

High-efficiency texture coding and synthesis on point-based pear surface

Yang, Hui Jun^{1,2}, Wang, Zhi Yao¹; Wang, Xin^{1*}; Chang, Jian²; Xu, Jing¹; Zhang, Jian Feng; Chen, Can³; Zhang, Jian Jun²

¹ College of Information Engineering and Department of Foreign Languages, Northwest A&F University, Yang'ling, Shaan'xi, China.

² National Centre for Computer Animation, Bournemouth University, Poole, Dorset, BH12 5BB, UK.

³ Changzhou University, GeHu Road, WuJin District, Changzhou, Jiang'su, China.

Abstract. The fruit images on points cloud acquired by the current 3D scanner from field will appear a visible seams, inconvenient data acquisition or taking large space due to unorganized background. We give a SAOW method to cope with the space efficiency and realistic effects of texture synthesis on pear point model. At first, a point-quadtrees is proposed to simplify the pear image division. Then, an adaptive multi-granularity morton coding scheme are presented to optimizing the memory space of pear image. At last, weighted oversampling mixing method is mainly focused on texture quality of pear surface. As shown in the experiment results, our adaptive division makes the memory space decline dramatically about 90.7% than non-division and 92.9% than general division respectively; adaptive code scheme helps to reduce the memory to 72.1% of ordinary morton code; weighted oversampling keeps the mixed texture more real and smoothly than current methods.

Key Words: Texture Division, Texton, Adaptive Morton Code, Texture Synthesis

1. Introduction

1.1. Motivation

Along with the development of the digital agriculture and precision farming, the authentic appearance attributes of pear organs and fruits surface, played a more important role in their cultivars selection, increasing product sales, strengthening publicity, promoting product price and rank, broadening the market, and improving the product prestige. However due to the complexity of fruit growing environment, difficulty of data acquisition, perishable after a long time storage and technical limitations, the photorealistic 2D images, so far, have still become the mainstream technology of realistic rendering and fruit quality analysis [1-4]. As we know, the measured color and intensity for the same surface patch observed in different photos will not agree due to lighting, camera setting and surface reflectance variations from the different views. Therefore, in order to present the real characteristic, we need to cope with anamorphosis arose by color, resolution, accuracy, pose, coverage, lighting, scaling and background [5, 6]. At the same time, the 2D image technology cannot satisfy the requirement of 360-degree panoramic observation of fruits, which cannot provide the agricultural researchers realistic data[2]. In addition, the current 3D scanner is difficult in acquiring point cloud data with "real texture" directly from the fruit-growing field. In

* Corresponding author: Department of Foreign Languages, Northwest A&F University, Yang'ling, Shaan'xi, China. 303863040@qq.com

order to improve realistic appearance data, we choose both 3D scanner and camera at the same time to obtain the three-dimensional geometric point clouds and texture characteristics of fruits respectively. As known, this independence among data will result in the mismatch between attributes and shape. In this paper, by proposing a texture mapping method on pear point cloud, we intend to improve texture organization efficiency and provide real impressions for the agricultural users and researchers.

1.2. Related work

For the past few years, the texture synthesis theory and technology has been widely used in scientific exploration, reverse engineering, medical imaging, machinery manufacturing, simulation, cultural relics protection gaming and entertainment. However, these traditional methods still have difficult to reproduce the real pear texture morphology accurately and acquire a good balance between texturing quality and space in areas of Computer Graphics, 3D reconstruction and image processing, due to the following status quo [2-9].

(1) The early three-dimensional modeling methods of the plant organs still depend on the forward modeling methods or 2D extension methods. The forward plant-modeling systems, such as, fractal theory, A-system, particle system[10], L-system[11], step by step growth method[12] and interactive plant modeling [13] have paid more attention to surprise VFX. However, it needs to pay more attention to the realistic appearance, color and texture of fruits, which are quite different from each other due to the diversity of growing environment and cultivars. The 2D extension methods mainly extended two-dimensional pixel-growth method to three-dimensional[7] and its improvement with controllable orientation, scale gradient, texon mask based gradient process [14], but it has a limitation on the minimum size of texture, which cannot be less than the template size. A block-based method was also extended to three-dimensional geometric surface for texture synthesis of different scale with unable in controlling the change of direction in texture synthesis[15]. Overall, the 2D-extension based methods mainly focused on fine, smooth, and was only suitable for texture synthesis on tiled artificial sample figure, but cannot solve the correspondences between pear texture discretization and geometric primitives[2].

(2) The texture synthesis methods based on Delaunay Triangular Mesh appearing in recent years became the mainstream method in the terrain reconstruction, product modeling and other fields[4]. While for the 3D fruit surface, the existing mesh-based graphics hardware is hard to develop its high-speed advantages, because a high precision fruit model will consume large numbers of storage and rendering time, due to the following deficiencies[9]. In these methods, triangles number is twice of vertices, and texture was generally mixed by simply assigning the average value of three vertex colors to triangles. At first, a high precision pear model needs the dense mesh which will result in high cost to maintain large mesh topology relationship of point, edge and the connection information, or some small curvature areas will appear only in one color in the large sheet of some surfaces because of the less triangular meshes. Then the number of triangles is inversely proportional to the approximation error, and that will need numerous meshes on the pear surface with a high curvature, which will result in numerous pretreatment of texture information and quality sacrifices in block based texture-rendering method[8]. Next, great differences between the resolution of the mesh number and display screen will cause texture distortion. At last, the challenges of hardware design brought by disorderly evaluation is to redo the pear texture synthesis

process while adding, deleting, editing any one of points. Therefore, the traditional mesh-based reconstruction method could be unreliable for complex, uncertain pear surface with requirement of higher continuity, and the direct point-based texture synthesis technology then was vastly needed.

