# Object Tracking and Matting for A Class of Dynamic Image-based Representations

Zhi-Feng Gan, Shing-Chow Chan
Department of Electrical and Electronic Engineering
The University of Hong Kong, Hong Kong
{zfgan, scchan}@eee.hku.hk

Heung-Yeung Shum
Microsoft Research, Asia
P. R. China
hshum@microsoft.com

## Abstract

Image-based rendering (IBR) is an emerging technology for photo-realistic rendering of scenes from a collection of densely sampled images and videos. Recently, an object-based approach for a class of dynamic image-based representations called plenoptic videos was proposed. This paper proposes an automatic object tracking approach using the level-set method. Our tracking method, which utilizes both local and global features of the image sequences instead of global features exploited in previous approach, can achieve better tracking results for objects, especially with non-uniform energy distribution. Due to possible segmentation errors around object boundaries, natural matting with Bayesian approach is also incorporated into our system. Furthermore, a MPEG-4 like object-based algorithm is developed for compressing the plenoptic videos, which consist of the alpha maps, depth maps and textures of the segmented image-based objects from different video plenoptic streams. Experimental results show that satisfactory renderings can be obtained by the proposed approaches.

## 1. Introduction

Image-based rendering (IBR) is an emerging and promising technology for photo-realistic rendering of scenes and objects from a collection of densely sampled images and videos. Central to IBR is the plenoptic function [1], which forms a new framework for developing sophisticated virtual reality and visualization systems. Another important advantage of IBR is the superior image quality that it offers over 3D model building, especially for very complicated real world scenes. It also requires much less computational power for rendering, regardless of the scene complexity. The capturing, compression and effective rendering are the fundamental problems in IBR research [1]. In [2], an object-based approach for rendering and the compression of a class of dynamic image-based representations called plenoptic videos was proposed. The plenoptic video is a 4D plenoptic function and it is obtained by capturing videos, which are regularly placed along a series of line segments, instead of a 2D plane in the static light fields [2, 3]. The main motivation is to reduce the large dimensionality and excessive hardware cost in capturing dynamic representations. Despite the simplification employed, plenoptic videos can still provide a continuum of viewpoints, significant parallax and lighting changes along line segments joining the camera arrays. In the object-based approach, objects at large depth differences are segmented into layers for rendering and compression. An important advantage is that the rendering quality in large environment can be significantly improved, as demonstrated by the pop-up lightfields [4]. In addition, by coding the plenoptic videos at the object level, desirable functionalities such as scalability of contents, error resilience, and interactivity with individual IBR objects, can be achieved.

An important step in the object-based approach is to segment the objects in the video streams into layers or image-based objects with different depth values. The object extracting method proposed in [2] is based on Lazy snapping, which is semi-automatic in nature. To reduce the segmentation time for segmenting plenoptic videos, one possibility is to obtain an initial segmentation of the video at a particular view using semi-automatic tools and rely on tracking techniques to segment the objects at different views and at subsequent time instants. Towards this end, an automatic object tracking approach using the level-set method is proposed in this paper. Our method, which utilizes both local and global features of the image sequences instead of global features exploited in [5], can achieve better tracking results for objects, especially with non-uniform energy distribution.

After the objects in a plenoptic video have been extracted, they can be rendered separately using the estimated depth map. Due to depth discontinuity and segmentation errors, matting [4, 6] is usually performed. Basically, the foreground color $F$, the background color $B$, and the opacity component $\alpha$ (also called the alpha map for mixing) of each pixel around the object boundary are

0-7803-9385-6/05/$20.00 ©2005 IEEE

estimated. Using the alpha map and texture estimated, it is convenient to composite the image-based objects onto the background of the original or other plenoptic videos. The Bayesian approach in [6] is adopted in our system because of its good performance. Furthermore, a MPEG-4 like object-based compression algorithm is also developed for compressing the object-based plenoptic videos, which consist of the alpha maps, depth maps and textures of the segmented image-based objects from different video plenoptic streams.

The paper is organized as follows: the construction of the proposed capturing system is described in Section 2. The proposed object tracking algorithm is discussed in Section 3. The matting algorithm and the proposed object-based compression algorithm are presented in Sections 4. Experimental results are presented in Section 5. Finally, conclusions are drawn in Section 6.

