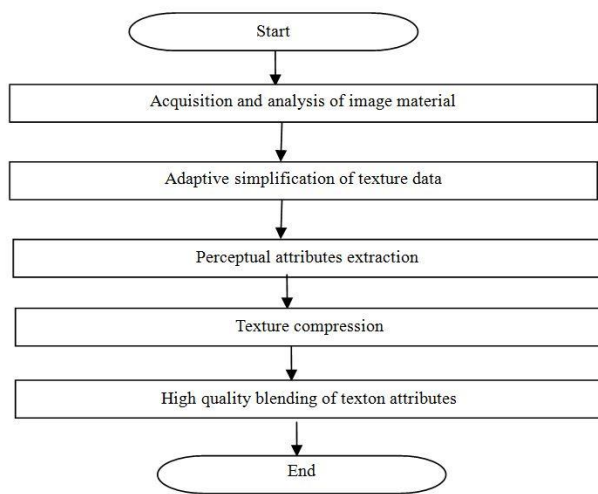


Fig. 1. Flowchart of texture synthesis.

2 RELATED WORK

In 3D point cloud modeling, triangular mesh-based texture synthesis methods have the additional requirement of converting point clouds to triangular shapes, resulting in high mesh density, unrealistic small curvature areas [20], and reproductions of whole texture synthesis in which situation any one of the points is added, deleted, or edited [21,22]. Block-based fast rendering methods require texture information pretreatment, an artificial process that sacrifices quality [23]. Consequentially, these methods are not applicable to those objects with complex structure, such as plant organs, leading to the increasing development of direct point-based synthesis technology.

The direct point-based model has several advantages compared to triangular meshes. It is considerably easier to edit sample points on the model and reconstruct the final texture directly with point information; however, there are no adjacency relations between points [23]. Therefore, point-based texture synthesis is typically only an auxiliary application of model editing [24], resulting in a poor experience [21, 25] because of the tactile feedback [26] and a low-quality synthesis effect [27]. R. F. Wang et al. presented texture restoring and mapping methods for a specific chosen area; however, these methods cannot generate a feasible texture structure or a smooth effect [28]. Zhao Xu reconstructed the Datong Yungang Grottoes Heritage Site by directly mapping the corrected digital images to the 2D coordinates of the point clouds, with some apparent distortion [29]. However, these techniques can only be used for the surface shading of point models, focus on the user's editing, and cannot reflect the real texture characteristics of a fruit surface.

Other point-based texture mapping systems are based on extracted features or require the manual selection of features between 2D images and 3D points [30, 31]. Lingy Liu presented a texture mapping system that focuses on the accuracy problem of geometric transformation [31]. Starting from a point, Paul Rosenthal rendered the lit point clouds into texture with color, depth and normal information point-by-point [32], and UAV-borne 3-D mapping systems allowed objects to be rendered with rich shapes and detailed textures [33], but they all resulted in a dramatic performance decline as the point number increased. Using Wasserstein barycenters for point clouds

in texture mixing has the additional requirement of averaging features and slicing approximation on each dimension distribution [34]. Color adjustment techniques eliminated visible seams in the texture map of a 3D object, although this does not fully avoid out-of-focus image areas in the viewing selection [8].

In this paper, the image and its corresponding geometric information are stored. After simplification and segmentation of the point model, the geometric properties and organization of the point cells changed dramatically, preventing the original texture information from being mapped to the corresponding geometric descriptor. Therefore, in light of the limitations of current methods in fruit texture synthesis, including coarse distortion, difficulty in texture data organization, and expensive memory space, this paper focuses on the efficient organization and actual texture blending of fruit texture.

3 ACQUISITION AND ANALYSIS OF FRUIT TEXTURE MATERIAL

To make our research more practical and applicable, we choose "micui" breeds of apple from the Northwest A&F University experimental field as the research object, which is also representative of certain texture properties and convenient for data acquisition. Its textural features are of great significance to agricultural researchers studying and comparing the history, development and characteristics of different apple varieties to generate improved fruits.

According to fruit object requirements, image accuracy, shooting speed, cost, device weight, file format, portability, and performance in time and space, we chose a digital camera with a PCP-400 laser 3D scanner as the image acquisition device [35,36]. The real size of a piece of acquired image data is approximately 9 cm*12 cm. Therefore, we designed the following acquisition scheme according to fruit size, surface curvature, scanner range, and ensuring a 10% overlap between two adjacent fruit slices. The side surfaces are captured nine times, and the bottom and top were captured once, yielding 11 groups of data. These point clouds are presented by the array $app(11)$ in Eq. (3-1).

$$app(11) = \{b, t, s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9\} \quad (3-1)$$

The corresponding image is presented by the array $apptex(11)$ in Eq. (3-2).

$$apptex(11) = \{b, t, s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9\} \quad (3-2)$$

The texture images were saved in *.BMP format, and the texture sizes of apple images are as follows: bottom, 560*459; top, 512*441; nine sides, 440*383. The corresponding texture images are shown in Fig. 2 and S1 Fig. 2. Geometric attributes were saved, and their corresponding texture information was organized, resulting in texture dislocation and a large amount of data.

4 TEXTURE SYNTHESIS ON FRUIT MORPHOLOGY

For the convenient integration between point and texture, we described the subject with a two low-dimensional uniform space, namely, geometric and color space. Then, the initial corresponding relationship between geometric properties and the color value of a point is defined in Eq. (4-1)

$$\{C, T\} = \{p_i, t_i\} = \{(x_i, y_i, z_i), (R_i, G_i, B_i)\} \quad (i = 1, 2 \dots M) \quad (4-1)$$

in which M is the maximum number of image colors corre-

sponding to the point cloud.

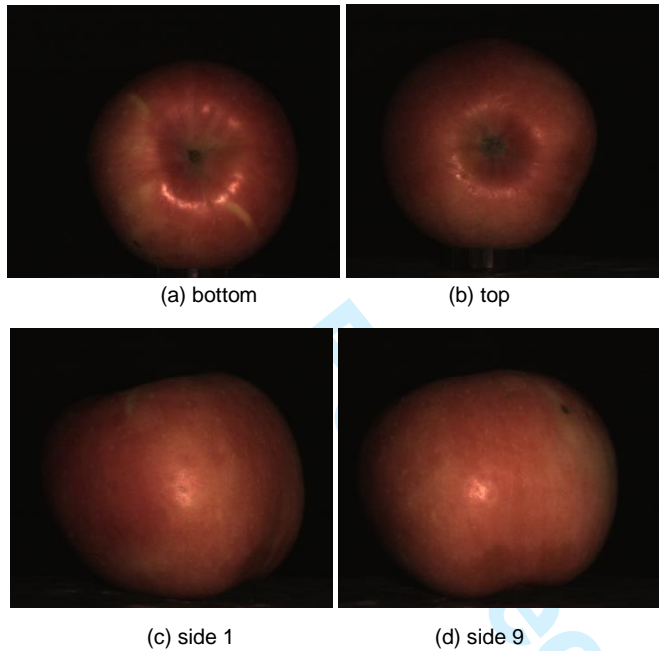


Fig. 2. Portion of the original image of the apple. Fig. 2a is the bottom, Fig. 2b is the top, Fig. 2c is side 1, and Fig. 2d is side 9.

4.1 Texture simplification

Every range was scanned from the right, and its outside surface was relatively flat (Fig. 3a). Therefore, the Z value range identifying depth was small, which turns the convex hull of the point cloud into a thin cuboid along the z-axis (Fig. 3b). At the same time, when viewed along the z-axis, the surface is convex without self-intersection; thus, the position of each point can be determined by the two-dimensional coordinates (x_i, y_i) . As a result, it was possible to directly extend the point-octree partitioning algorithm to a point-quadtrees algorithm for texture division by shielding the z component.

4.1.1 Adaptive texture segmentation

The proposed adaptive point-quadtrees segmentation is an im-

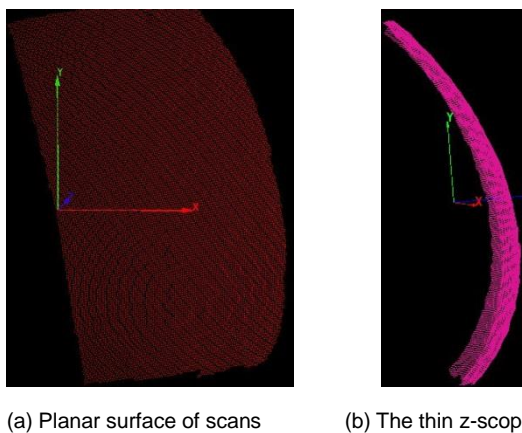


Fig. 3. Features of fruit scans. Fig. 3a presents the planar surface of the scan, and Fig. 3b presents its thin scope in the z-direction.

provement of the conventional quartering concept [Error! Reference source not found., Error! Reference source not found.]. Our adaptive segmentation process $CQT(C, T)$ takes the segmentation parameter of the point cloud as a split condition and the point set and texture in Eq. (4-1) as input data. It performs a top-down asymmetrical recursive process by utilizing the divi-

sion position p_{split} produced by point-octree division [Error! Reference source not found.]. In each loop, we developed the following four-step segmentation process. First, the first two parameters of split coordinates $p_{split}(x, y, z)$ are passed to the texture split position $t_{split}(x, y)$. Then, four empty subsets T_j are prepared for the current segmentation, which is used to save four parts of the pixel set: the upper left, lower left, upper right and lower right. Next, a search among all of the pixels in the current set using an interactive process pushes each pixel into one of the four subsets by comparing the parameter relationships. Finally, each non-empty subset is returned and prepared as a new quadtree node for the next iterating split. The pseudo

```

CQT(C, T) loop
{
  If (Leaf_node(C) is true) then
    put Leaf_node(T) into bucket  $B_T$ 
   $t_{split} \leftarrow p_{split}$ 
  initial  $T_j = \emptyset$  ( $j=0, 1, 2, 3$ )
  for  $i = 1$  to  $|T|$  //classify the pixel into 4
     $r_{split} \leftarrow MAX(t_{split} - pix_i)$ 
     $j \leftarrow Get\_cqt\_code(t_{split}, pix_i)$ 
     $T_j \leftarrow T_j \cup pix_i$ 
  for  $j = 0$  to 3 //records the non-null set
    if  $T_j \neq \emptyset$  then
       $c_j \leftarrow Cpt\_code(T_j)$ 
  return New_cqt_node( $t_{split}, r_{split}, c_{0..3}$ )
}

```

Fig. 4. Pseudo code of texture subdivision. This is the adaptive segmentation process $CQT(C, T)$.

code of this process is shown in Fig. 4.

