An Interactive Construction Method of 3D Objects from Chinese Ink Paintings

Lijie Yang[1(✉)], Tianchen Xu[2], and Xiaoshan Li[2]

[1] College of Computer Science and Technology, Huaqiao University, Xiamen, China
yanglijie81@gmail.com

[2] Faculty of Science and Technology, University of Macau, Macau, China
TianchenX@acm.org, xsl@umac.mo

Abstract. We propose an interactive construction method of 3D objects from Chinese ink paintings for the challenge problem of generating the Chinese ink animation. *Marching Cube* method is the popular method of 3D modeling; however, it has the limitation on constructing the objects with complex shapes by lines in Chinese ink paintings. Based on our method, we develop a software system for constructing 3D objects interactively by the manual input of brush strokes from the Chinese ink paintings. And then the system can generate 3D objects with 2D surface mesh and 3D skinned mesh automatically. Finally, the system renders them with the original ink effect textures of input image. Our experiments show that the approach is suitable for converting 2D painting image into 3D objects. This work would be helpful for the problem of generating 3D animation of Chinese ink painting.

Keywords: Animation · Geometry processing · 3D modeling · Non-photorealistic rendering (NPR) · Chinese ink painting

1 Introduction

Chinese ink painting, where only black ink is generally used, is famous for its freehand brushwork and natural aesthetic values. Its characteristic interests such as freedom and lightness can be addressed without stop by moving from the ancient freehand brushwork of Chinese landscape painting to 2D ink-and-wash cartoons, such as "Where is Mummy", even to recent 3D ink-and-wash animations, e.g. "Ode to summer" [6]. However, generating cartoons in style of Chinese ink painting are tedious and time consuming due to variability of lines and the spontaneous dispersion of ink on *Xuan* paper. It is apparent that these works are significantly labor-intensive, although current 3D animation software can improve a little bit. For enhancing the efficiency and reducing manpower in producing of 3D ink-and-wash cartoon, in this paper we propose a method for constructing 3D objects from Chinese ink paintings for the challenge problem of generating Chinese ink animation. We develop a software system based on our method, and experimental results are acceptable.

Z. Pan et al. (Eds.): Transactions on Edutainment XI, LNCS 8971, pp. 179–186, 2015.
DOI: 10.1007/978-3-662-48247-6_16

1.1 Related Work

As far as we know, there has been little or no work on our topic. However, several related works have been explored into the following aspects:

- **3D Model Rendering.** As the rise of research on Non-Photorealistic rendering (NPR), various freehand paintings were gradually brought into the digital painting world. In the field of Chinese ink painting rendering, Way et al. [7] raised some methods using silhouette and texture strokes to draw trees and mountains in Chinese landscape painting. Li [4] analyzed and simulated the techniques of traditional Chinese landscape paintings using Maya. Like our motivation, Yuan et al. [10] implemented a real-time rendering system for generating a Chinese ink-and-wash cartoon to free the animators from laboriously designing traditional Chinese painting appearance. Xu et al. [11] extracted classified strokes from 3D model and attached typical strokes' textures onto its surface. However, these work mainly concentrated on the rendering techniques.
- **Brush Simulation.** Simulation of the interaction of water, ink, paper and brushes has been regarded as another popular area of NPR. Much work used position and pressure of a brush to determine location and width of a stroke [1, 3]. In image-based simulation, novel methods were devised for reproducing the drawing and writing processes of the static Chinese calligraphy and painting images [8, 9].

In a word, the rendered or animated 3D models in the recent research are mostly derived from the real objects like tree, frog, etc. rather than the objects with the shape characteristics in Chinese painting. In contrast, we concern more about the problem of rendering with the input 2D image and adopting post-processing for emulating some special ink effects like "dry brush" and "diffused edge". The novelty of our paper lies in the fact that we construct the 3D geometry model of the brush stroke in Chinese ink painting, which transfers a painting image to 3D scene with the features of "line modeling" (i.e. translating life into line). Given an image of painting, after the complex image is decomposed into brush strokes (Sect. 2.1), our 3D geometry model construction (Sect. 2), which is the core of our framework, prepares the required data of the 3D model for skeletal animation technique, then we preserve the original ink effects over the surfaces of 3D models by mapping the textures which are sampled from the input painting images. Here we choose bird-and-flower painting in great freehand style where painting objects mainly cover plants, fish, insects, birds, pets etc. as the representative objects since these paintings are quite popular and classical. Given the left three images in Fig. 5(d), masterpieces by the famous painter Qi-Baishi, we generate their 3D models and make an animation to demonstrate our approach (Sect. 3).