(3) With the development of 3D scanning technology and improvement of the scanner precision[16], the point-based modeling method has been widely used in various precision field because of advantage of non-topologic connection among points[17]. This made it possible to acquire and present the cloud points of pears with rich details and high-density more flexibly and then render more veritably in reverse engineering[18]. It is much easier to edit the sample point on the surface of the model and reconstruct the final texture directly by the point information, because the sampling points can be used to express not only model position but also the texture properties. Nevertheless, the direct points based texture synthesis seems to be difficult due to no adjacency relations between points[9], hence most of the texture synthesis on point model is only an auxiliary application of model editing[19]. Such as PointShop3D system edited 3D model [20], The point based three-dimensional surface shading system [21], pixel based texture synthesis and bump mapping result on point model [22]. Nevertheless, they still need better-distributed reflection sampling, and cannot solve the texture rendering problems of the pear surface caused by nonuniform sampling units.

1.3. Contribution and Organization

In light of problems of current methods in the pear texture synthesis, including coarse, distortion, difficulty in texture data organization, expensive in memory space, this paper proposed SAOW(simplification->adaptive code-> Overlapping sampling -> weighted mixing) method. We mainly focused on efficient organization, less storage space and authentic texture synthesis of pear texture information based on the scanned intensity information of point cloud and image information from synchronized external digital camera. The main contribution is as follows with procedure shown in Fig.1.

(1) Texture organization: based on point-octree segmentation cell-surfel [23], we convert 2D texture to three-dimensional space by developing an adaptive point quadtree for texture division and organization.

(2) Improvement of texture storage performance: improve access performance and reduce the storage size of texon by proposing multi-granularity Morton code strategy on the pear texture structure.

(3) High-quality texture mixture: a weighted oversampling scheme was designed to improving the texture synthesis quality on pear surface.

In what follows, we first discuss the texture structuration and organization in Section 2. Then, introduce the adaptive coding method of pear texture in Section 3. Next, give texture mixing scheme in Section 4. Finally, Sections 5 and 6 of this paper show the analysis of experiment, comparison and conclusion.

2. Structuration of pear texture

2.1. Collection and analysis of pear image

In view of practical and reliable of research, "QINSU" cultivars of pear from Northwest A&F

University experimental station in Yangling, Shaanxi, China, were used for this study, which texture traits are of great significance for orchard researchers to study and compare the history, development and characteristics of different pear cultivars to breed better generations.

A PCP-400 laser 3D scanner with inbuilt digital camera is chosen as our non-contact, portable device for acquiring texture image. The real size of acquired image are about 9.88cm×10.62cm for six sides, 7.76cm×8.5cm for two bottoms and 7.62cm×7.8cm for two tops respectively. Therefore, according to the fruit size, surface curvature, scanner space, and 10% overlapping between two adjacent pear slices, we design the following scheme. The side surface is taken from six angles, the bottom and the top are for twice respectively, and eventually we get 10 groups of data. The point clouds and corresponding image are presented by arrays pear (10) in Eq. (2-1) and peartex (10) in Eq. (2-2) respectively.

$$pear(10) = \{b_1, b_2, t_1, t_2, s_1, s_2, s_3, s_4, s_5, s_6\} \quad (2-1)$$

in which, b_1, b_2 --the two bottom scans, t_1, t_2 --the two top scans, s_i --the six side scans.

$$peartex(10) = \{bt_1, bt_2, tt_1, tt_2, st_1, st_2, st_3, st_4, st_5, st_6\} \quad (2-2)$$

in which, bt_1, bt_2 --the two bottom scans, tt_1, tt_2 --the two top scans, st_i --the six side scans.

The texture images are saved with *.BMP format and the texture size of pear images are: two bottoms, 220×241, two tops, 216×221, six sides, 280×301. Some part of pear texture images are shown in Fig.2. However, due to the geometric attributes and texture information being saved and organized separately, the fruit surface may appear texture dislocation with massive amount of data.