In the above pseudo code, T_j is the pixel set of the j^{th} quadtree node in the current loop, and r_{split} is the radius of each node in the current segmentation. Leaf_node (T) is a process that converts pixel set T to a leaf node. Get_cqt_code (t_{split}, pix_i) produces a quarter code in the 0-3 range for any pixel pix_i according to the current split position t_{split} , which is used to distinguish quarter areas. Cpt_code (T_j) generates a new serial number c_j ($0 \leq c_j \leq 3$) for those nonempty pixel subsets T_j produced by segmentation. New_cqt_node ($t_{split}, r_{split}, c_{0..3}$) calls the $CQT(C, T)$ function to generate a new quadtree node according to the location, radius and serial number. Eventually, the texture will be adaptively divided into a series of subsets, and all of the sets for the leaf nodes are recorded as basket B_T . This segmentation algorithm evades the complex process of recalculating the parameters and greatly simplifies the segmentation process utilizing surfel-based geometric parameters.

4.1.2 Definition of the texton descriptor

According to a theory by B. Julesz and S. Magda, the texton is defined as the attribute of an image block [Error! Reference source not found., Error! Reference source not found.]. The above quadtree segmentation results for texture can be considered texton sets, in which each texton is a set of several pixels belonging to the same area. We derived the following conclusion on data features by analyzing the above segmentation. First, texture is segmented based on geometric feature points, with the additional requirement of two different expression spaces: shape and color. Then, the result of texture segmentation will cover all of the textures with the leaf nodes. Therefore, we calculated the texton descriptor using the leaf node data

contained in the bucket in Eq. (4-2)

$$[B_{sha}, B_{tex}] = \{P_j, T_j\} \quad j = 1, 2, \dots, n \quad (4-2)$$

in which B_{sha} is a basket of surfels, geometric leaf node; B_{tex} is basket of textons, texture leaf node; P_j is surfels among baskets B_{sha} ; T_j is the textons among basket B_{tex} ; and n is the number of leaf nodes in the basket. Each position of baskets has a texton index (texton tag) j , in which the local neighborhood of pixels are saved as Eq. (4-3)

$$T_j = \{pix_{j,1}, \dots, pix_{j,m}\} \quad (4-3)$$

in which $pix_{j,i}$ is the i^{th} pixel of the j^{th} texton and M is the number of pixels in the neighborhood of the j^{th} leaf node.

To reduce the search time for future steps, the geometric distance between pixels within texton T_j was calculated in advance and saved to a two-dimensional array $D_j[m][m]$ in Eq. (4-4).

$$D_j[i][k] = \sqrt{(x_{pt(j,i)} - x_{pt(j,k)})^2 + (y_{pt(j,i)} - y_{pt(j,k)})^2} \quad (4-4)$$

in which $1 \leq i, k \leq m$, $D_j[i][k]$ is the position of a pixel.

For the convenient calculation of the texton attribute in the following steps, the texton descriptor τ is defined in four dimensions according to the location and color attribute of the pixels within a neighborhood and is denoted by Eq. (4-5)

$$\tau = \{D_j[i][k], pix_{j,i}(R, G, B)\} \quad (4-5)$$

$$j = 1, 2, \dots, n, i = 1, 2, \dots, m.$$

4.2 Transformation of color perception traits

With the previous simplification and descriptor definition, each texton contains a number of discrete pixels denoted as m , which might produce visual cracks between different textons. In the following section, we will describe a smooth texture synthesis method with high storage performance by utilizing the visual perception principle.

There is no common consensus on how to perform a color or texture analysis in the perceptual sense. A simpler method is to address the $c(R,G,B)$ color value by directly employing the calculation method of three-dimensional geometric coordinates. Furthermore, the mean value $\mu(c)$, principal component $\Psi(c)$, and standard deviation $\sigma(c)$ are all calculated using this PCA method. RGB color space can also be converted to the uniform color space HSV(Chroma, saturation, purity) containing more perceptual characteristics before performing PCA. Nevertheless, the calculation process will be more complex due to the sensory evaluation contained in the color and texture space. Thus, this method is not suitable for real-time rendering, cannot consider geometric relationships and cannot discard the correlation. If we consider the position and direction, the complex PCA analysis would be upgraded to eight dimensional vectors in Eq. (4-6), which will result in a more complex computation [Error! Reference source not found.].

$$P_i = \{x_i, y_i, z_i, \theta_i, \phi_i, r_i, g_i, b_i\} \quad (4-6)$$

In this paper, human visual properties were adopted to reduce data size. According to the color theory of R. W. G. Hunt, human visual perception intensity to red, blue, and green is 40:20:1, which has been used correspondingly to adjust the quantization level and compress the color code, improving efficiency and saving space [Error! Reference source not found.]. S. Rusinkiewicz et al. predicted the color from the par-

ent node in the hierarchy and saved additional space by quantization of red, green, and blue for the 5-6-5 bit, respectively [Error! Reference source not found.]. To further compress the data volume, we decreased the memory space of the image by using the human eye's visual principle, which states that the resolving power of the human eye to color details is far lower than to brightness details [Error! Reference source not found.]. We transformed the fruit texture from RGB color components to an YCrCb digital display system, which contained one brightness component and two chrominance components, as shown in Eq. (4-7).

$$Y = 0.299R + 0.587G + 0.114B \quad (4-7a)$$

$$Cr = (0.500R - 0.4187G - 0.0813B) + 128 \quad (4-7b)$$

$$Cb = (-0.1687R - 0.3313G + 0.500B) + 128 \quad (4-7c)$$

As a result, we received a grayscale composed of the luminance signal and two monochromatic images composed of chroma signals. Then, these independent monochrome images will be encoded, and consequently, the resolution of the color components can be reduced without significantly affecting the image quality. For convenience of description, we will refer to the YCrCb as Lab in the subsequent steps.

4.3 Texture space compression scheme

After the color perception transformation, each texton T_j including m pixels can be seen as a planar array with $3*m$ values. Within each texton, each pixel is made of three components, as in Eq. (4-8), thus requiring excessive space to store all of the image information.

$$T_j = \{L_{j,1}, a_{j,1}, b_{j,1}, \dots, L_{j,m}, a_{j,m}, b_{j,m}\} \quad (4-8)$$

Our space algorithm is mainly presented in the following two parts using the texton symbols T_j and $pix_{j,i}$ in Eq. (3-5b) as description notation.

4.3.1 Texton subdivision

Other than unexpected injuries, the fruit surface textures are typically continuous in a local area. At the same time, according to the apple's geometric splitting [Error! Reference source not found.] and texture segmentation rule in 4.3, the average number of pixels contained within textons is 57.1 (Table S11). The pixel quantity on larger surface curvatures is less than this value, whereas smaller curvatures contains more pixels than average. From the scanning parameters in Section 3, the size of each texton is a square with a side length of approximately 1.16 mm, and the texture distortion or mosaic will appear on a texton of this size if it is presented in one color.