2 Geometry Model Construction of 3D Stroke

Unlike western paintings, Chinese ink painting generally uses lines (i.e. brush strokes which produce a painting hypothetically) to represent an entity in real world. So we construct our 3D geometry model at the level of brush strokes for keeping this natural feature of "line modeling". Since skeletal animation is quickly becoming the popular animation technique with its high efficiency and dramatic results, we construct all the data structures (i.e. mesh and bones, skeleton hierarchy, and weight allocation) to comply with its common format. Also our 3D models can be imported into most 3D modeling software to implement animation applications and products.

2.1 Brush Stroke Decomposition

In general, a painting object with a complex shape is painted by several brush strokes hypothetically, some of which may be overlapped and noised. Before mesh construction of 3D stroke, we develop an interactive tool for stroke extraction. Given a painting image, first we decompose it into regions of similar color intensities and then merge contiguous regions that likely belong to the same brush strokes. Second after smoothing the shape boundaries of strokes, we extract the contours of strokes by edge detection. In particular, as shown in Fig. 1(a), the fragmented shape, such as "Fin1", "Fin2", "Fin3", "Fin4" and "Caudal fin", which are produced by some distinct brushworks like "dry brush", as well as the overlapped strokes like strokes "Body" and "Dorsal fin" can't be automatically restored by the smoothing algorithm. So the refinement of these strokes' contours has to rely on user interaction.

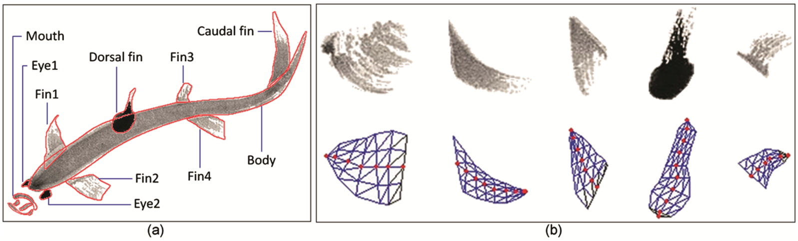

Fig. 1. (a) Brush strokes enclosed by the red contours in a fish painting image. (b) Triangulation examples of five strokes (Color figure online).

2.2 Surface Mesh of 2D Stroke

Unlike *Marching Cube* (Sect. 3), we create efficient topology with clear edge flow to mesh 2D surface of a stroke faithfully following its shape. Figure 2 illustrates our process of 2D surface mesh construction given a pseudo-stroke.

Contour-Based 2D Mesh. The establishment of our mesh structure depends on a stroke contour which represents the stroke's shape.

1. **Medial Axis Computation.** The curves in our 2D mesh should describe the geometric information of a stroke. We devise a simple method to find out the medial axis of a shape with its contour instead of skeletons, since the existing thinning algorithms most probably produce branches to noise the shape description. Given a contour of a single stroke shape, we proceed:

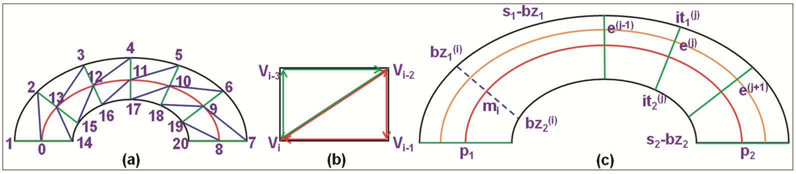

Fig. 2. Given a surface of pseudo-stroke, (a) our triangulation method provides a index table {(0,1,2), (0,2,13), (13,2,3), (13,3,12), (12,3,4), (12,4,11)…} with the vertices indices (0,…,20). (b) and the four vertices of a single polygon in a counterclockwise order. (c) Effect of each item in our 2D surface meshes construction.

(a) *Contour partition.* We segment the contour into two sides as s_1 and s_2, which both present a stroke shape; we first sample the contour of the shape into n points at one-pixel interval and then classify the points into two sides by two special points on the contour p_1 and p_2, which decide the start and end of the stroke when a painter drops and lifts a brush tip to paint the stroke. To determine p_1 and p_2, we adopt an extraction method of feature points [2] to find out corner points as candidates of p_1 and p_2. In the case of more than two corner points, we iteratively merge two of them into an approximate point on the contour according to their midpoints and curvatures, until p_1 and p_2 are identified.