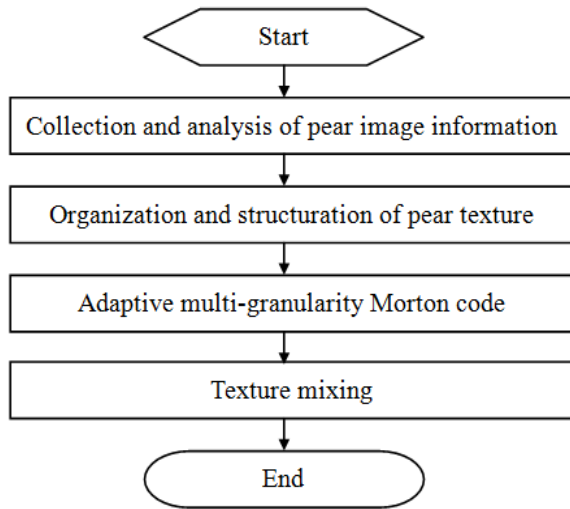


Fig. 1. Procedure of pear texture synthesis

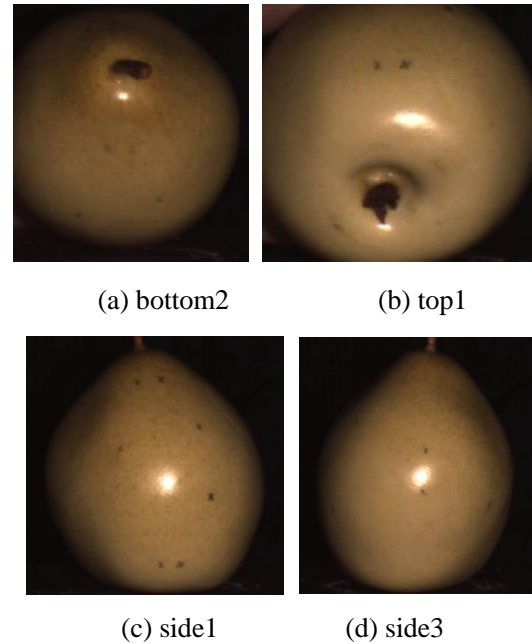


Fig.2. Part of the original images of pear

2.2. Texture organization and division

The corresponding relationship between geometric properties and the color value of a pear scan is defined with two low-dimensional spaces, in the Eq. (2-3),

$$\{P, C\} = \{p_i, c_i\} = \{(x_i, y_i, z_i), (r_i, g_i, b_i)\} \quad (i = 1, 2 \dots N) \quad (2-3)$$

in which N is the maximum image resolution.

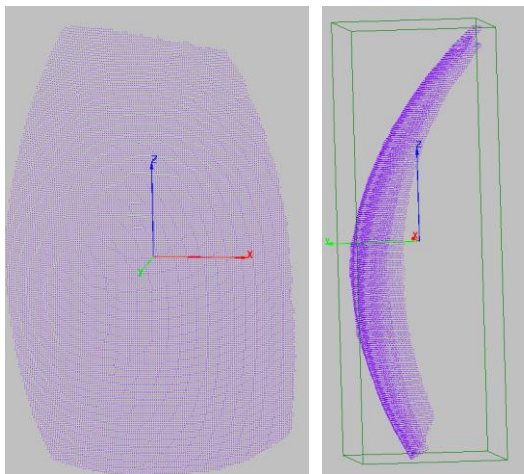
However, the color cannot be mixed directly on geometric point clouds due to the point clouds have been simplified into surfel-based structure in advance. Therefore, all the captured

pear images should first be organized into small cell-texton by dividing texture according to the corresponding division coordinates of point clouds [Classifying Images of Materials: Achieving Viewpoint and Illumination Independence].

Due to scanned from the right ahead, the outside surface of each scan is quite flat, as shown in Fig. 3(a), and the convex shell of every pear point cloud appear thin cuboid along the y-axis without self-intersection, as shown in Fig. 3(b). Which make it possible to determine each pixel by the two-dimensional coordinates (x_i, z_i) and extend point-octree partitioning algorithm to texture division by filtering y-axis. For convenient process in next steps, the scans are first transformed from x-z plane to x-y by Eq. (2-4), in which, θ is $\pi/2$, rotation angle along x-axis.

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta & 0 \\ 0 & \sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (2-4)$$

Based on the thin convex shell of point and idea of conventional quartering [24], by taking the splitting parameters p_{split} of point cloud as condition [23], color value in Eq. (2-3) as input data, we proposed a point-quadtrees division process QT_div(P,C). It performs a top-down asymmetrical recursive process with four-step division in each loop. Firstly, the parameters x and y of split position $p_{split}(x,y,z)$ are passed to the texture division coordinate $t_{split}(x,y)$. Second, four empty subsets T_j are prepared for the current division, which are used to save four part of pixel sets respectively, which is upper left, lower left, upper right and lower right. Third, search among all pixels in the current set by the iterative process, and push each pixel pix_i into one of four subsets by comparing the relationship between pixel parameters and the current split position using the function $Get_qt_code(t_{split}, pix_i)$. Forth, all non-empty subsets T_j are returned and prepared as new quadtree nodes for the next iterating division by function $QT_code(T_j)$. Finally, a function $New_qt_node(t_{split}, r_{split}, c_{0...3})$ is designed to call $QT_div(P,C)$ for generating a new quadtree node according to the location, radius and serial number. Which eventually return a series subsets of the leaf nodes recording as basket B_T . The flow chart of point-quadtrees is shown in Fig.4.



(a) Flat surface of scans (b) The thin shell Y-axis

Fig.3. Characteristics of pear ranges

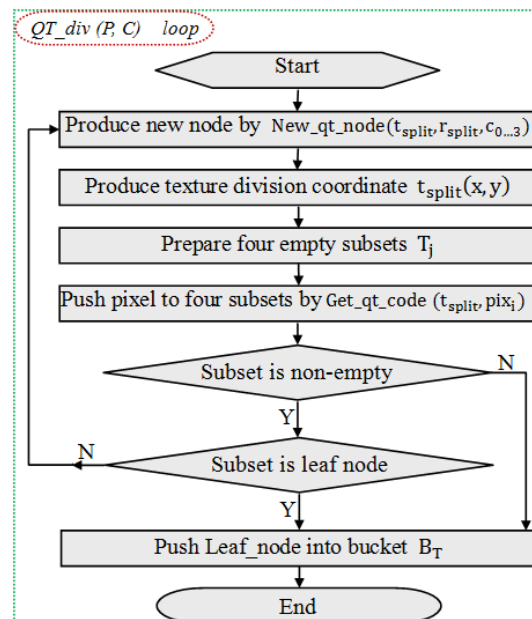


Fig.4. Flowchart of texture division QT_div()

After division, the leaf nodes is organized as a series texton sets which will cover all pear texture, each of them is a set of several pixels as well as attribute of an image block [8, 25]. The leaf nodes bucket B_T is in Eq. (2-5).

$$B_T = \{T_j\} = \{pix_{j,i}\} \quad j = 1, 2, \dots, n, \quad i = 1, 2, \dots, m_j \quad (2-5)$$

in which B_T -basket of textons; T_j -the textons among basket, also texture leaf nodes; n -the number of leaf nodes in the basket; j -texton tag which noted each texton position in basket; $pix_{j,i}$ -the i^{th} pixel of j^{th} texton; m_j -the number of pixels in the j^{th} leaf node.