## 2. The Capturing System for Plenoptic Video

Figure 1 shows the proposed plenoptic video system used to capture dynamic scenes. This system consists of two linear arrays of cameras, each hosting 6 JVC DR-DVP9ah video cameras. The spacing between successive cameras in the two linear arrays is 15cm and the angle between the arrays can be flexibly adjusted. More arrays can be connected together to form longer segments. Because the videos are recorded on tapes, the system is also more portable for capturing outdoor dynamic scenes. Along each linear camera array, a 4D simplified dynamic light field is captured, and the user's viewpoints are constrained along the linear arrays of video cameras. The use of multiple linear arrays allows the user to have more viewing freedom in sport events and other life performance. The proposed system represents a design tradeoff between simplicity and viewing freedom. Other configurations can also be employed. The cameras are calibrated using the method in [7]. In order to use this method to calibrate our camera array, a large reference grid was designed so that it can be seen simultaneously by all the cameras. Using the extracted intrinsic and extrinsic parameters of the cameras, the videos of the cameras can be rectified for rendering. After capturing, the video data stored on the tapes can be transmitted to computers through FireWire interface. All these components are relative inexpensive and they can readily be extended to include more cameras.

## 3. Object Tracking Using Level-set Method

As mentioned earlier, objects at large depth differences are segmented into layers and are compressed and rendered separately. This helps to avoid the artifacts at object boundaries due to depth discontinuities. In the proposed method, an initial segmentation of the objects is first obtained using semi-automatic approach. Tracking techniques are then employed to segment the objects at other video streams and subsequent time instants. Our method is based on the level set method or geometric partial differential equations (PDE). The use of PDE and curvature-driven flows in tracking, segmentation and image analysis has received great attenuation over the last few years [8]-[12]. The basic idea is to deform a given curve, surface, or image according to the PDE, and arrive at the desired result as the steady state solution of this PDE. The problem can also be viewed as minimizing a certain energy function:

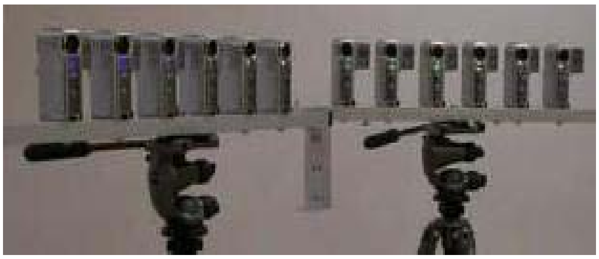


Figure 1. Two linear camera arrays, each consists of 6 JVC video cameras

$$U_I(C) = \int_I F(C,x)dx\,. \qquad (1)$$

as a function of a curve or surface *C*. The subscript indicates that the energy is computed from the given images *I*. Usually, *F*(*C*, *x*) is designed to measure the deviation of the desired curve from *C* at point *x*. To minimize the functional in (1), the variational approach can be employed to convert it to a partial differential function (PDF). A necessary condition for *C* to be a local minimum of the functional is $U_I^{'}(C) = 0$ . A general numerical approach is to start with an initial curve $C_0$ and let it evolve over a fictitious time variable *t* according to a PDE, which depends on the derivative $U_I^{'}(C)$ as follows:

$$\frac{\partial C(t)}{\partial t} = U_I^{'}(C(t))\,. \qquad (2)$$

However, conventionally finite difference methods are unsuitable to solve (2), because the PDE might be singular at certain points. A major breakthrough in solving (2) is due to Sethian and Osher [13], and the method is commonly referred to as the level-set method. The basic idea behind the level-set method is to represent a curve or surface in "implicit form" such as the zero level sets or isophone of a higher dimensional function. More formally, the time evolution of curves *C*(*x*, *t*) is represented as the level-set of an embedding function $\phi(x,t)$ :

$$L_c(x,t) := \{(x,t) \in R^2 : \phi(x,t) = c\}\,. \qquad (3)$$

where *c* is a given real constant. (2) can be rewritten as a PDE of $\phi(x,t)$ as follows:

$$\frac{\partial \phi(t)}{\partial t} = \beta \|\nabla \phi\| . \tag{4}$$

where $\beta$ is the velocity of the flow in the normal direction and it is derived from $U_I^{'}(C(t))$ above. The initial curve $C_0$ is associated with the level set with $c$=0, i.e. zero level set, and its time evolution is computed numerically by solving the following equation for $\phi(t)$, after discretizing at a sufficiently small time interval or step $\Delta t$:

$$\phi((n+1)\Delta t) = \phi(n\Delta t) + \Delta t \cdot G(\phi, x) . \tag{5}$$

where $G(\phi, x)$ is an appropriate approximation of the right hand side of (4). The desired solution is obtained when the PDE converges at sufficiently large value of *n*. For our object tracking problem, we define the following energy function for curve *C*:

$$U_I(C) = \alpha \int_I C_{inside} dxdy + \beta \int_I C_{outside} dxdy + \lambda Length(C) . \tag{6}$$

where $C_{inside}(x,y)$ and $C_{outside}(x,y)$ are two functions designed respectively to control the expansion and contraction of the curve *C* at location (*x*,*y*), and *Length*(*C*) measures the length of the curve. If we assume that the pixel values are independent and Gaussian distributed with means $c_{in}$ and $c_{out}$ respectively inside and outside the curve, then it can be shown that the PDE so obtained can be written as (details omitted due to page limitation):

$$\left.\frac{\partial \phi}{\partial t}\right|_{(x,y)} = \lambda \cdot div(\frac{\nabla \phi}{|\nabla \phi|}) - \alpha(u_{(x,y)} - c_{in})^2 + \beta(u_{(x,y)} - c_{out})^2 . \tag{7}$$

where $\alpha, \beta$ and $\lambda$ are positive parameters, $u_{(x,y)}$ is the value of pixel (*x, y*), $c_{in}$ denotes the driving force inside the curve *C*, and $c_{out}$ represents driving force outside the curve *C*. The third term, which is derived from *Length*(*C*), makes the curve smooth and continuous.

There are two different methods for determining $c_{in}$ and $c_{out}$ : global-based and local-based methods. The global-based method which is adopted in [5] utilizes all the pixels to drive curve *C*, where $c_{in}$ denotes the mean of all pixels inside the curve *C*, and $c_{out}$ is the mean of all pixels outside the curve *C*. There are many advantages about global-based method, e.g. fast evolution speed and insensitive to noise. However, some fine features along the boundary of the objects to be tracked might be lost. Figure 2 shows an example tracking result using the global-based method. It can be seen that the girl's right hand is outside the curve, because its mean is more similar to the background than to its body. On the contrary, local-based method uses local mean value inside a window instead of all the image pixels. In [12, 14], a local-base method is exploited, where $c_{in}$ and $c_{out}$ are set as follows: $c_{in} = u_{(x+i,y+j)}$, where $(u_{(x,y)} - u_{(x+i,y+j)})^2$ is the minimum value over all integer pairs $(i, j)$ such that $|i| \le m$ and $|j| \le m$ and pixel $(x+i, y+j)$ is inside the curve *C*; $c_{out} = u_{(x+i,y+j)}$ , where $(u_{(x,y)} - u_{(x+i,y+j)})^2$ is the minimum value over all integer pairs $(i, j)$ such that $|i| \le m$ and $|j| \le m$ and pixel $(x+i, y+j)$ is outside the curve *C*. Obviously, this method utilizes local features of the image to cope with objects having a non-uniform energy distribution. Unfortunately, this method would be sensitive to image noise, because only one pixel is chosen for determining both $c_{in}$ and $c_{out}$ . Here, we propose to combine the advantages of both the global-based and local-based methods by employing the following $c_{in}$ and $c_{out}$ :

$$\begin{cases} c_{in} = \text{average}(u_{(x+i,y+j)}), \text{where } |i| \le m, |j| \le m \text{ and} \\ \qquad \text{pixel } (x+i, y+j) \text{ is inside the curve } C \\ c_{out} = \text{average}(u_{(x+i,y+j)}), \text{where } |i| \le m, |j| \le m \text{ and} \\ \qquad \text{pixel } (x+i, y+j) \text{ is outside the curve } C \end{cases} . \tag{8}$$