(b) *Contour equal division.* We use piecewise cubic Bézier curve to fit the points inside s_1 and s_2 because a single piece curve hardly approximates to the contour of the stroke shape which is commonly complicated and varied in Chinese ink painting. Additionally, we apply *Least Square Method (LSM)* to minimize the squared distance between the original data and the fitted data. Denoting two Bézier approximations of s_1 and s_2 by bz_1 and bz_2, we sample the contour into $n + 1$ points by dividing bz_1 and bz_2 into n equidistant slices. Let bz_1^i and bz_2^i be the i-th point on the bz_1 and bz_2 respectively.

(c) *Midpoints detection.* We calculate the midpoints m_i between bz_1^i and bz_2^i to fit the final medial axis by piecewise cubic Bézier curve, which is also regarded as the spine axis of our mesh.

2. **Edge Flow Construction.** We construct edge flow from four sided polygons (quads) in the mesh which is good for animation production: defining longitude lines along the medial axis of the stroke and latitude lines perpendicular to it in our edge flow, we sample the spine axis into m points and use the notation sp_j to refer to the j-th point on the spine axis; second we construct normal line, as a latitude line, from the spine axis crossing at sp_j, and the intersection points (it_1^j, it_2^j) between normal line and two contour sides bz_1 and bz_2 appear as vertices of the mesh. Along each normal line, we also sample two slices, between sp_j and two intersections, into k equidistant points as vertices of the mesh. Denoting the e-th point on the j-th slice by e^j, we define the lines from e^j to e^{j-1} and e^{j+1} as the edges, i.e. longitude lines.

Triangulation. Based on longitude and latitude lines and vertices in the mesh, we obtain a large number of faces with the multiply-connected polygons. For recording the mesh, we provide a succinct triangulation encoding method, where the vertices can be indexed by triangulating the faces sequence (with repetitions). As one polygon shown in Fig. 2(b) with four vertices, (V_{i-3}, V_{i-2}, V_{i-1}, V_i), a sequential triangulation table encodes the set {(V_i, V_{i-3}, V_{i-2}), (V_i, V_{i-2}, V_{i-1})}. In Fig. 1(b), given five strokes of "fins" in first row, we triangulate all the polygonal faces and encode the vertices in a counterclockwise order with triangulation tables.

2.3 Skinned Mesh of 3D Stroke

To meet the requirements of 3D mesh in skeleton animation, we operate the 2D surface mesh as follows;

1. **2D Surface Model Extrusion.** As shifting a 2D surface mesh into *x*-*z* plane of 3D coordinate system, our system first clones the mesh and mirror the replica about the *x*-*z* plane automatically. Then we extrude the original one to the upper half of the 3D mesh by stretching *y*-coordinates of the vertices along spine axis and latitude lines, and the replica is extruded in the same way for the lower half. The stretched vertices on the cross-section would be along a curve (Bézier by default) which is optional and custom for users since 3D shapes of strokes are various. Finally we weld the overlapped vertices with hash map (re-mapping the index buffer) in order to save the vertex buffer and make the mesh components convenient to manage. Given 2D mesh of "body", Fig. 3(a) shows its stretched vertices in different planes. Actually, it's not necessary to extrude all 2D surface models of the strokes since some of their shapes approximate to 2D planes in real world.
2. **Bone Extraction.** Bone is the kernel of a 3D model. Given the number of bones N_B, our bone extraction provide new points along the original spine axis as the joints of the bones J_k ($k = (0, \ldots, N_B)$) with the data recorded in spine axis of a 2D mesh. Thus the new joint is calculated by a linear interpolation between two vertices along the spine axis, where the weight of the interpolation is determined by
$$\mathrm{w_k} = \mathrm{k}\frac{\mathrm{N_S}}{\mathrm{N_B}} - \frac{\mathrm{L_A}}{\mathrm{L_S}}, \tag{1}$$
3. **Skeleton Hierarchy Construction.** In skeletal animation, a skeleton hierarchy is used to link one bone to the other, including only one "root bone" as well as its siblings or children. So we devise a recursive searching algorithm to implement this hierarchy for the bones in a 3D model. Taking our fish model as an example (Fig. 3(b)), for the bones in main mesh (the mesh of "body" with a "root bone"), our algorithm searches each bone for its closest bone on both sides of "root bone" and marks them as the child nodes; for the bones in other meshes, after defining one of the bones in main mesh as a local root, the algorithm recursively searches the rest nodes of the hierarchy from this local root. In this way we connect all the bones in the different meshes of the model.
4. **Weight Allocation.** In the traditional work flow of animation production, after modeling, the vertices need to be bound to the bones and allocated the weights manually. For the objects in bird-and-flower painting, which mainly cover plants and animals, we devise the weight allocation principle, as the blending area is relatively large (Fig. 3(c)):