3. Adaptive Morton code scheme for pear texture

Although defined in two-dimensional coordinate system, the texture needs to be mapped on a three-dimensional model, so it still should be stored and organized by octree space coding method. The current octree methods in encoding and application have mainly faced the following troubles. The regular octree node is crazy in memory consumption, because it need at least 10 pointers, which point to memory of the eight-child nodes, parent node and the node attribute data respectively. The depth-first coding and three-dimensional run length coding are better at space compression efficiency, but it have rarely been used in actual application as a result of lower query efficiency [26]. The later Linear Octree encoding saves memory space and is suitable for the situation in this paper, which solely focus on the Octree leaf nodes. Because it only stores address code and attribute values for the real leaf node, and implies the size and location information of the leaf node by taking the address code as the leaf nodes code. Nevertheless, the current Linear Octree and its improved algorithm[27] are only suitable for the full Octree by regular splitting three-dimensional mesh, also will have poor performance in query and storage efficiency while applied in our fruit point model due to its incomplete Octree with random segmentation position and multi-resolution. In order to further enhancing the performance of storage and access, we proposed multi-scale adaptive Morton coding scheme to improve the linear octree encoding efficiency, by considering the partition granularity.

The normal space size of the texture set T is $2^n \times 2^n \times 2^n$ after octree partitioning, and we give the n -digit Morton code on any leaf node as Eq. (3-1 a).

$$M_q = q_{n-1} \cdot 8^{n-1} + q_{n-2} \cdot 8^{n-2} + \dots + q_0, \quad q_i \in \{0, 1, \dots, 7\} \quad (3-1a)$$

According to the previous partition rules on the fruit[23], the digit of texton data coding is 6 at most, whose 6-digit Morton code corresponding to leaf node is given in Eq. (3-1 b)

$$M_{q_5} = q_0 q_1 q_2 \dots q_5 = q_5 \cdot 8^5 + q_4 \cdot 8^4 + \dots + q_1 \cdot 8 + q_0 \quad (3-1b)$$

in which q_i - the node code of each division ,value range is: $0, \dots, 7$.

The completely split can produce a 6-level octree with partition unit number of $2^6 \times 2^6 \times 2^6 = 262,144$, which need to be encoded with 6-digit octal code for each texton. However, according to our previous result[23], the maximum units number of the biggest scans is no more than 8000, which means only 3.05% of the units are valuable, or 96.95% of coding is redundancy. Furthermore, the segmentation process in this paper makes the most of the units stop segmentation before 6-level. Due to ease conditions by considering some factors like normal vector, the code requirement of most units are less than 6-digit.

In order to reduce the redundant code, saving storage and improving the retrieval speed, this

paper designs the coding scheme of five different granularity M_{qi} according to possible splitting level-1 to level-5, as in Eq. (3-1b) to Eq. (3-1f).

$$M_{q4} = q_0 q_1 q_2 \dots q_4 = q_4 \cdot 8^4 + q_3 \cdot 8^3 + \dots q_1 \cdot 8 + q_0 \quad (3-1c)$$

$$M_{q3} = q_0 q_1 q_2 q_3 = q_3 \cdot 8^3 + q_2 \cdot 8^2 + q_1 \cdot 8 + q_0 \quad (3-1d)$$

$$M_{q2} = q_0 q_1 q_2 = q_2 \cdot 8^2 + q_1 \cdot 8 + q_0 \quad (3-1e)$$

$$M_{q1} = q_0 q_1 = q_1 \cdot 8 + q_0 \quad (3-1f)$$

in which, M_{qi} - the i^{th} split level.

4. Weighted texture synthesis on pear surface

After the previous analysis and processing, each texton is composed of a set of discrete pixels m_j by adaptive coding, hence, visual crack may be produced among different textons. The proposed texture synthesis scheme can keep the texture fidelity by taking advantage of weighted oversampling between textons and taking the texton symbols T_j and $pix_{j,i}$ in Eq. (2-5) as description notation. On usual, the color values of local block within textons region T_j are calculated by averaging each color components respectively in Eq. (4-1),

$$r_{ji} = \sum_{l=1}^{n_i} r_{j,i,l} / n_i \quad g_{ji} = \sum_{l=1}^{n_i} g_{j,i,l} / n_i \quad b_{ji} = \sum_{l=1}^{n_i} b_{j,i,l} / n_i \quad (4-1)$$

in which, n_i - number of pixel in each local block, r_{ji} , g_{ji} , b_{ji} - red, green, blue color of $pix_{j,i}$ of texton T_j .

However, this kind of estimation method is confined to the Markov random field (MRF) model in determining the color of textons. That means, there is an obvious chicken-and-egg problem, the pixels A and B may exist in each other's neighbourhood, namely, the probability distribution of A depends on B, and vice versa. In this paper, we improve the Markov random field model to solve the interdependence between neighbor pixels, by estimating neighbor pixel values based on weighted template and overlapping oversampling.

