Automatic 3D-Object Modeling From Multiple Uncalibrated Images Using Active Contour

Tze-kin LAO Kin-hong WONG Kam-sum LEE Siu-hang OR

Department of Computer Science and Engineering
The Chinese University of Hong Kong
Hong Kong, China

Abstract ***A new algorithm of generating 3D model from multiple uncalibrated images is presented. Images are taken from an objects rotating in front of a stationary camera. The object is segmented from these multiple uncalibrated images using active contouring. The boundary points of the object are obtained from the active contour, a wireframe containing triangles of the 3D model can be generated. The texture of the 3D model is mapped from a panoramic image generated from some of the uncalibrated images. The presented approach aims at the reduction of human interaction on locating the point correspondences and provided a system integration of existing methods.***

Keywords: 3D-Object Modeling, Panoramic Image, Active Contour

1 Introduction

In recent years there has been increased interest, within computer graphic community, in image based rendering system. These image-based systems is composed of a set of photometric observations. In photogrammetry the initial problem of camera calibration, two-dimensional image registration, and photometrics have progressed towards the determination of three-dimensional models.

Many researchers are working on point correspondence technique to perform image registration. However, it usually requires a number of correspondents point in order to have accurate modeling result. Users are required to figure out the points before modeling. This is a time-consuming process and also the accuracy of corresponding point is not guaranteed. View morphing proposed by Manning and Dyer [1] is one of the point correspondence technique of object modeling. Seitz and Dyer [4] also proposed a view morphing method which involved human interaction. An automatic method, Active Contouring, is proposed to find out the corresponding points.

A wireframe model can be generated by the points obtained. The shape of the object is estimated by this wireframe. The rendering of this 3D model does not satisfy the requirements of computer animation in terms of realism. Additional information about surface texture has to be added to the 3D model to improve its appearance. Texture mapping is a technique to enhance the realism immersed. A texture image is mapped onto the surface of a 3D model. Niem and Broszio[3] assigns texture to a surface polygon from that camera image, which ensures the highest texture resolution.

The objective of this work is to reduce the human interaction on locating point correspondences of an object during the process of generating 3D model. The model generated is in VRML V2.0 format. Those models are used in the field of computer animation, object display in web historical center, film production, etc.. These applications require an simple and easy algorithm without knowing the theory of camera calibration and point correspondence.

2 Active Contour Model

Active contour or snake are curves defined within an image domain that can move when there exist internal forces within the curve itself and external force derived from the image data. The influence of these two forces will conform the active contour to an object boundary. Snake are widely used in many applications, including edge detection, shape modeling, segmentation and motion tracking. In this paper, active contour technique is used for finding the boundary points of an image.

2.1 Parametric Snake Model

According to Xu and Prince [6], a traditional active contour is a curve $\mathbf{x}(s) = [\mathbf{x}(s), \mathbf{y}(s)], s \in [0,1]$, that moves through the spatial domain of an image to minimize the energy functional

$$E = \int_0^1 \frac{1}{2}(\alpha|\mathbf{x}'(s)|^2 + \beta|\mathbf{x}''(s)|^2) + E_{ext}(\mathbf{x}(s))ds \quad (1)$$

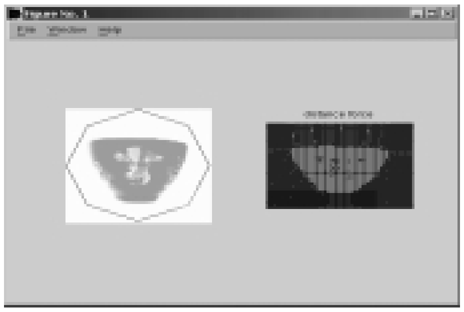
where α and β are weighting parameter that control the snake's tension and rigidity respectively. $\mathbf{x}'(s)$ and $\mathbf{x}''(s)$ denote the first and second derivatives of $\mathbf{x}(s)$ with respect to s. The external energy function E_{ext} takes on its smaller values at the boundaries. It can be derived from the image given.

Given a gray-level image $\mathbf{I}(x,y)$, typical external energies an active contour is led by two typical external energies toward step edges.