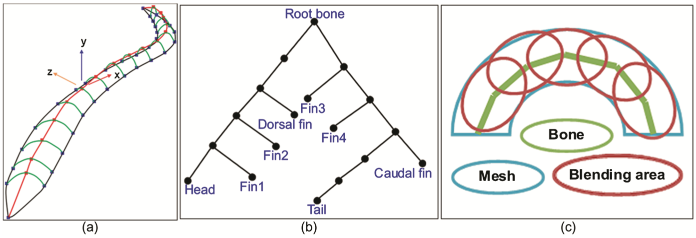

Fig. 3. In our fish model, (a) stretching operation (red and green curves) of its 2D surface mesh. (b) its bone hierarchy. (c) our weight allocation principle.

$$(\mathrm{B}(w) - p) \cdot (B(w)') = 0, \tag{2}$$

where $B(w) = w^2 b_0 + 2(1 - w)wjt + (1 - w)^2 b_1$, and $B(w)'$ is the derivative of $B(w)$.

5. **Pose Standardization.** In order to make the geometry deformation process easy, before pose standardization, we need to unify the bind-pose which is initially originated from the painting object. For bird-and-flower paintings, the bind-pose is defined by "A-shaped" pose which is popular in human beings models. For example, we deal with the bind-pose in our fish model with the body along the x-axis from the head to the tail and the fins rotated 45° from the body spine. Then we standardize the pose by recovering the mesh and the skeleton into such bind-pose with the root joint at the centre of the model coordinate O_{md}: each bone in the hierarchy is translated to make its start joint at O_{md} with $(-L_p,0,0)$, where L_p denotes the length of its parent bone and its sign(+) represents the positive direction of the bone; second, we align the bones to the bind-pose axis (e.g. in fish model, along x-axis for body and the line with 45° to the spine for fins) with a rotation and translate the bones back with $(L_p,0,0)$. The 3D meshes in the middle of Fig. 5(a)–(c) demonstrate our construction method.

2.4 Animation Set

In order to produce an animation, we import our data structures of 3D mesh, which comply with the general format of skeletal animation, into the modeling software Autodesk® 3ds Max® to design key frames of the 3D model in time sequences along the specified motion paths. Finally, we calculate the continuous frames by a linear interpolation technique.

3 Experimental Results

Based on above proposed method, we develop a software system for constructing 3D models from Chinese ink paintings interactively.

1. **Results.** We design two application programs for 3D modeling which are used for a 2D surface mesh and the 3D skinned mesh (Fig. 4). As shown in Fig. 5(a)–(c),

given the 2D painting images in the first column, our system generates the corresponding 3D models (the middle column) and renders them with original ink effects (the last column). The three images on the right of Fig. 5(d) display a series of snapshots of our 3D animation.

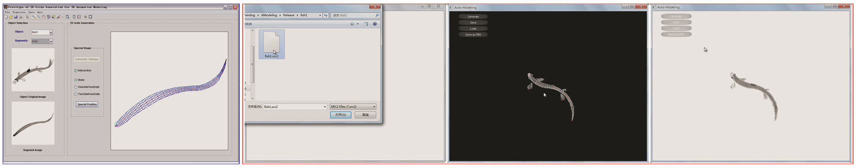

Fig. 4. Given a fish painting, user interfaces for constructing the 2D surface mesh (in a blue rectangle) and the 3D skinned mesh (in a red rectangle) (Color figure online).