Therefore, we calculate texton color by applying the three components, calculating each color component separately in the same manner, and the difference among the neighbor colors is noted by a symbol. Considering the texture authenticity and the complexity of pixel-by-pixel calculation, a neighborhood-based subdivision within textons is necessary. The current neighborhood methods are poor at determining the neighbor edges due to the assumption that the pixel color depends on the probability distribution of the adjacent pixel values. In this paper, we improved the neighborhood methods by classifying the neighbor pixels based on visual perception. Concretely, we subdivide the pixels within texton T_j into k classes depending on the sensitivity of the human eye relative to signal by clustering pixels in Eq. (4-8) based on their luminance components. After subdivision, these textons are denot-

ed as second-level textons in Eq. (4-9a) and shown in Eqs. (4-9b) to (4-9d).

$$T_j = \{T_{j1}, T_{j2}, \dots, T_{jk}\} \quad (4-9a)$$

$$T_{j1} = \{L_{j1,1}, a_{j1,1}, b_{j1,1}, \dots, L_{j1,n_1}, a_{j1,n_1}, b_{j1,n_1}\} \quad (4-9b)$$

$$T_{j2} = \{L_{j2,1}, a_{j2,1}, b_{j2,1}, \dots, L_{j2,n_2}, a_{j2,n_2}, b_{j2,n_2}\} \quad (4-9c)$$

...

$$T_{jk} = \{L_{jk,1}, a_{jk,1}, b_{jk,1}, \dots, L_{jk,n_k}, a_{jk,n_k}, b_{jk,n_k}\} \quad (4-9d)$$

As a result, all of the m pixels within textons in Eq. (4-8) are classified into k groups, and the pixel numbers of all of the groups are denoted as n_1, n_2, \dots, n_k in Eq. (4-10).

$$m = n_1 + n_2 + \dots + n_k \quad (4-10)$$

4.3.2 Sub-sampling scheme design

If we adopted whole sampling methods, the average storage of each pixel is the same with RGB mode because each pixel also has three components: luminance (L), Color 1 (a), and Color 2 (b). Therefore, the sub-sampling method is performed on the secondary-level texton T_{ji} according to the perception principles in Section 4.2. Currently, by taking 4 continuous pixels as a basic unit, there are 3 types of common sub-sampling, 4-2-2, 4-1-1 and 4-2-0 [Error! Reference source not found.]. In 4-2-2 format, two pixels are sampled on three components, the other two pixels are sampled only on the L component, the total number of components is 8, and the average components per pixel is 2, as shown in Fig. 5a. In 4-1-1 format, one pixel is sampled on three components, the other three pixels are sampled only on the L component, the total number of components is 6, and the average components per pixel is 1.5, as shown in Fig. 5b. In 4-2-0 format, four pixels are all sampled on three components, each pixel saves its own L component separately, the two color components are calculated separately by the sampled

colors of four pixels, the total number of components is 6, and the average components per pixel is 1.5, as shown in Fig. 5c.

To save additional texton memory space, we designed a new weighted sub-sampling scheme $n_i:2:0$ on the secondary textons T_{ji} . In this scheme, the luminance signal and color difference signals are represented as a different frequency. To reduce the required memory capacity, the critical pixel is sampled on three components, luminance (L), Color 1 (a), and Color 2 (b), and non-critical pixels are only represented by their own luminance components, i.e., their two color values are represented the same as the others. All of the pixels are sampled on three components, each pixel saves its own L component separately, and a and b components are mixed from n_i neighboring pixels within the texton T_{ji} , as shown in Fig. 5d.

This sub-sampling method converts secondary textons from the form in Eq. (4-9) to that in Eq. (4-11).

$$T_{ji} = \{L_{ji,1}, L_{ji,2}, \dots, L_{ji,n_i}, a_{ji}, b_{ji}\} \quad (4-11)$$

The total dimension is $n_i + 2$, and the average number of components h_{ji} required by each pixel in T_{ji} is computed in Eq. (4-12).

$$h_{ji} = (n_i + 2)/n_i \quad (4-12)$$

Accordingly, the average number of components h_j required by each pixel in T_j is computed by Eq. (4-13).

$$h_j = \sum_{i=1}^k (n_i + 2) / \sum_{i=1}^k n_i = (m + 2k)/m \quad (4-13)$$

Thus, the storage compression ratio τ_{ji} of T_{ji} and storage compression ratio τ_j of T_j were computed, respectively in Eq. (4-14a) and Eq. (4-14b),

$$\tau_{ji} = 3/h_{ji} = 3n_i/(n_i + 2) \quad (4-14a)$$

$$\tau_j = 3/h_j = 3m/(m + 2k) \quad (4-14b)$$

in which n_i, k and other parameters are referenced in Eq. (4-10).

By analyzing Eq. (4-14), we made the following conclusion.

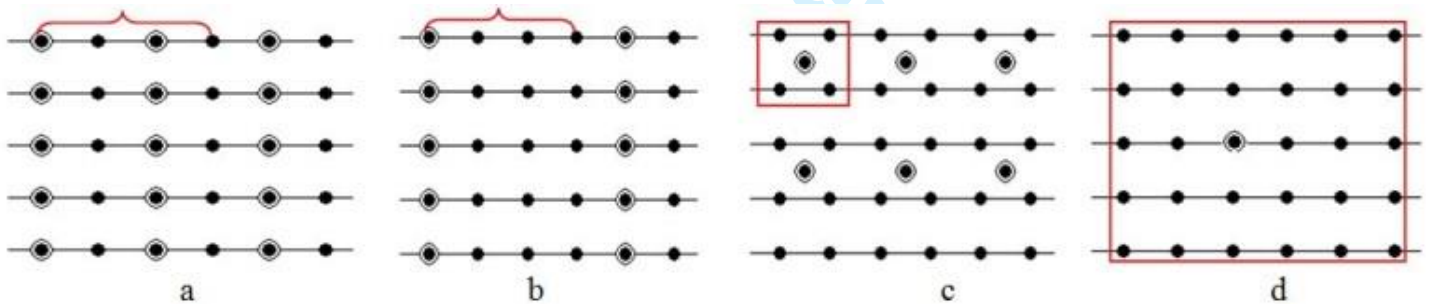


Fig. 5. Four types of subsampling modes. L, a, b--component, ● only L--component. Fig. 5a shows 4-2-2 sampling, Fig. 5b is 4-1-1 sampling, Fig. 5c is 4-2-0 sampling, and Fig. 5d is n_i-2-0 sampling.

For each secondary texton, the compression ratio is inversely proportional to the number of pixels within a texton; therefore, a higher number of pixels results in a higher compression ratio. In the first-level textons, the compression ratio is also inversely proportional to the number of pixels within a texton, whereas it is proportional to the number of secondary textons within a first-level texton. This means that the number of secondary textons should be as small as possible without decreasing the quality of the texture. Therefore, an appropriate number of secondary textons is more important for ensuring the balance between saving memory space and texture quality.

4.4 Authentic texture blending

In the above sub-sampling method, it was necessary to cal-

culate two color components for each secondary texton T_{ji} . The values of pixels within the region are calculated by averaging each color component in accordance with Eq. (4-15).

$$a_{ji} = \sum_{l=1}^{n_i} a_{ji,l} / n_i \quad (4-15a)$$

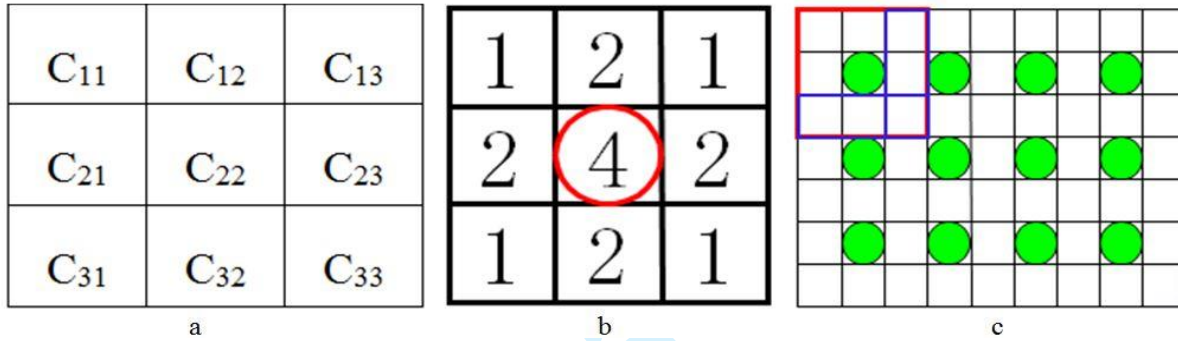
$$b_{ji} = \sum_{l=1}^{n_i} b_{ji,l} / n_i \quad (4-15b)$$

These types of pixel value estimation methods are confined to the Markov random field (MRF) model for determining the color of texels. However, there is a "chicken-and-egg" problem, namely, pixels A and B might exist in each other's neighborhood such that the probability distribution of A depends on B and vice versa. In this paper, we improved the Markov random field model by estimating neighbor pixel values based on a

weighted template and overlapping oversampling to solve the interdependence between neighbor pixels.

Nevertheless, under normal circumstances, those pixels closer to the centroid have a greater contribution to the color values of the second-level textons, so the direct averaging of MRF will result in color distortion. To represent the authenticity of fruit color more reasonably, we employed a weighted average based on distance such that the weight of a pixel is inversely proportional to its distance from the centroid. Under the assumption of a 3*3 pixel area (larger or smaller than this size, it can be extended based on the template), we will sample the color component values of each pixel as shown in Fig. 6a. Then, the color components throughout T_{ji} are calculated using the rules of Eq. (4-16) according to the component value (Fig. 6a) and weighting template (Fig. 6b).

Fig. 6. Weighted oversampling of textons. Fig. 6a shows the color component values of sampling, Fig. 6b is a weighted mask, and Fig. 6c is the over-



lapping sampling pattern.