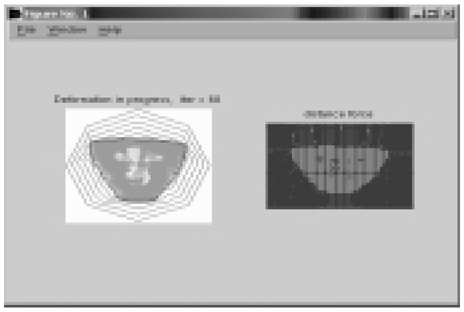
$$E^1_{ext}(x,y) = -|\nabla \mathbf{I}(x,y)|^2 \quad (2)$$

$$E^2_{ext}(x,y) = -|\nabla(\mathbf{G}_\sigma(x,y) * \mathbf{I}(x,y))|^2 \quad (3)$$

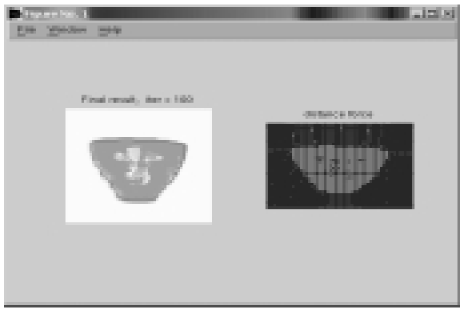
where $\mathbf{G}_\sigma(x,y)$ is a two-dimensional Gaussian function with standard deviation σ and ∇ is the gradient operator. A larger σ's will cause the boundaries to become blurry and distorted from the above observation. In order to make the effect of the boundary fall at some distance from the boundary, i.e., to increase the "capture range" of the active contour, such large σ is necessary.



(a)



(b)



(c)

Figure 1 : Snake model (a) Initialization (b) Deforming Process (c) Final Result

A snake that minimize E must satisfy the

Euler equation

$$\alpha \mathbf{x}''(s) - \beta \mathbf{x}''''(s) - \nabla E_{ext} = 0 \qquad (4)$$

This can be viewed as a force balance equation

$$\mathbf{F}_{int} + \mathbf{F}^1_{ext} = 0 \qquad (5)$$

where $\mathbf{F}_{int} = \alpha \mathbf{x}''(s) - \mathbf{x}''''(s)$ and $\mathbf{F}^1_{ext} = -\nabla E_{ext}$. The internal force $\mathbf{F}_{int}$ tries to expand the active contour outward while the external potential force $\mathbf{F}^1_{ext}$ pulls the active contour towards the desire image contour.

To find a solution to (4), the active contour is made dynamic by treating $\mathbf{x}$ as function of time t as well as s, i.e. $\mathbf{x}(s,t)$. Then, the partial derivative of $\mathbf{x}$ with respect to t is then set equal to the left hand side of (4) as follows

$$\mathbf{x}_t(s,t) = \alpha \mathbf{x}''(s) - \beta \mathbf{x}''''(s) - \nabla E_{ext} \qquad (6)$$

When the solution $\mathbf{x}(s,t)$ stabilizes, the term $\mathbf{x}_t(s,t)$ vanish and we achieve a solution of (4). This solution can be solved iteratively.

2.2 Edge Map

We can use the following equation

$$f(x,y) = -E^i_{ext}(x,y) \qquad (7)$$

because an edge map $f(x,y)$ can be derived from the image $I(x,y)$, where i=1,2,3, or 4. In general, the field ∇f has vectors pointing toward the edges resulting a narrow capture range. Moreover, in homogeneous regions, $I(x,y)$ is constant, ∇f is zero, and therefore no information about nearby or distant edges is available.

3 Basics of Texture Mapping

3.1 Texture Mapping From Parameterization

According to Niem and Broszio[3], texture mapping is a process of mapping a texture image onto a surface of a 3D model which is represented by a set of polygons, i.e. wireframe model. The model is consecutively projected to a 2D screen. There are three different types of coordinate systems used in the texture mapping process.

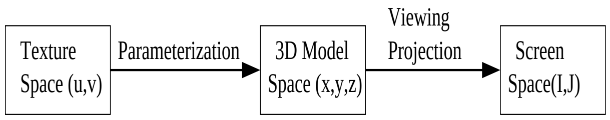


Figure 2 : Mapping from texture space to screen space using parameterization and viewing projection

- *texture* space is labeled with (u,v) coordinates
- *3D model* space is labeled with (x,y,z) coordinates
- *screen space* is labeled with (I,J) coordinates

We can define the process of mapping a texture space onto the 3D model space as *p*arameterization and the process of mapping the 3D model space onto screen space as *v*iewing projection As only the parameterization is part of 3D modeling, the viewing projection which is part of image synthesis will not be investigated in this paper.