After analysis, we discover that those pixels with closer distance to centroid have the greater contribution to the color values of textons, so the direct averaging on MRF will result in color distortion. In order to represent the authenticity of fruit color more reasonably, we first employ the weighted average based on distance, so that the weight of pixels is inversely proportional to its distance from the centroid. Then the texton is split into 3×3 pixel block (in the edges, where block may be smaller than this size, it can be extended based on the template). Third, we will sample on color component values of each pixel as shown in Fig.5 (a). And finally, color components of each block are calculated in Eq. (4-2) according to values in Fig.5(a) and weighting template in Fig.5(b).

$$e/4 + (b + d + f + h)/8 + (a + c + g + i)/16 \quad (4-2)$$

However, the above calculation methods will lead to the same motif color within one texton, and then distinct gap between two adjacent texton is easily produced which is not enough for a smooth transition. We implement smooth transition of color between adjacent textons based on the overlapping region sampling. In addition to the pixels within the neighbourhood, the pixels on the boundary of other textons adjacent to the current texton are sampled in row and column directions for color calculation of the current texton, as shown in Fig.5(c). In which the pixels in blue rectangle color are the overlapping boundary regions between current and adjacent textons.

5. The analysis and comparison

5.1. Experiment platform

We simulate pear texture synthesis on point cloud in the following experimental condition.

Hardware platform:

CPU: Intel(R) Xeon(R) CPU E5-1650 0 @3.20GHz

RAM: 4.00GB

Graphics card: NVIDIA GeForce GTX 560Ti

Software: OpenGL2.0, VS2010, matlab2010

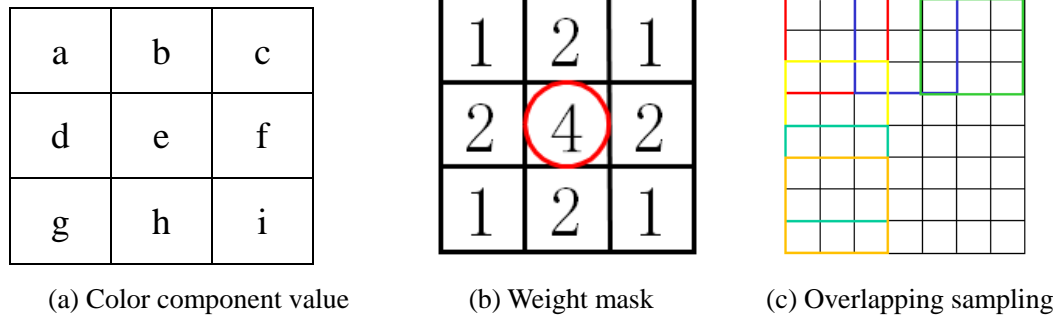


Fig.5. Weighted oversampling of texton

5.2. Adaptive division result of pear texture

We split the original texture in Fig.2. on pixel-level according to our division strategy in Section 2.2, and the results are shown in Fig.6., a series sets of textons.

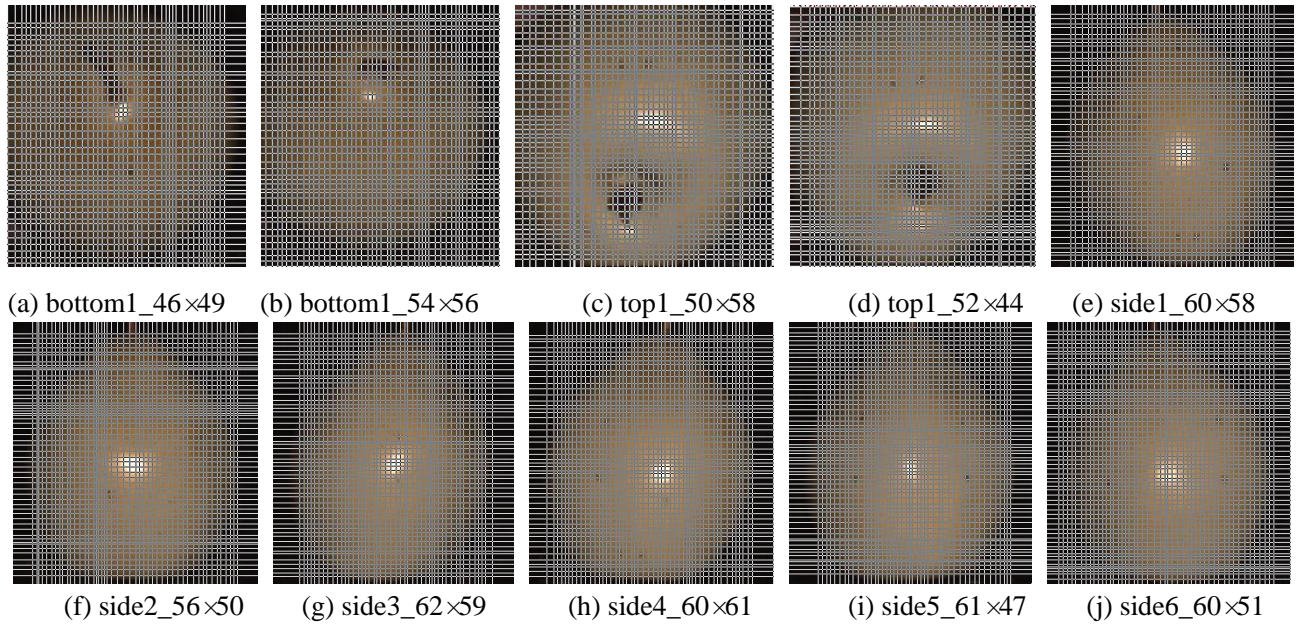


Fig.6. Pear texture division on its surface with corresponding scale of row x column