Under the subdivision result of Section 4.3.1, the first-level texton size is approximately 0.58 mm. If each pixel retains a luminance component, the luminance resolution approximates to 0.058 mm. It might exceed the discernibility of the human eye, which cannot distinguish at such precise a level [Error! Reference source not found.]. To further reduce the amount of data, this paper designed the following compression scheme based on the perceptual system by mixing a luminance value for each first-level texton and a chromaticity value for each second-level texton. First, calculate the luminance component L_{ji} of each secondary texton T_{ji} based on Eq. (4-16). Next, compute chrominance components for first-level textons based on the two color components a_{ji} and b_{ji} of the second-level textons. Then, calculate the chrominance components a_j and b_j for first-level textons using Eq. (4-16) while keeping smooth transitions in color and luminance by overlapping weighted sampling (Fig. 6c). Finally, the properties of texton T_j are stored in Eq. (4-17).

$$T_j = \{L_{j1}, L_{j2}, \dots, L_{jk}, a_j, b_j\} \quad (4-17)$$

The total number of dimensions is $k+2$, and the average number of components h_j required by each pixel in T_j is shown in Eq. (4-18).

$$h_j = (k + 2)/m \quad (4-18)$$

The average number of components per pixel h_j is far less than that in Eq. (4-13). Thus, the τ_j of each T_j is shown in Eq. (4-19).

$$\tau_j = 3/h_j = 3m/(k + 2) \quad (4-19)$$

The storage compression ratio τ_j is far less than that in Eq. (4-14b).

$$\frac{C_{22}}{4} + \frac{(C_{12} + C_{21} + C_{23} + C_{32})}{8} + \frac{(C_{11} + C_{13} + C_{31} + C_{33})}{16} \quad (4-16)$$

However, the above calculation methods will lead to the same motif color within one texton, and a distinct gap between two adjacent texton is easily produced that is not sufficient for a smooth transition. We implemented a smooth color transition between adjacent textons based on overlapping region sampling. In addition to the pixels within the neighborhood, the pixels on the boundary of other textons adjacent to the current texton are sampled in the row and column directions for the color calculation of the current texton, as shown in Fig. 6c, in which the pixels in the blue rectangle are the overlapping boundary regions between current and adjacent textons.

5 RESULTS AND COMPARISON

5.1 Experiment platform

We simulated the texture synthesis for the apple point in the following environment.

Hardware environment:

CPU: Intel(R) Core(TM) 2 Duo CPU P7350 @2.00GHz

RAM: 2.00GB

Graphics card: NVIDIA Geforce GT 130M

Software platform: OpenGL, VS2010, Matlab 2010

5.2 Simplification test of apple texture

We split the original texture (Fig. 2, Fig. S1) at the pixel level according to our segmentation strategy from Section 4.1.1, and the segmentation results are shown in Fig. 7 as a series of textons sets.

The final division coordinates corresponding to Fig. 7 are shown in Table 1, which only lists apple side S1 and the bottom B in Eq. (3-2).

The results show that the maximum texton number produced by quadtree segmentation on apple textures is 79*80, far less than the octree division number of $2^6 \times 2^6 \times 2^6$, meaning that our methods have simplified the segmentation result and decreased cell number.

5.3 Texture synthesis of the apple sample

5.3.1 Texture feature evaluation

By randomly choosing a texton (in row 45, column 11) from apple texture S1 as an example, we evaluated its properties according to the proposed SPSPDW algorithm. The interior pixel number m is 56, and the texton size is 7*8 pixels, expressed as

$\text{pix}_{45,11,i}, i = 1, 2, \dots, 55, 56$. The RGB distribution is shown in

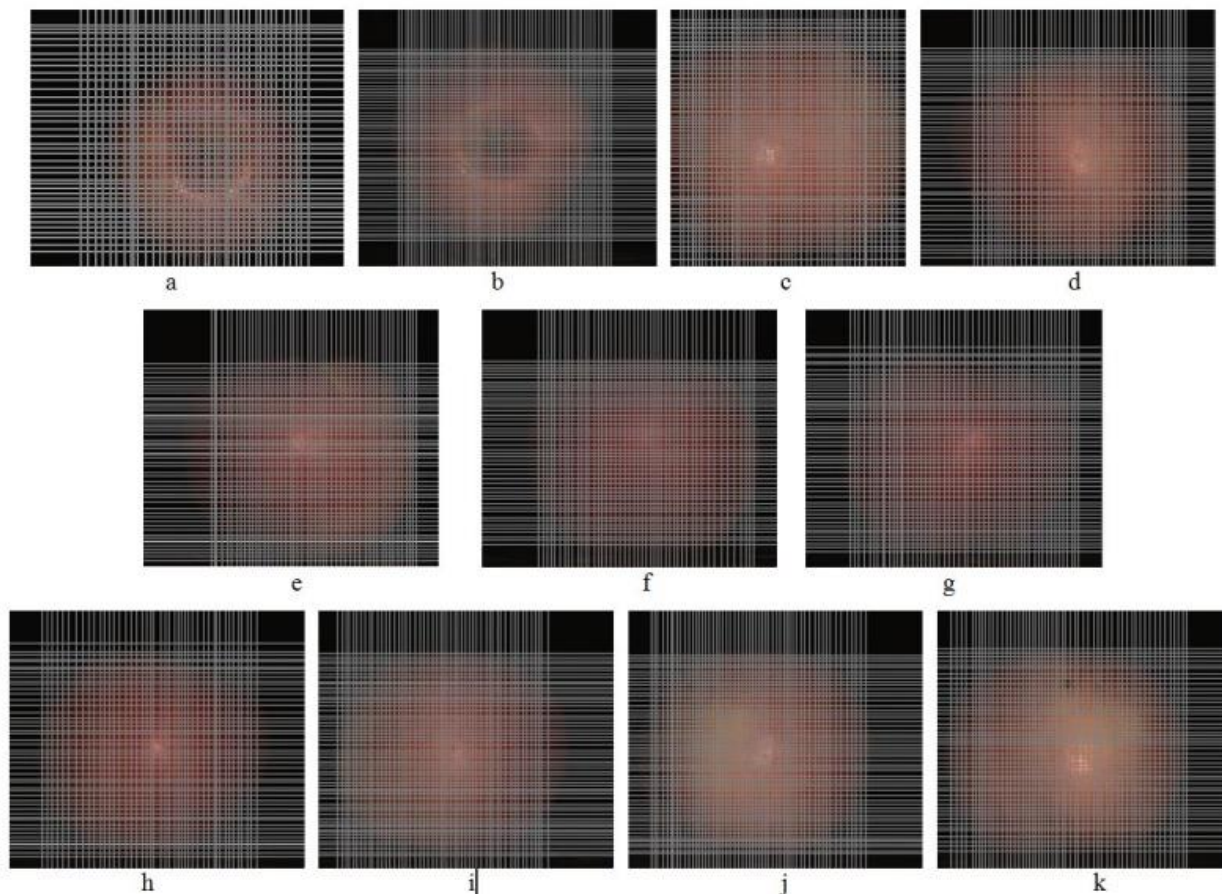


Table 2.

Fig. 7. Texture segmentation scale of the apple surface (row*column). Fig. 7a is the bottom, and its segmentation result was (79*80). Fig. 7b is the top, and its segmentation result was (58*68). Fig. 7c to Fig. 7k are the 9 sides, and its segmentation results were (59*67, 50*49, 54*51, 49*49, 52*54, 51*46, 60*54, 57*58, and 50*58).

TABLE 1. PART OF SEGMENTED POSITIONS OF S1 AND B OF APPLE

Texture image	Division axis	Division coordinates
S1 of apple	X-axis	0,78,82,88,89,94,98, 103,107,..., 352,359,365,440
	Y-axis	0,44,49,55,61,68,74,79,84,91,95,..., 338,345,350,383
B of apple	X-axis	0,30,31,36,37,43,44,54,55,..., 406,407,419,420,431,432,459
	Y-axis	0,92,93,105,106,119,120,131,132,..., 482,484,496,498,560

TABLE 2. RGB VALUE OF TEXTON $T_{45,11}$

Y	X	0	1	2	3	4	5	6
	0		(107,46,31)	(104,47,31)	(104,47,36)	(108,48,36)	(108,52,37)	(114,57,37)
1		(102,45,31)	(106,46,31)	(106,49,36)	(111,49,36)	(111,51,37)	(112,57,37)	(112,59,38)
2		(102,43,31)	(106,45,31)	(106,48,36)	(111,49,36)	(111,51,63)	(112,54,34)	(112,56,37)
3		(96,43,31)	(109,45,31)	(109,47,36)	(109,47,36)	(109,49,34)	(113,52,34)	(113,54,37)
4		(96,44,31)	(109,45,31)	(109,46,31)	(109,46,31)	(109,48,34)	(113,49,34)	(113,52,38)
5		(104,45,31)	(102,46,31)	(102,44,31)	(105,44,31)	(105,47,34)	(108,48,34)	(108,51,38)
6		(104,45,31)	(102,46,31)	(102,45,33)	(105,45,33)	(105,47,32)	(108,49,32)	(108,52,32)
7		(97,43,31)	(104,45,31)	(104,46,33)	(104,46,33)	(106,47,32)	(110,49,32)	(110,50,32)

In step 1, the texton is transformed from an RGB-based representation to a Lab representation by using Eq. (4-7), and the transformation results corresponding to Table 2 are shown in Table 3.