This paper will investigate the parameterization of polygon surface representing the 3D model. A linear parameterization of a triangular polygon is easily performed by specifying the related texture space coordinates $[u_i v_i], i = 1,2,3$ of each triangle vertex $[x_i, y_i, z_i], i = 1,2,3$. This information is sufficient to determine the parameters $a_i, i = 1 \cdots 9$ of an affine mapping from texture space to 3D model space which is given in homogeneous matrix notation by:

$$\begin{pmatrix} x & y & z \end{pmatrix} = \begin{pmatrix} u & v & 1 \end{pmatrix} . \begin{pmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & a_9 \end{pmatrix} \qquad (8)$$

If the triangles are sufficiently small, the error can be kept small.

3.2 Generate Cylindrical Image for Texture Mapping

As the image are taken in two-dimension, the horizontal edges of the object will be banned toward to photo. The resultant texture mapping on the 3D model is not matched at the boundary of triangle when the image come from different camera view. To solve this problem, a cylindrical image should be generated first. The method of generating panorama is based on the proposed algorithm from Szeliski and Shum[5]. However, some modification is necessary as their algorithm concerns environmental panoramic image mosaic.

To build a cylindrical panorama, a sequence of images is taken by a camera mounted on a turntable. If camera focal length or field of view is known, each perspective image can be wrapped into cylindrical coordinates. World coordinates $p = (X, Y, Z)$ is mapped to 2D cylindrical screen coordinates (θ, v) using

$$\theta = tan^{-1}(X/Z), v = \frac{Y}{\sqrt{X^2 + Z^2}} \qquad (9)$$

where θ is the panning angle and v is the scanline. Similarly, world coordinates can also be mapped into 2D spherical coordinates (θ, ϕ) using

$$\theta = tan^{-1}(X/Z), \phi = tan^{-1}(\frac{Y}{\sqrt{X^2 + Z^2}}) \qquad (10)$$

Once each input image have been wrapped, constructing the panoramic mosaic becomes a pure translation problem. Ideally, to build a cylindrical or spherical panorama from a horizontal panning sequence, the object in the image needed to be segmented first before the translational motion. Vertical jitter and optical twist can be compensated by a small vertical translations. Therefore, both a horizontal translation t_x and a vertical translation t_y are estimated for each input image.

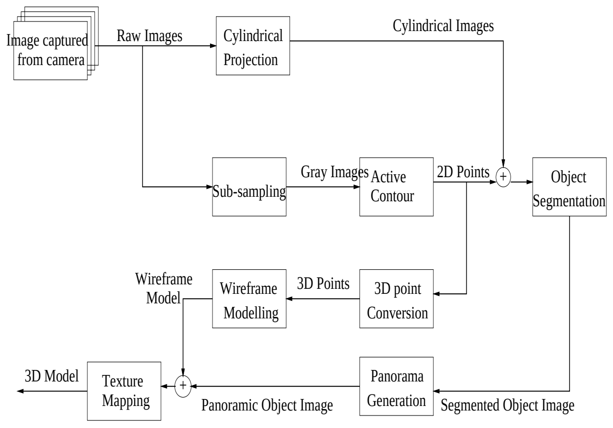


Figure 3 : System Block Diagram

To recover the translational motion, the incremental translation $\delta \mathbf{t} = (\delta t_x, \delta t_y)$ is estimated by minimizing the intensity error between two images,

$$E(\delta \mathbf{t}) = \sum_i [I_1(x'_i + \delta \mathbf{t}) - I_0(\mathbf{x_i})]^2 \qquad (11)$$

where $\mathbf{x_i} = (x_i, y_i)$ and $\mathbf{x'_i} = (x'_i, y'_i) = (x_i + t_x, y_i + t_y)$ are corresponding points in the two images, and $\mathbf{t} = (t_x, t_y)$ is the global translational motion field which is the same for all pixels. A simple feathering algorithm is applied to reduce the discontinuity in intensity and color between the images being compositied, i.e. the pixels in each image are weighted proportionally to their distance to the edge. Once registration is finished, a single panoramic object image can be written out.