Tab.1. Part of division positions of pear texture

Texture image	Division axis	Division coordinates
B1 of pear	X-axis	0,19,23,24, 29,30,35,36,39,...,214,220,226,232,237, 241
	Y-axis	0,3,10,16,20,24,29,33,38,42,..., 196,200, 206,212,216,220
T1 of pear	X-axis	0,1,6,9,14,19,25, 30,36,39,...,198,202,207,211,215,221
	Y-axis	0,7,13,18,26,33,40,47,47,48,..., 192,198,202,207,212,216
S2 of pear	X-axis	0,15,20,26,31,36,39,47,51,..., 265,273,279,284,291,301
	Y-axis	0,23,28,30,34,39,45,49,54,...,238,244,249,253,257,280

S5 of pear	X-axis	0,8,14,18,23,28,33,37,42,..., 274,276,280,285,290,301
	Y-axis	0,35,40,45,49,54,60,66,70,...,239,244, 249,252,256,280

The final division values are shown in Tab.1, which only lists the positions of bottom1, top1, side1 and side5 of pear texture in Eq. (2-2).

The results show that the maximum texton number produced by the proposed adaptive point quadtree division on pear textures is 60×61, far less than the octree division number of $2^6 \times 2^6 \times 2^6$. That means, our method have simplified the division result and decreased the cell number.

5.3. Multi-granularity Morton code and comparison on pear texture

Multi-granularity Morton Coding: based on the convex hull of pear point set referred above, the proposed adaptive morton method in Section 3 is employed for coding on textons T_j . By randomly choosing a texton (in row 27, column 52) from Pear texture S1 as an example, it's split by five-level division and the morton code is generated according to scale in Eq. (3-1c). At last, according to the division result in Fig.6(c), the quaternary morton code of $T_{[26][51]}$ is 23101 and size is 4×6 pixels, as encoded in Eq.(5-1).

$$M_{[26][51]} = q_0q_1q_2q_3q_4 = 23101 \quad (5 - 1)$$

Memory analysis and comparison: The required memory space on 10 pear textures is given for different methods: no division no code (ndnc), general division no code (gdnc), adaptive division no code (adnc), adaptive division morton code (admc), adaptive division adaptive code (adac). The comparison result is shown in Table 4. There is only RGB color components need to save in the methods ndnc and gdnc, but in other three methods, each texton need both RGB color components and position code. Each color component will take one byte, and position code need six of quaternary.

Tab.2. Memory space of pear texture(Byte)

Texture code	b1	b2	t1	t2	s1	s2	s3	s4	s5	s6
ndnc	159060	159060	143208	143208	252840	252840	252840	252840	252840	252840
gdnc	262144	262144	262144	262144	262144	262144	262144	262144	262144	262144
adnc	18432	18432	18432	18432	18432	18432	18432	18432	18432	18432
admc	10143	13608	13050	10296	15660	12600	16461	16470	12901.5	13770
adac	7242.1	9933.8	9409.1	7320.5	11009.0	9147.6	11769.6	12089.0	9392.3	10038.3

As seen from Tab.2, the memory space of 'ndnc' is high, 'gdnc' is highest because of many empty cells are produced; 'adnc' is less dramatically about 90.7% than 'ndnc', 92.9% than 'gdnc', but all image has same space because of no coding. 'admc' is less 26.8% than 'adnc' by ordinary morton coding[27]. at last, our 'adac' makes pear texture occupy 72.1% of 'admc' at average, and more than a quarter reduction than the average morton on overall, whose memory comparison is shown in Fig.7. by adaptive coding.

5.4. Texture mixing comparison

In the following, the proposed texture mixing is tested on 10 cases of one pear from 10 angles, and the effects were compared with other texture synthesis algorithms. At first, attribute of each texton will be evaluated according to proposed weighted oversampling algorithm in Section 4, whose properties are extended to overlapping formation based on the template in Fig.5(c) and weighted template in Fig.5 (b). Then, by applying proposed method SAOW on pear surface, we

have received the texture synthesis result in Fig.8, which only give parts of the result.

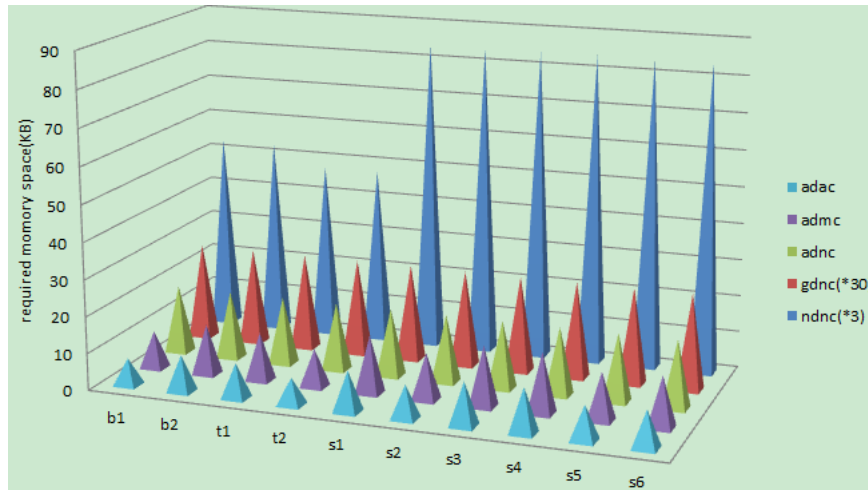
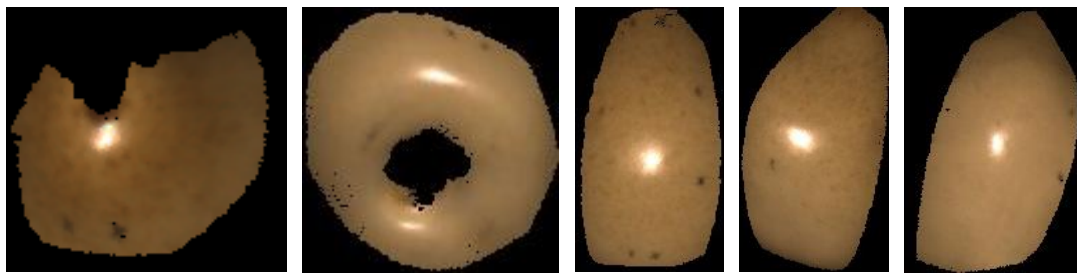