TABLE 3. LAB VALUE OF TEXTON $T_{45,11}$

X \ Y	0	1	2	3	4	5	6
0	(62,110,159)	(62,110,157)	(62,112,157)	(64,111,158)	(67,111,157)	(71,108,158)	(72,108,157)
1	(60,111,157)	(62,110,159)	(64,111,157)	(66,111,160)	(67,110,159)	(71,108,157)	(72,108,156)
2	(59,112,158)	(61,110,159)	(63,112,157)	(66,111,160)	(67,109,159)	(69,108,158)	(70,109,157)
3	(57,113,155)	(62,110,161)	(64,112,159)	(64,112,159)	(65,110,159)	(68,108,159)	(69,109,158)
4	(58,112,155)	(62,110,161)	(63,109,160)	(63,109,160)	(64,110,159)	(66,109,161)	(68,110,159)
5	(61,111,158)	(61,111,157)	(59,111,158)	(60,111,159)	(62,111,158)	(64,110,159)	(66,111,157)
6	(61,111,158)	(61,111,157)	(60,112,157)	(61,111,158)	(62,110,158)	(64,109,158)	(66,108,157)
7	(57,112,155)	(61,111,158)	(61,111,158)	(62,111,159)	(62,110,158)	(65,109,159)	(65,108,159)

In step 2, the pixels within each texton were classified into 12 categories based on our visual perception clustering. $T_{45,11}$ is subdivided into textons with $k=12$ in Eq. (4-9), and the number of pixels contained within each secondary texton is shown in Eq. (5-1).

$$n_1 = n_2 = n_9 = n_{11} = n_{12} = 5 \quad (5-1a)$$

$$n_3 = n_4 = n_6 = n_{10} = 4 \quad (5-1b)$$

$$n_5 = n_9 = 6 \quad (5-1c)$$

$$n_7 = 3 \quad (5-1d)$$

The subdivided texton structure is shown in Fig. 8a, in which different colors represent different secondary textons.

In step 3, by applying our subsampling scheme to each secondary texton, we obtain the component structures of secondary textons as in Eq. (4-11) with $j=45$, $i=11$ and $k=1 \dots 12$. Then, the calculated average dimensions of secondary texton components are $\{7,7,6,6,8,6,5,8,7,6,7,7\}$. By substituting this value into Eq. (4-12), the average components $h_{45,11,i}$ of all pixels can be evaluated as in Eq. (5-2).

$$h_{j,1} = h_{j,2} = h_{j,9} = h_{j,11} = h_{j,12} = (5 + 2)/5 = 1.4 \quad (5-2a)$$

$$h_{j,3} = h_{j,4} = h_{j,6} = h_{j,10} = (4 + 2)/4 = 1.5 \quad (5-2b)$$

$$h_{j,5} = h_{j,9} = (6 + 2)/6 \approx 1.3 \quad (5-2c)$$

$$h_{j,7} = (3 + 2)/3 \approx 1.7 \quad (5-2d)$$

Then, by substituting $h_{45,11,i}$ into Eq. (4-9a), the compression ratios $\tau_{45,11,i}$ of secondary textons are derived as in Eq. (5-3).

$$\tau_{j,1} = \tau_{j,2} = \tau_{j,9} = \tau_{j,11} = \tau_{j,12} = 3/1.4 \approx 2.14 \quad (5-3a)$$

$$\tau_{j,3} = \tau_{j,4} = \tau_{j,6} = \tau_{j,10} = 3/1.5 = 2 \quad (5-3b)$$

$$\tau_{j,5} = \tau_{j,9} = 3/1.3 \approx 2.25 \quad (5-3c)$$

$$\tau_{j,7} = 3/1.7 \approx 1.76 \quad (5-3d)$$

in which j is the expression 45, 11.

At the same time, this has verified the conclusion in Eq. (4-14a) that secondary textons with more pixels, such as $T_{45,11,5}$ and $T_{45,11,9}$, will have a higher compression ratio. Finally, by substituting $m=56$ and $k=12$ into Eqs. (4-13) and (4-14b), re-

spectively, we can derive the average component number $h_{45,11}$ per pixel and the compression ratio $\tau_{45,11}$ of the first-level texton $T_{45,11}$ as shown in Eq. (5-4).

$$h_{45,11} = (56 + 2 * 12)/56 = 1.43 \quad (5-4a)$$

$$\tau_{45,11} = 3 * 56 / (56 + 2 * 12) = 2.1 \quad (5-4b)$$

In step 4, the mixing property value of each texton is determined.

First, we evaluate the property for each secondary texton whose segmentation result is shown in Fig. 8a. According to the oversampling template (Fig. 6c), the attribute areas for each secondary texton were extended to the overlapping formation in Fig. 8b. The interlapping 3×3 rectangles show a different color representing the sampling area of secondary textons. Based on the weighted template (Fig. 6), color components of the pixels in Table 2 were substituted into Eq. (4-16), and we deduced Lab attribute values for each secondary texton, as shown in the first eight rows of Table 4.

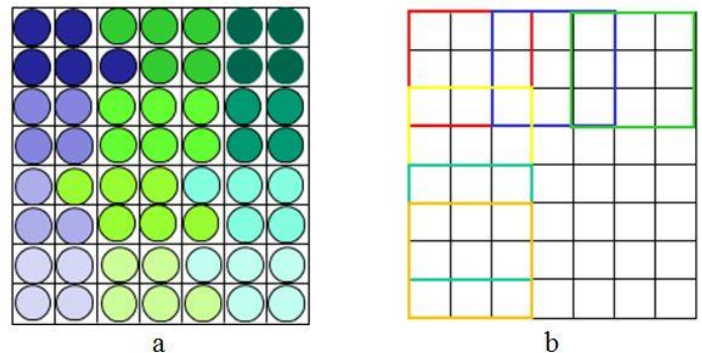


Fig. 8. Sub-texton and oversampling of $T_{45,11}$. Fig. 8a is a sub-texton of $T_{45,11}$, and Fig. 8b shows the oversampling mode of sub-textons.

Next, the attributes of first-level textons in Eq. (4-17) are evaluated based on the above properties of secondary textons to further reduce storage. The luminance components of 12 secondary textons $T_{45,11,i}$ were first evaluated according to Eq. (4-16) and Table 3, as shown in the last row of Table 4. Next,

TABLE 4. LAB VALUE OF SUB-TEXTON

T_{ji} Lab	T_{j1}	T_{j2}	T_{j3}	T_{j4}	T_{j5}	T_{j6}	T_{j7}	T_{j8}	T_{j9}	T_{j10}	T_{j11}	T_{j12}
$L_{ji,1}$	62	62	71	59	63	69	58	62	64	61	60	62
$L_{ji,2}$	62	64	72	61	66	70	61	63	66	61	61	64
$L_{ji,3}$	60	67	71	57	67	68	61	63	68	57	61	66
$L_{ji,4}$	62	66	72	62	64	69	-	59	64	61	61	65
$L_{ji,5}$	64	67	-	-	64	-	-	60	66	-	62	65
$L_{ji,6}$	-	-	-	-	65	-	-	62	-	-	-	-
a_{ji}	110.63	110.88	108.56	110.94	110.94	108.56	110.69	110.25	109.56	111.19	110.94	109.38
b_{ji}	157.94	158.63	157.5	158.38	159.0	158.0	158.75	159.25	159.25	157.31	158.13	158.13
L_{ji}	61.75	65.06	69.88	61.31	65.25	68.81	61.06	62.75	65.88	60.5	61.0	64.06

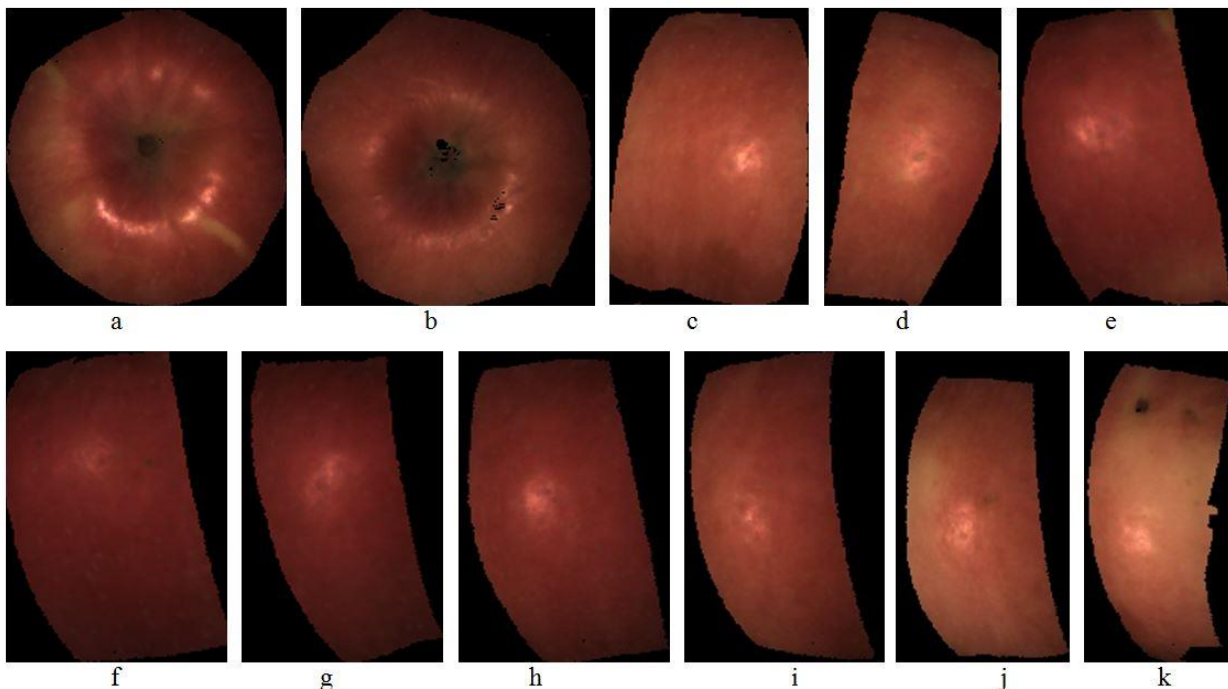


Fig. 9. Texture synthesis result of our SPSDW method on the apple surface. Fig. 9a is the bottom, Fig. 9b is the top, and Fig. 9c to Fig. 9k are the 9 sides.

chrominance components were calculated by applying a weighted average on a_{ji} and b_{ji} in Table 4, yielding the results $a_j = 26.1$ and $b_j = 23$. Finally, the value attributed to texton $T_{45,11}$ was derived in the form of 14 dimensional components as shown in Eq. (5-5).