4 System Integration

4.1 Point Acquisition Through Active Contour

This paper proposed an approach to object modeling based on above techniques of image processing. Figure 3 shows the system block diagram of the proposed system. The process begins with obtaining multiple view images of an object from a turntable. Each image

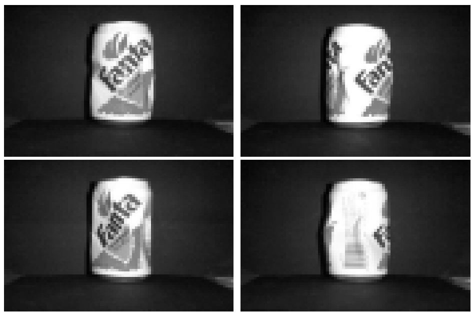


Figure 4 : Some Multiple View Images of a Deformed Can

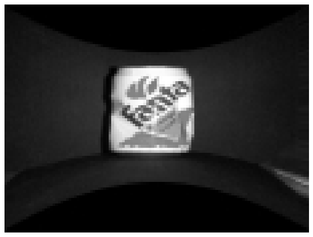


Figure 5 : Cylindrical Projection of a Deformed Can

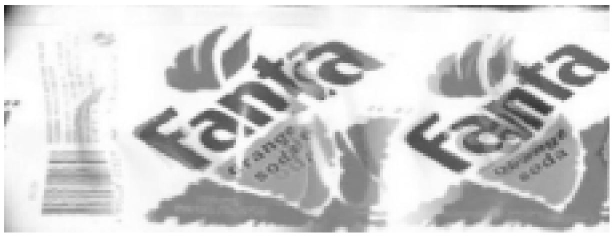


Figure 6 : Panoramic View of a Deformed Can

is then sub-sampled and converted into gray image, because if whole image is used in the snake algorithm, the computation time will be very long. These gray images are passed to the snake algorithm proposed by Xu and Prince[6]. The algorithm obtain the points from an initial snake. Then, the snake converges by several iterations based on the internal and external force and a final snake is resulted.

4.2 Object Segmentation and Panorama Generation

On the other hand, the original images need to be converted to cylindrical projected im-

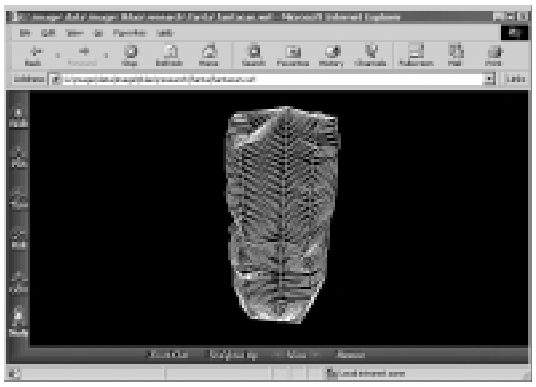


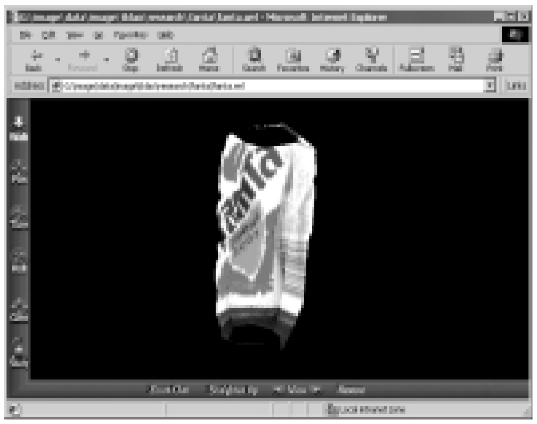
Figure 7 : Wireframe Model of a Deformed Can

ages in order to have better texture mapping. These images are first oversampled to about 1.2 times so that the resulting image would have the same length on y-axis because the y-axis of the image would be smaller than the original photo dimension.

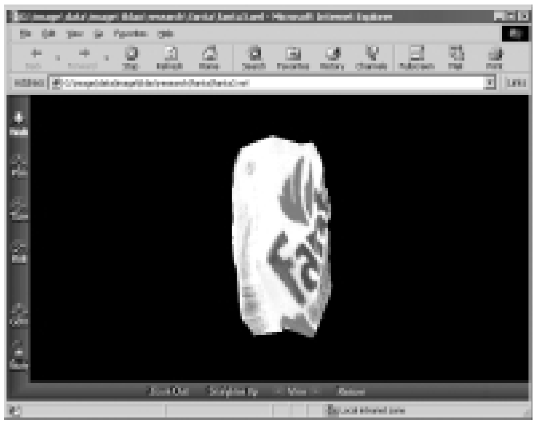
4.3 3D modeling and Texture Mapping

Meanwhile, the points obtained from the snake algorithm are needed to be manipulate first before making a wireframe model as the points are in 2D coordinates. A rotational matrix is used for the conversion of 2D points into 3D points. The points are rotated along the y-axis and the resultant points are in 3D coordinates. The 3D point is then used to generate the 3D model.