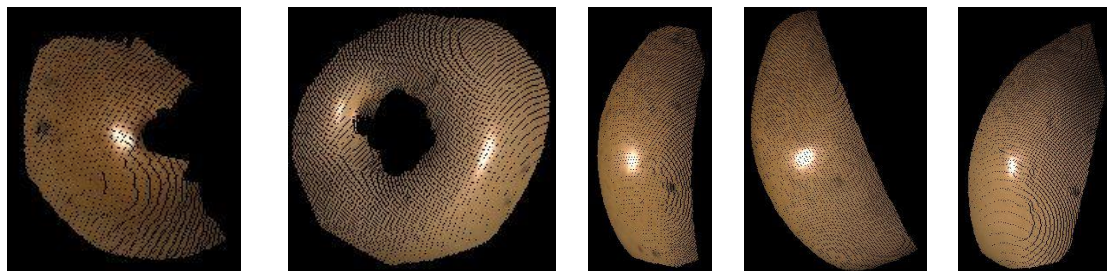
Fig.7. Contrast on storage of different coding algorithm



(a) bottom1 (b) top2 (c) side1 (d) side2 (e) side5

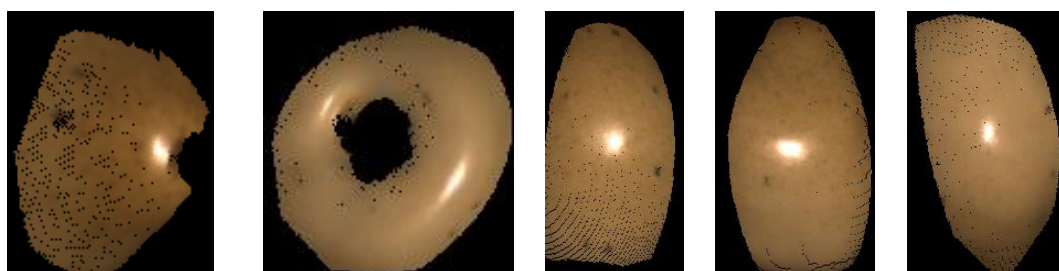
Fig.8 Texture mixing on parts of pear surface

We also apply the direct two-dimensional mapping method[28] and color adjustment technology[5] on the same pear surface for texture synthesis, the comparison results are shown in Fig.9 and Fig.10 respectively, which only give parts of the result.



(a) bottom1 (b) top2 (c) side1 (d) side2 (e) side5

Fig.9 Texture synthesis on parts of pear by direct two-dimesional mapping[28]



(a) bottom1 (b) top2 (c) side1 (d) side2 (e) side5

Fig.10 Texture synthesis on parts of pear by color adjustment technique[5]

By contrasting texture synthesis effects on pear scans between Fig.9, 10 and 8, we can find that, direct two-dimensional mapping method is simple, coarse, easily leads to "seams" and results in surface texture discontinuous. Color adjustment technique can eliminate "seams" to a certain extent with the visual effect approximating to our method, but point cloud needs to be converted to triangulation mesh in advance, and is subjected on out-of-focus of image areas.

5.5. Test on other fruit

In order to make our proposed method SAOW can be commonly used on other fruits, an additional experiment on lemon fruit is given in fig.10.

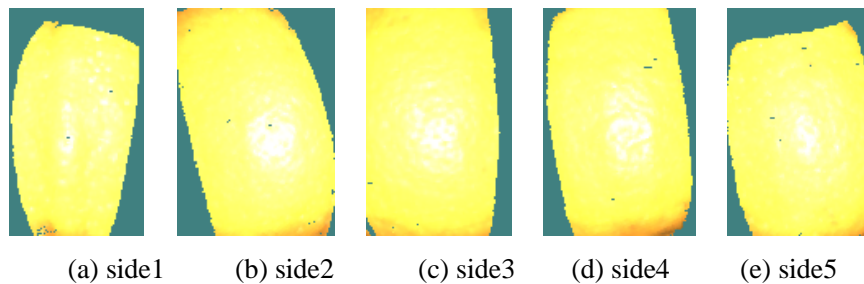


Fig.10 Texture mixing on lemon surfaces

6. Conclusion

A point set based SAOW texture synthesis algorithm proposed in this paper solved the seams on texture, low coding efficiency and large memory space in current methods. First, an adaptive division has simplified the complexity of texture-splitting and saved about 92.9% memory than general division by improving quadtree. Then, the proposed adaptive multi-granularity coding has saved 26.7% of average memory space than ordinary Morton. At last, the weighted oversampling mixing method, can keep the texture authenticity comparing with previous techniques. Experimental results also show that, proposed SAOW algorithm has optimized space performance, quality storage and balance between VFX and performance of texture mixing.