For texton $T_{45,11}$, the average number of components per pixel and storage compression ratio are calculated based on Eqs. (4-18) and (4-19), respectively, in Eq. (5-6).

$$h_{45,11} = 14/56 = 0.25 \quad (5-6a)$$

$$\tau_{45,11} = (3 * 56)/14 = 12 \quad (5-6b)$$

The average number of components per pixel is 0.25, signifi-

$$T_{45,11} = \{ \{61.75, 65.06, 69.88, 61.31, 65.25, 68.81, 61.06, 62.75, 65.88, 60.5, 61, 64.06\}, 110.2, 158.4 \} \quad (5-5)$$

cantly less than the 1.43 calculated according to Eq. (5-4a). The compression rate is 2.1 far greater than 12, which was calculated according to Eq. (5-4b). By applying the proposed method on the surfel-based apple surface, we generated the texture synthesis result shown in Fig. 9. Nevertheless, this high compression rate is at the expense of losing image quality and thus should be deliberate when considering larger textons.

5.3.2 Comparison and analysis

(1) Qualitative comparison

We applied a direct two-dimensional mapping method [Error! Reference source not found.] and color adjustment

technology [Error! Reference source not found.] to the same apple's surface for texture synthesis, and the comparison results are shown in Figs. 10 and 11, respectively.

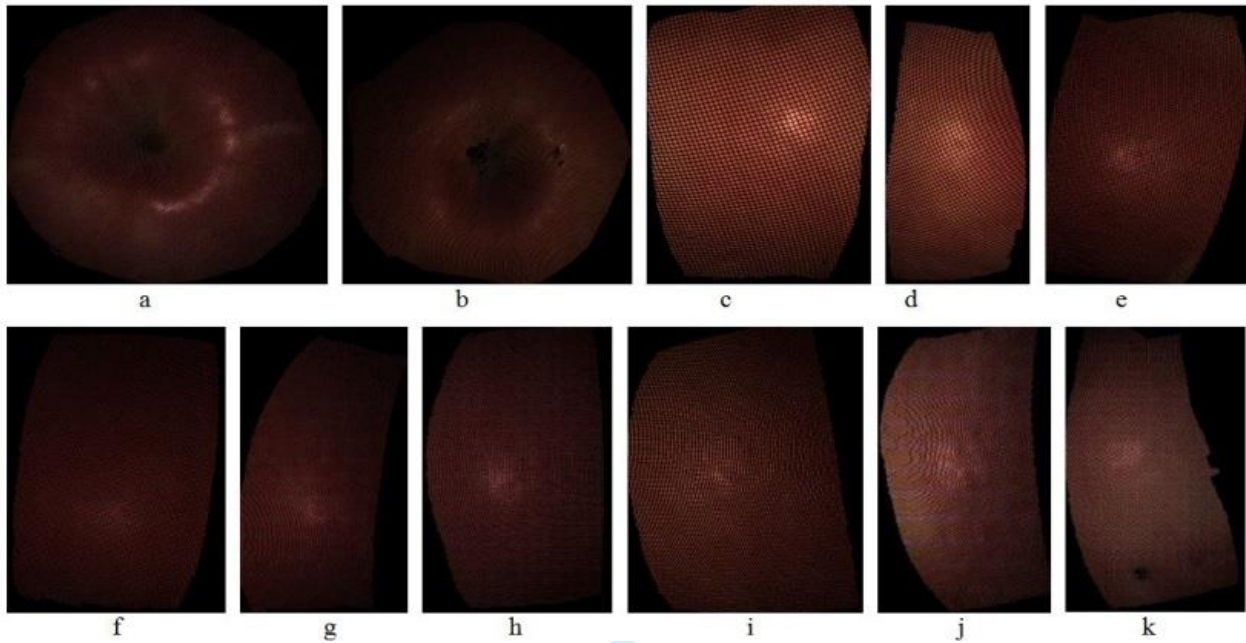
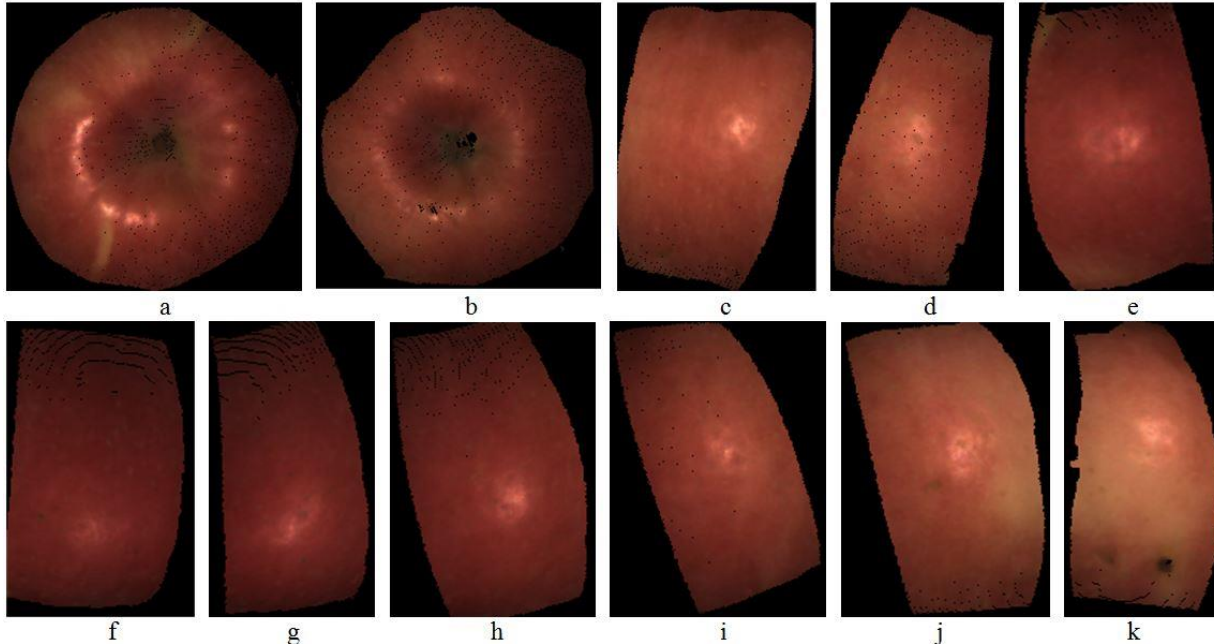


Fig. 10. Texture synthesis of the apple based on two dimensional mapping [Error! Reference source not found.]. Fig. 10a is the bottom, Fig. 10b is the top, and Fig. 10c to Fig. 10k are the 9 sides.

Fig. 11. Texture synthesis of the apple based on the color adjustment technique [Error! Reference source not found.]. Fig. 11a is the bottom, Fig. 11b



is the top, and Fig. 11c to Fig. 11k are the 9 sides.

By contrasting texture synthesis effects on apple scans between Figs. 10, 11 and 9, we made the following conclusions. The direct two-dimensional mapping method is simple, coarse, easily leads to "seams" and results in a discontinuous surface texture. Color adjustment techniques eliminate "seams" to a certain extent, and the visual effect approximates to our method, but the point cloud must be converted to a triangulation mesh in advance and is restricted by out-of-focus image areas.

(2) Quantitative analysis of performance

The direct two-dimensional mapping method mapped pixels to point clouds directly, and the required texture memory space was calculated in Eq. (5-7a) (Byte).

$$M_{DT} = p_{num} * 3 \quad (5-7a)$$

in which M_{DT} is the memory space of direct two-dimensional mapping and p_{num} is the number of pixels in a texture image.

The multi-resolution quantization method proposed by S. Rusinkiewicz and M. Levoy simplifies the red-green-blue

components to 5-6-5 digits [Error! Reference source not found.], and the required texture memory space is calculated in Eq. (5-7b) (Byte).

$$M_{MQ} = p_{num} * (5 + 6 + 5)/8 =$$

$$p_{num} * 2 \quad (5 - 7b)$$

in which M_{MQ} is the memory space for multi-resolution quantization.

R. W. G. Hunt proposed a color visual model, and the color code was quantized based on human visual perceptual intensi-

ty by setting the red-green-blue proportion to 40-20-1 [Error! Reference source not found.], and the required texture memory space is calculated in Eq. (5-7c) (Byte).

$$M_{CVM} = p_{num} * (1 + 1/20 + 1/40) = p_{num} * 1.075 \quad (5 - 7c)$$

in which M_{CVM} is the memory space of the color visual model.