## 4. Object Matting and Compression

As mentioned earlier, due to possible segmentation errors around boundaries and finite sampling at depth discontinuities, it is preferred to calculate a soft, instead of a hard, membership functions between the image-based objects and the background. In other words, the boundary pixels are assumed to be a linear combination of the corresponding pixels from the foreground and background:

$$I = \alpha F + (1 - \alpha) B . \tag{9}$$

where *I, F* and *B* are the pixel's composite, foreground and background colors, and $\alpha$ is the pixel's opacity component or the alpha map. Using this model, it is possible to matte a given object with the original at different views and other background. The digital analog of the matte (the $\alpha$ -map) is introduced by Porter and Duff [15] in 1984. In natural matting, all variables $\alpha$ , *F* and *B* need to be estimated and the problem is to find the most likely estimates for $\alpha$ , *F* and *B*, given the observation *I*. This can be formulated the maximization of the posteriori probability $P(F, B, \alpha | I)$ . Using the Bayesian rule, we have:

$$\max_{F,B,\alpha} P(F, B, \alpha | I) = \max_{F,B,\alpha} P(I | F, B, \alpha) P(F, B, \alpha) / P(I) . \tag{10}$$

Since the optimization parameters are independent of *P*(*I*), the latter can be dropped. Further, if $F, B, \alpha$ are assumed to be independent, then (10) can be written as:

$$\arg\max_{F,B,\alpha} P(F, B, \alpha | I) = \arg\max_{F,B,\alpha} P(I | F, B, \alpha) P(F) P(B) P(\alpha)$$
$$= \arg\max_{F,B,\alpha} \{\ln P(I | F, B, \alpha) + \ln P(F) + \ln P(B) + \ln P(\alpha)\}. \tag{11}$$

Taking the derivatives of (11), one gets a set of equations in the estimates of $\alpha$ , *F* and *B*. Interested readers are referred to [6] for more information.

It can be seen from above that for the proper rendering of an image-based object, we shall also need an alpha map and additional geometrical information in the form of a depth map, apart from its conventional texture pictures or maps. The alpha map is produced through the natural matting method discussed above, and they are used in the composition and rendering of the image-based objects. This information needs to be compressed for efficient storage and transmission of the plenoptic videos. We have adopted an object-based coding scheme, which shares many useful concepts with the MPEG-4 standard, except that it supports the compression of the depth maps and employs disparity compensation for better coding efficiency [16]. More precisely, the video streams of the plenoptic videos are divided into groups where one of the video streams called the main stream is encoded by the MPEG-4 algorithm without reference to other streams. The main stream then provides spatial prediction to adjacent secondary video streams, which employs both temporal and spatial predictions to better explore the redundancy in the video streams. Since alpha maps and depth maps are similar to the image textures, both of them are coded in the same way as the luminance component of the texture picture. We now present the experimental results of the proposed system.

## 5. Experimental Results

The performance of the proposed tracking method is evaluated using the *Dance* sequence captured by our IBR system. For each frame, the initial curve $C_0$ is the tracking result of the previous frame, and the object curve of the first frame is obtained manually. The level-set contour evolution is implemented using the narrow band method, where (7) is used as the speed function. The window size *m* for the local energy calculation is fixed to 6. The parameters $\alpha, \beta$ and $\lambda$ are not fixed, where they are dependent on $c_{in}$ and $c_{out}$ of each pixel. Figure 2 shows the tracking result of the global-based method. Figure 3(a)-(i) shows the tracking results on *Dance* sequence using our method, where the boundary is well delineated. The tracking results on synthetic *Ball* sequence using our method are demonstrated in Figure 4(a)-(h). It can be seen from the results that the proposed method gives more reasonable result for objects with non-uniform energy distribution.

The results of natural matting the image-based object are illustrated in Figure 5. Figures 5(a) and (b) show an example snapshot of a segmented image-based object called "dance" and its associated alpha map computed. Figures 5(c) and (d) show example renderings of the image-based object, after matting with two different backgrounds or scenes. Figure 6 shows the compression results for the texture and shape coding of the *Dance* sequence obtained by the proposed algorithm using the VM rate control algorithm. The curves denoted by "MPEG-4" represent the results using MPEG-4 without spatial prediction, while the ones denoted by "SP-3" and "SP-5" represent the coding results using the proposed algorithm with spatial prediction using 3 and 5 video streams, respectively. It can be seen that the proposed object-based coding scheme has a considerable improvement in PSNR performance (~1 dB) over the direct application of the MPEG-4 to individual video object streams. Details of the algorithms will be reported elsewhere.

## 6. Conclusion

We have proposed an automatic object tracking approach for a class of dynamic image-based representations called plenoptic videos based on the level-set method. Natural matting with Bayesian approach is employed to improve the rendering quality under depth discontinuity and possible segmentation errors， and it allows us to composite the image-based objects onto different plenoptic videos. Furthermore, a MPEG-4 like object-based algorithm is also developed for compressing the object-based plenoptic videos, which consist of the alpha maps, depth maps and textures of the segmented image-based objects. Experimental results using real-world sequences demonstrate the usefulness, good quality, and flexibility of the proposed approaches.