A wireframe model is generated through the 3D point. The points and faceset is written on a VRML V2.0 file. As the number of points are large enough, the polygons on the wireframe model are small so that the resultant object is more realistic and accurate. The last procedure is the texture mapping. The texture is mapped according to the faceset and places the panorama of the object on the polygons.



(a)



(b)

Figure 8 : Result of Object Modeling of a deformed Can

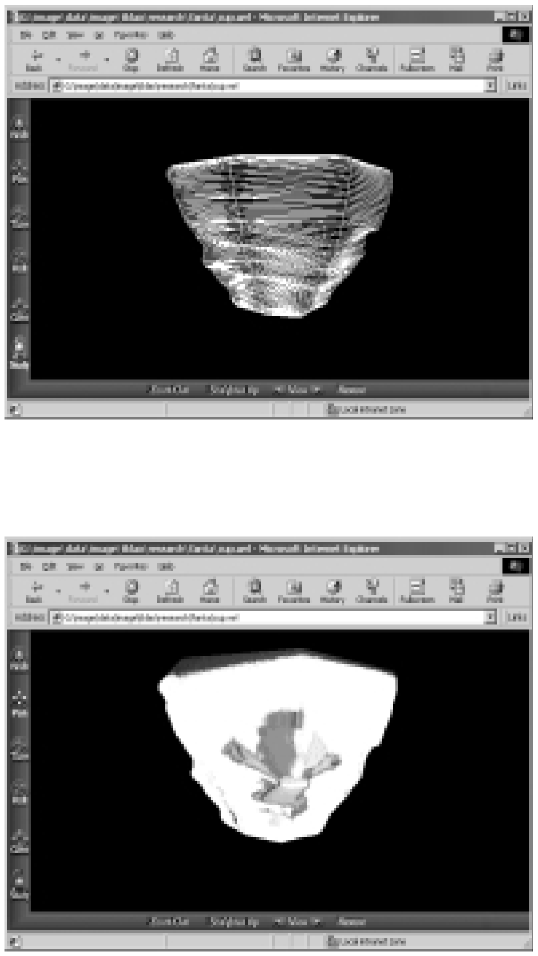


Figure 9 : Wireframe and Object Modeling of a Blow

5 Experimental Results

The final result of an object model is shown on Figure 8. This is a deformed can based on eight images taken from a camera. Figure 8a shows that the texture is mapped directly from the original image and Figure 8b shows that the texture is mapped from the panoramic view of the object. The boundaries between image in Figure 8a can be seen easily as they are come from different image. The intensity is not smoothed. In Figure 8b, the boundaries are removed as the texture is only come from one image. Figure 9 is another result using a blow as the object.

6 Conclusion

To conclude, the proposed method would give better result on cylindrical object. Regarding other shape of objects, further modification or adjustment is needed. The improvement of this method can be done by allowing different shapes of object. The quality of texture is another area needed to be improved. This method provide a new algorithm of object modeling without knowing the point correspondence and camera calibration. This is a convenience method which can be implement into a commercial product.

References

[1] Russell A. Manning and C. R. Dyer, *Dynamic View Morphing,* Technical Report 1387, Computer Science Department, University of Wisconsin-Madison, 1998.

[2] L. McMillian and G. Bishop, *Plenoptic Modelling: An Image-Based Rendering System,* Computer Graphic Proceedings, Annual Conference Series, 1995

[3] W. Niem and H. Broszio, *Mapping Texture from Multiple Camera Views onto 3D-Object Models for Computer Animation,* Proceedings of the International Workshop on Stereoscopic and Three Dimension Imaging, 1995

[4] Steven M. Seitz and C. R. Dyer, *Toward Image-Based Scene Representation Using View Morphing,* Proc. Intl. Conf. on Pattern Recognition, 1996

[5] R. Szeliski and H.Y. Shum, *Creating Full View Panoramic Image Mosaics and Environment Maps,* SIGGRAPH'97, Aug. 1997

[6] C. Xu and J. L. Prince, *Gradient Vector Flow: A New External Force for Snakes,* IEEE Proc. Conf. on CVPR, 1997.