According to texture size (Fig. 2, Fig. S1), the texture synthesis results in Section 5.3.1 and Eq. (5-7), we evaluated the memory space of two-level splitting in this paper, shown as M_{SPSDW1} and M_{SPSDW2} in Table 5.

TABLE 5. MEMORY SPACE OF APPLE TEXTURE(BYTE)

texture method	b	t	S1	S2	S3	S4	S5	S6	S7	S8	S9
p_{num}	257040	225792	168520	168520	168520	168520	168520	168520	168520	168520	168520
M_{DT}	771120	677376	505560	505560	505560	505560	505560	505560	505560	505560	505560
M_{MQ}	514080	451584	337040	337040	337040	337040	337040	337040	337040	337040	337040
M_{CVM}	276318	242726	181159	181159	181159	181159	181159	181159	181159	181159	181159
M_{SPSDW1}	299254	277342	210046	210046	207534	207534	212944	208418	209798	209924	214502
M_{SPSDW2}	33747	33663	28260	25022	25015	22968	27828	24641	27119	27314	28791

The comparison of memory space for different methods is shown in Fig. 12. We drew the following conclusions by analyzing the memory space required by apple textures in Table 5. On average, the first-level compression-based method saved 58.8% more memory space than the direct two-dimensional map and 38.1% more than multi-resolution quantitative methods but required 15.1% more for the visual color model. Our SPSDW-based secondary compression saved memory space by 94%, 92.3%, 85.7% than above methods, respectively. The average compression ratio of the first and second textons was 2.43 and 19.7, higher than the values of 1.0, 1.5 and 2.79 for the direct two-dimensional map, multi-resolution quantitative methods, and the color visual model, respectively.

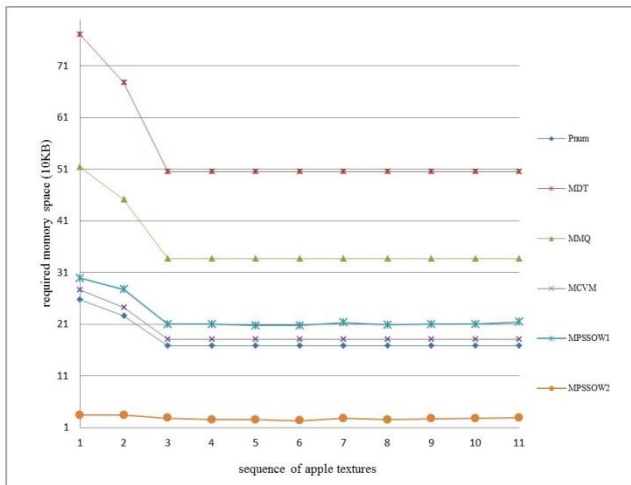


Fig. 12. Contrast of memory use for different compression algorithms.

p_{num} is the number of pixels in a texture image, M_{DT} is the memory space for direct two-dimensional mapping, M_{MQ} is the memory space for multi-resolution quantization, M_{CVM} is memory space for the color visual model, M_{SPSDW1} is the memory space for the first-level compression, and M_{SPSDW2} is the memory space for the secondary compression.

6 CONCLUSION

An SPSDW algorithm was proposed to ensure a balance between quality and space performance for point-based fruit texture synthesis, which solved the problems of the rough surface, low coding efficiency and large memory space required for current methods. First, the texture-splitting process was simplified by improving the quadtree, and the segmentation complexity declined from three to two dimensions. Then, our subdivision and down sampling methods made the average memory space of textons decrease to approximately 1/19.7 of the original image. Finally, the proposed weighted oversampling scheme improved the texture mixing authenticity compared with previous techniques, which also saved 94% more space than the direct two-dimensional mapping method, 92.3% more than the resolution of quantitative methods, and 85.7% more than the color vision model. Experimental results also showed that our texture synthesis algorithm was optimized for performance, quality and storage and can be extended to other fruits, providing theoretical and practical values for solving the authenticity and space efficiency of texture synthesis on a fruit surface.

However, in the texture synthesis process, the luminance compression from the secondary texton to the first level utilizes brightness similarity in the local area instead of using the human visual principle. Luminance distortion will appear if the splitting level is lower or there are more pixels in one texton. In the future, it might be advisable to add the size of the first-level texton to the luminance compression strategy.

ACKNOWLEDGMENT

The authors wish to thank Nan Geng, MeiliWang and the anonymous reviewers for their helpful feedback. This work was supported in part by a grant from the Fundamental Research Funds for the Central Universities (No. 2014YB067,

2452015199), grant from the National High Technology Research and Development Program of China (863 Program, No. 2013AA10230402), the National High Technology Research and Development Program of China (2013BAD15B02), Scholarship Council and Scientific Research Foundation for P.H.D from Northwest Agriculture & Forest University of China (2014BSJJ060). Also, Jian Chang is the corresponding authorship.