It is noteworthy, that the high compression rate is at the expense of a losing image quality, so it should be deliberating while dealing with the larger size textons.

Acknowledgment

This work was supported in part by a grant from the Fundamental Research Funds for the Central Universities (No. 2014YB067, 2452015199, 2452015195), grant from the National High Technology Research and Development Program of China (863 Program, No. 2013AA10230402), National Natural Science Foundation of China(61402374), Scholarship Council and Scientific Research Foundation for PH.D from Northwest Agriculture & Forest University of China (2014BSJJ060).

References

1. Cetinkaya, N., et al., *Quantification of applied dose in irradiated citrus fruits by DNA Comet Assay*

- together with image analysis. *Food chemistry*, 2016. **192**: p. 370-373.
2. Farinella, G.M., et al., *Retrieval and classification of food images*. *Computers in Biology and Medicine*, 2016. **77**: p. 23-39.
 3. Zhu, Z., et al., *Dynamic texture modeling and synthesis using multi-kernel Gaussian process dynamic model*. *Signal Processing*, 2016. **124**: p. 63-71.
 4. ZHANG, D., et al., *FACIAL BIOMETRICS USING NONTENSOR PRODUCT WAVELET AND 2D DISCRIMINANT TECHNIQUES*. *International Journal of Pattern Recognition & Artificial Intelligence*, 2009. **23**(3): p. 521-543.
 5. Pan, R. and G. Taubin, *Color adjustment in image-based texture maps*. *Graphical Models*, 2015. **79**: p. 39-48.
 6. Gatys, L.A., A.S. Ecker, and M. Bethge, *Texture Synthesis Using Convolutional Neural Networks*. *Computer Science*, 2015. **70**(1): p. 51-5.
 7. Turk, G. *Texture synthesis on surfaces*. in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. 2001. ACM.
 8. Magda, S. and D. Kriegman. *Fast texture synthesis on arbitrary meshes*. in *ACM SIGGRAPH 2003 Sketches & Applications*. 2003. ACM.
 9. Mariethoz, G. and S. Lefebvre, *Bridges between multiple-point geostatistics and texture synthesis: Review and guidelines for future research*. *Computers & Geosciences*, 2014. **66**(C): p. 66-80.
 10. Lino, A.C.L., I.M. Dal Fabbro, and C. de Almeida, *Fruit Surface Topographic Survey Supported By a Phase Shifting Projection Moiré Technique*. *AIP Conference Proceedings*, 2008. **992**(1): p. 1028-1033.
 11. Rodkaew, Y., et al., *Animating plant growth in L - system by parametric functional symbols*. *International journal of intelligent systems*, 2004. **19**(1 - 2): p. 9-23.
 12. De Ret'fye, P. and J. Houll, *Tomato Growth Modeling Based on Interaction of its Structure-Function*.
 13. Zeng, L., et al., *Interactive Plant Modeling Based on Freehand Sketches*, in *Transactions on Edutainment XI*. 2015, Springer. p. 114-126.
 14. Zhang, J., et al., *Synthesis of progressively-variant textures and application to arbitrary surfaces*. 2007, Google Patents.
 15. Soler, C., M.-P. Cani, and A. Angelidis, *Hierarchical pattern mapping*. *ACM Transactions on Graphics (TOG)*, 2002. **21**(3): p. 673-680.
 16. Koyuncu, B. and K. Kullu. *Development of an optical 3D scanner based on structured light*. in *Proceedings of the 9th WSEAS international conference*. 2010.
 17. Nan, L., et al., *SmartBoxes for interactive urban reconstruction*. *ACM Transactions on Graphics (TOG)*, 2010. **29**(4): p. 93.
 18. Rosell, J.R., et al., *Obtaining the three-dimensional structure of tree orchards from remote 2D terrestrial LIDAR scanning*. *Agricultural and Forest Meteorology*, 2009. **149**(9): p. 1505-1515.
 19. Zwicker, M., *Continuous reconstruction, editing, and rendering of pointsampled surfaces*. 2003, PhD thesis, ETH Zürich, Zürich, Switzerland.
 20. Wang, J.-T., et al., *A PCA AND PERTURB BASED FRAGILE WATERMARKING SCHEME FOR 3D MODELS*. 2010.
 21. Adams, B., et al. *Interactive 3D painting on point-sampled objects*. in *Eurographics Symposium on Point-Based Graphics*. 2004. Citeseer.
 22. Clarenz, U., M. Rumpf, and A. Telea. *Finite elements on point based surfaces*. in *Proc. Eurographics Symp. Point-Based Graphics*. 2004.
 23. Yang, H.-j., et al., *Research on segmentation of pear shape from unorganized point clouds*. *Journal of Multimedia*, 2013. **8**(4): p. 394-401.

24. Raper, J., *Geographic information science*. Annual review of information science and technology, 2009. **43**(1): p. 1-117.
25. Kennel, P., C. Fiorio, and F. Borne, *Supervised image segmentation using Q-Shift Dual-Tree Complex Wavelet Transform coefficients with a texton approach*. Pattern Analysis and Applications, 2015: p. 1-11.
26. Kraak, M.-J. and F. Ormeling, *Cartography: visualization of spatial data*. 2011: Guilford Press.
27. Wei, J., J. Tang, and G. Wu, *Multi-resolution organization and management of massive 3D seismic data volume*. Progress in Exploration Geophysics, 2013. **49**(3): p. 240-244.
28. Zhao, X., et al., *3D Reconstruction Method for Large Scale Relic Landscape from Laser Point Cloud*. Geomatics and Information Science of Wuhan University, 2008. **33**(7): p. 684-687.