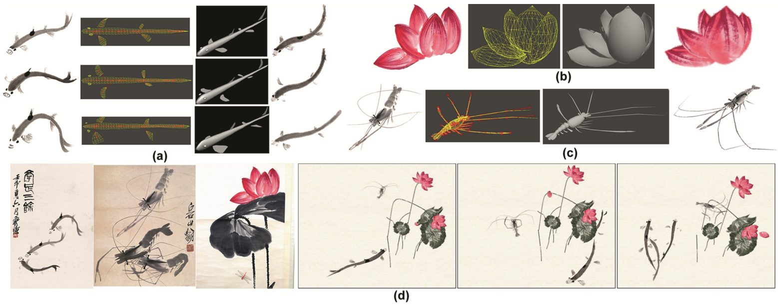

Fig. 5. (a)–(c) 3D models of lotus, fish and shrimp paintings by our system. (d) Given three images of Chinese ink painting (left), snapshots of our 3D animation (right).

2. **Evaluation.** *Marching Cube* [5] is a well known algorithm in 3D modeling, which extracts a polygonal mesh of an isosurface from a 3D scalar field (also called *voxels*). It has some obvious advantages such as simple rendering and manipulation, high resolution and efficient computation. However, there are several problems as follows, which are solved by our regular surface meshes: (a) the algorithm may generate the ambiguous faces, which bring about holes and wrong surfaces in a model; (b) it is incapable of modeling brush strokes in Chinese ink painting, since bones of 3D models are by no means extracted from the ruleless mesh; and (c) the mesh of skeletal animation is rendered more efficiently than *Marching Cube*.

4 Conclusion

This paper presents a novel sketch-based method to constructing 3D models from 2D images of Chinese ink paintings interactively. The method includes geometry processing of 2D surface mesh and 3D skinned mesh. The main objective is to generate 3D geometry models for generating Chinese ink animation. In addition, the obtained data structures of a 3D model with minimum user interactions can be used by other mainstream software of modeling and animation. Therefore, it can be useful for animation production. Although this is just our first small step on this topic, the experimental results are satisfied. In the future, we will improve the automatization of some processing steps. And we should mesh the different parts of 3D object with the different densities, and enhance the ink effect of stroke outline as well.

References

1. Chu, N.S.H., Tai, C.L.: Moxi: real-time ink dispersion in absorbent paper. ACM Trans. Graph. **24**(3), 504–511 (2005)
2. He, S., Shaojun, Z., Jingliang, X., Chunyan, Y.: Extracting profile feature points by a method based on arc length. Opto-Electronic Eng. **31**(11), 59–62 (2004)
3. Lee, J.: Simulating oriental black-ink painting. Comput. Graph. Appl. IEEE **19**(3), 74–81 (1999)
4. Li, X.: Rendering technology of 3D digital Chinese ink-wash landscape paintings based on maya. In: Pan, Z., Cheok, A.D., Müller, W., Liarokapis, F. (eds.) Transactions on Edutainment IX. LNCS, vol. 7544, pp. 150–159. Springer, Heidelberg (2013)
5. Lorensen, W.E., Cline, H.E.: Marching cubes: a high resolution 3D surface construction algorithm. SIGGRAPH Comput. Graph. **21**, 163–169 (1987)
6. Neoh, T.: Ode to summer. In: ACM SIGGRAPH 2003 Video Review on Animation Theater Program: part 2 - vol. 146. p. 10 (2003)
7. Way, D.L., Lin, Y.R., Shih, Z.C.: The synthesis of trees Chinese landscape painting using silhouette and texture strokes. J. WSCG 499–506 (2002)
8. Yang, L.J., Xu, T.C.: Animating Chinese ink painting through generating reproducible brush strokes. Sci. China Inf. Sci. **56**(1), 1–13 (2013)
9. Yang, L.J., Xu, T.C., Li, X.S., Wu, E.H.: Feature-oriented writing process reproduction of Chinese calligraphic artwork. In: ACM SIGGRAPH Asia 2014 Technical Briefs, no. 5, pp. 1–4 (2014)
10. Yuan, M., Yang, X., Xiao, S., Ren, Z.: Gpu-based rendering and animation for Chinese painting cartoon. In: Proceedings of Graphics Interface 2007, pp. 57–61 (2007)
11. Xu, T.C., Yang, L.J., Wu, E.H.: Stroke-based realtime ink wash painting style rendering for geometric models. In: ACM SIGGRAPH Asia 2012 Technical Briefs, no. 19, pp. 1–4 (2012)