REFERENCES

- [1] Zhou R, Damerow L, Sun Y, Blanke MM. Using colour features of cv. 'Gala' apple fruits in an orchard in image processing to predict yield. *Precision Agriculture*. 2012;13: 568-580.
- [2] Cetinkaya N, Ercin D, Özvtan S, Erel Y. Quantification of applied dose in irradiated citrus fruits by DNA Comet assay together with image analysis. *Food Chemistry*. 2016;192: 370-373.
- [3] Lino ACL, Dal Fabbro IM, De Almeida C. Fruit surface topographic survey supported by a phase shifting projection moiré technique. In: Conference proceedings of the 9th latin - american meeting on optics, lasers and applications (OPTILAS), Vol. 992. Melville: AIP Publishing; 2008. pp. 1028-1033.
- [4] Rodkaew Y, Chuai-Aree S, Siripant S, Lursinsap C, Chongstitvatana P. Animating plant growth in L-System by parametric functional symbols. *Int J Intell Syst*. 2004;19: 9-23. doi: 10.1002/int.10147
- [5] Zhu Q, Huang W, Qu H, Huang Q, Zheng M. Fine-grained simulation of the plant stem growth process. *Transactions of the Chinese Society for Agricultural Machinery*. 2011;42(3):192-196. Available: CAB Abstracts, Ipswich, MA. Accessed January 27, 2016.
- [6] Dong HW. A review of mesh segmentation. *Journal Image and Graphics*. 2010;15: 181-193.
- [7] Stahlhut O. Extending natural textures with multi-scale synthesis. *Graph Models*. 2005;67: 496-517. doi: 10.1016/j.gmod.2005.01.006
- [8] Pan R, Taubin G. Color adjustment in image-based texture maps. *Graph Models*. 2015;79: 39-48. doi: 10.1016/j.gmod.2015.04.002
- [9] Kumar S, Chagné D, Bink MC, Volz RK, Whitworth C, Carlisle C. Genomic selection for fruit quality traits in apple (*Malus domestica* Borkh.). *PLOS ONE*. 2012;7: e36674. doi: 10.1371/journal.pone.0036674
- [10] Portis E, Cericola F, Barchi L, Toppino L, Acciarri N, Pulcini L, et al. Association mapping for fruit, plant and leaf morphology traits in eggplant. *PLOS ONE*. 2015;10: e0135200. doi: 10.1371/journal.pone.0135200
- [11] Koyuncu B, Kullu K. Development of an optical 3D scanner based on structured light. In: Proceedings of the 9th WSEAS international conference. Stevens Point: WSEAS; 2010. pp. 17-22.
- [12] Colombo C, Comanducci D. A desktop 3D scanner exploiting rotation and visual rectification of laser profiles. In: Proceedings of the 4th IEEE international conference. Piscataway: IEEE; 2006. pp. 49.
- [13] Sithole G, Vosselman G. Automatic structure detection in a point cloud of an urban landscape. In: Remote sensing and data fusion over urban areas 2nd joint workshop. Piscataway: IEEE; 2003. pp. 67-71.
- [14] K.Nanaa, M.Rizon, A.M.N.Rahman, Y.Ibrahim, A.Aziz. Detecting Mango Fruits by Using Randomized Hough Transform and Backpropagation Neural Network. *Information Visualisation (IV)*. 2014; 388-391. doi:10.1109/IV.2014.54
- [15] Vincent G, Harja D. Exploring ecological significance of tree crown plasticity through three-dimensional modelling. *Ann Bot*. 2008; 101: 1221-1231. doi: 10.1093/aob/mcm189
- [16] Rosell JR, Llorens Jordi S, Rosell JR, Llorens J, Sanz R, Arnó J, et al. Obtaining the three-dimensional structure of tree orchards from remote 2D terrestrial LiDAR scanning. *Agric For Meteorol*. 2009; 149: 1505-1515. doi: 10.1016/j.agrformet.2009.04.008
- [17] Gross M, Pfister H. Point-based graphics. San Francisco: Morgan Kaufmann Publishers; 2007.
- [18] Hui-jun Y, He Dong-jian*, Shi Peng. A simplified and accurate registration for splat-based fruit scans. *ICIC Express Letters, Part B: Application*. 2012; 3: 733-741.
- [19] Yang H, He D, Jiang S, Wang H. Research on simplification and extraction technology of apple point cloud. *J of Applied Sciences*. 2013; 13: 3684-3690. doi: 10.3923/jas.2013.3684.3690
- [20] Lin H W, Tai C L, Wang G J. A mesh reconstruction algorithm driven by an intrinsic property of a point cloud. *Computer-Aided Design*. 2004; 36(1): 1-9. Doi:10.1016/S0010-4485(03)00064-2.
- [21] Jen-Tse W, Peng-Cheng W, Shyr-Shen A. A PCA and Perturb Based Fragile Watermarking Scheme for 3D models. *Int J Innovative Comput, Inf Contr*. 2010; 6: 3145-3156.
- [22] Meng ZZ, Qian L. Cultural relics fine modeling based on point clouds. *Bulletin Surveying and Mapping*. 2011;12: 40-43.
- [23] Magda S, Kriegman D. Fast texture synthesis on arbitrary meshes. In: Proceedings of the 14th eurographics workshop on rendering. Leuven: EGSR03; 2003. pp. 82-89.
- [24] Zwicker M, Pauly M, Knoll O, Gross M. Pointshop 3D: an interactive system for point-based surface editing. *ACM Transactions Graph*. 2002;21:322-329. doi: 10.1145/566654.566584
- [25] Zwicker M. Continuous reconstruction, editing, and rendering of point-sampled surfaces. PhD Dissertation, ETH Zürich, Switzerland; 2003. Available: <http://cds.cern.ch/record/893710>
- [26] Adams B, Wicke M, Dutre P, Gross M, Pauly M. Interactive 3D painting on point-sampled objects. *Eurographics Symposium on Point-Based Graphics*. 2004: 57-66.
- [27] Clarenz U, Rumpf M, Telea A. Finite elements on point based surfaces. In: Proceedings of EG symposium of point based graphics. Zurich: Point Based Graphics Resource; 2004. pp. 201-211.
- [28] Wang RF. Studies on the digital geometry processing of point-sampled models. PhD Dissertation, Zhejiang University. 2007, Zhejiang, China. Available: <http://cdmd.cnki.com.cn/Article/CDMD-10335-2008045883.htm>
- [29] Zhao X, Zhou KQ, Yan L. 3D reconstruction method for large scale relic landscape from laser point cloud. *Geomatics and Information Science of Wuhan University*. 2008;33: 684-687.
- [30] Troccoli A, Allen PK. A shadow based method for image to model registration. In: 2nd IEEE workshop on video and image registration. Piscataway: IEEE; 2004.
- [31] Liu L, Stamos I. Automatic 3D to 2D registration for the photorealistic rendering of urban scenes. In: Computer vision and pattern recognition, Vol. II. San Diego: CVPR; 2005. pp. 137-143.
- [32] Rosenthal P, Linsen L. Image-space point cloud rendering. In: Proceedings of computer graphics international. Istanbul: CGI; 2008. pp. 1-8.
- [33] Nagai M, Tianen Chen T, Shibasaki R, Kumagai H, Ahmed A. UAV-borne 3-d mapping system by multisensor integration. *IEEE Trans Geosci Remote Sensing*. 2009;47: 701-708. doi: 10.1109/TGRS.2008.2010314
- [34] Julien R, Peyré G, Delon J, Marc B. Wasserstein barycenter and its application to texture mixing. In: Scale space and variational methods in computer vision. Israel: Springer; 2012. pp. 435-446.
- [35] Kolb A, Barth E, Koch R, Larsen R. Time-of-flight cameras in computer graphics. *Computer Graphics Forum*. 2010; 29: 141-159.
- [36] Li B. Robust processing of 3D point clouds. Changsha: National University of Defense Technology; 2011.
- [37] Bai JJ, Zhao XS, Chen J. Indexing of discrete global grid using linear quadtree. *Geomatics and Information Science of Wuhan University*. 2005;30: 805-808.
- [38] Zheng CY, Qiu GF, Zhang ZD. Geographic information science. 2nd ed. Wuhan: Wuhan University Press; 2011.
- [39] Julesz B, Bergen JR. Human factors and behavioral science: textons, the fundamental elements in preattentive vision and perception of textures. *Bell Syst Tech J*. 1983; 62: 1619-1645. doi: 10.1002/j.1538-7305.1983.tb03502.x
- [40] Hunt RWG. Color reproduction and color vision modeling. In: Proceedings of the color imaging conference: transforms and transportability of color. Scottsdale: Society for Imaging Science and Technology; 1993. pp. 1-5.
- [41] Rusinkiewicz S, Levoy M. QSPat: a multiresolution point-rendering system for large meshes. *Computer Graphics ACM SIGGRAPH*. 2000: 343-352.
- [42] Lin FZ. Multimedia. 2nd ed. Beijing: Tsinghua University Press; 2010.

Huijun Yang, an associate professor in College of Information Engineering, Northwest A&F University in China. She received her Doctor's degree from Northwest A&F University in 2013, M.Sc. degree in Computer Science Technology at National University of Defense Technology and the B.S. degree in Computational Mathematics and Applicable Software at Shanxi University, Shanxi, China in 1997. Her research interest covers: computer graphics, Three-dimensional modelling, Virtual scene design and point cloud reconstruction. Hitherto, she has won paper Grand Prize on The presentation of 2012 National agricultural engineering doctoral academic forum and The Second Prize of excellent paper on the fourth doctoral academic forum of Northwest A&F University; She led 4 funded research projects, and participated in other 6 projects as the key contributor. She has published over 21 academic papers in the peer-reviewed journal, some of papers number were: 10.3923/jas.2013.3684.3690, 10.4304/jmm.8.4.394-401, 10.4028/www.scientific.net/AMM.229-231.1746, 10.1007/s11042-015-

1
2 2941-6, 10.1145/2087756.2087821). She has also published 1 textbook
3 (Multimedia Technology and Application) as a co-author and obtained 7
4 SOFTWARE COPYRIGHT.

5 **Ehtzaz A Chaudhry.** I am a Postdoctoral Researcher working at The Na-
6 tional Centre for Computer NCCA, the UK No.1 educational and research
7 base for computer animation. My research has focused on deformation,
8 modelling, character animation an efficient skeleton-driven and physics-
9 based skin deformation technique using curve-based surface modelling
10 derived from static and dynamic ordinary differential equations. He re-
11 ceived his Doctor's degree from Bournemouth University in 2015, M.Sc.
12 degree in University of Westminster, UK 2006 and American International
13 University, Pakistan, 2001, and the B.S. degree in The International Uni-
14 versity (TIU), Pakistan in 2000. Hitherto, he participated 2 projects and has
15 published over 10 academic papers in the peer-reviewed journal, some of
16 papers number were: 10.1109/TMAG.2002.805873,
10.1109/2945.582343, 10.1109/CADCG.2009.5246930,
10.1109/CADCG.2009.5246930, 10.1109/CGIV.2010.14).

17 **Jian Chang.** an associate professor and an active scientist in computer
18 animation with over 15 years' research experience in a world leading cen-
19 tre, the NCCA, the UK No.1 educational and research base for computer
20 animation. His research has focused on physics based modelling (defor-
21 mation & fluid), motion synthesis, virtual reality (surgery simulation), and
22 novel HCI (eye tracking, gesture control and haptic). He received his Doc-
23 tor's degree from Bournemouth University in 2007, M.Sc. degree in Solid
24 Mechanics at Chongqing University 2001 and the B.S. degree in Chong-
25 qing University, China in 1999. Hitherto, he has win best paper awards on
26 14th ACM SIGGRAPH International Conference on Virtual Reality Continuum
27 and Its Applications in Industry and Theory and Practice of Computer
28 Graphics, Proceedings (TPCG'04); He led and participated 14 projects. He
29 has published over 47 academic papers in the peer-reviewed journal (in-
30 cluding high-impact SIGGRAPH ASIA & Computer Graphic Forum) and
31 have been the programme committee member for over 20 International
32 conferences, some of papers number were: 10.1007/s00371-014-0943-4,
33 10.1002/rcs.1582, 10.17083/ijsg.v14.24, 10.1007/s00371-010-0533-z,
34 10.1111/cgf.12471). He has obtained 7 SOFTWARE COPYRIGHT.

35 **Jian J Zhang.** an Professor of Computer Graphics at the National Centre
36 for Computer Animation, Bournemouth University and leads the Computer
37 Animation Research Centre. He has worked at various universities includ-
38 ing Bristol University, University of Newcastle upon Tyne, Chongqing Uni-
39 versity, Queens University of Belfast and Brunel University. As Principle
40 Investigator, Prof Zhang has secured 7 million pound (GBP) research
41 grants to the National Centre for Computer Animation from the EPSRC,
42 AHRC, EU FP7 and EU H2020 as well as other funding organisations. His
43 research focuses on a number of topics relating to 3D Computer Animation,
44 including virtual human modelling and simulation, geometric modelling,
45 motion synthesis, deformation and physics-based animation. He is also
46 interested in virtual reality and medical visualisation and simulation. Prof
47 Zhang has published over 200 peer reviewed journal and conference publi-
48 cations. Prof Zhang has chaired over 30 international conferences and
49 symposia, and serves on a number of editorial boards. Prof Zhang has
50 given over 30 invited and keynote talks around the world. Prof Zhang is one
51 of the two co-founders of the EPSRC-funded multi-million pound Centre for
52 Digital Entertainment (CDE) with Prof Phil Willis at the University of Bath.
53
54
55
56
57
58
59
60