## 7. Acknowledgements

The Authors would like to thank Qing Wu and King-To Ng for producing the compression results.

## 8. References

[1] H. Y. Shum, S. B. Kang and S. C. Chan, "Survey of Image-based Representations and Compression Techniques," *IEEE Trans. on CSVT,* vol. 13, no. 11, pp. 1020-1037, Nov, 2003.

[2] Z. F. Gan, S. C. Chan, K. T. Ng and H. Y. Shum, "An Object-Based Approach to Plenoptic Videos," to appear in *Proc. of IEEE ISCAS'2005.*

[3] S. C. Chan, K. T Ng, Z. F. Gan, K. L. Chan and H. Y. Shum, "The Plenoptic Videos: Capturing, Rendering and Compression," in *Proc. of IEEE ISCAS'2004*, vol. 3, pp. 905-908, May, 2004.

[4] H. Y. Shum, J. Sun, S. Yamazaki, Y. Li and C. K. Tang, "Pop-up light field: An interactive image-based modeling and rendering system," *ACM Trans. on Graphics*, vol. 23, issue 2, pp. 143 -162, April 2004.

[5] T. F. Chan and L. A. Vese, "Active Contours Without Edges," *IEEE Trans. on Image Processing*, vol.10, no.2, Feb, 2001.

[6] Y. Y. Chuang, B. Curless, D.Salesin, and R. Szeliski, "A Bayesian Approach to Digital Matting," in *Proc. of IEEE CVPR'2001*, vol.2, pp.264-271, 2001

[7] Z. Zhang, "A Flexible New Technique for Camera Calibration," *IEEE Trans .on PAMI*, vol. 22(11), pp. 1330-1334, 2000.

[8] S. J. Osher and R. P. Fedkiw, "Level Set Methods and Dynamic Implicit Surfaces," Springer Verlag, 2002.

[9] N. Paragios and R. Deriche, “Geodesic Active Contours and Levels Sets for Detection and Tracking of Moving Objects,” *IEEE Trans. on PAMI*, vol.22, pp. 266-280, 2000.

[10] G. Sapiro, “Geometric Partial Differential Equations and Image Analysis,” Cambridge University Press, Cambridge, U.K., 2001.

[11] J. A. Sethian, “Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision and Materials Sciences,” Cambridge University Press, Cambridge, U.K., 1996.

[12] A. R. Mansouri, “Region Tracking via Level Set PDEs without Motion Computation,” *IEEE Trans. on PAMI*, vol.24, no.7, pp.947 – 961, July, 2002.

[13] S. J. Osher and J. A. Sethian, “Fronts Propagation with Curvature Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations，” *J. Compute. Phys*.79, pp.12-49, 1988.

[14] A. Yilmaz, X. Li and M. Shah, “Object Contour Tracking Using Level Sets,” in *Proc. of ACCV’2004*, Korea, 2004.

[15] T. Porter and T. Duff, “Compositing Digital Image,” In *SIGGRAPH’84*, pp. 253-259, July, 1984.

[16] S. C. Chan, K. T Ng, Z. F. Gan, K. L. Chan and H. Y. Shum, “The Compression of Simplified Dynamic Light Fields,” in *Proc. of IEEE ICASSP’2003,* vol. 3, pp. 653-656, April, 2003.

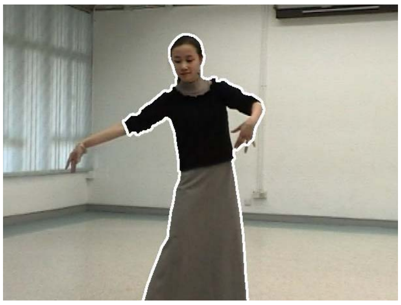


Figure 2. Tracking result of global-based method, the girl’s right hand is outside the curve.

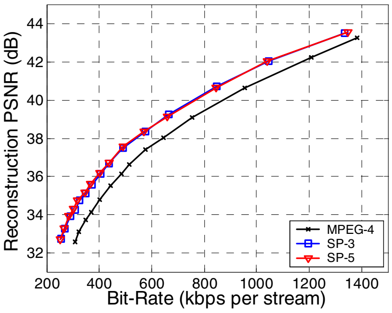
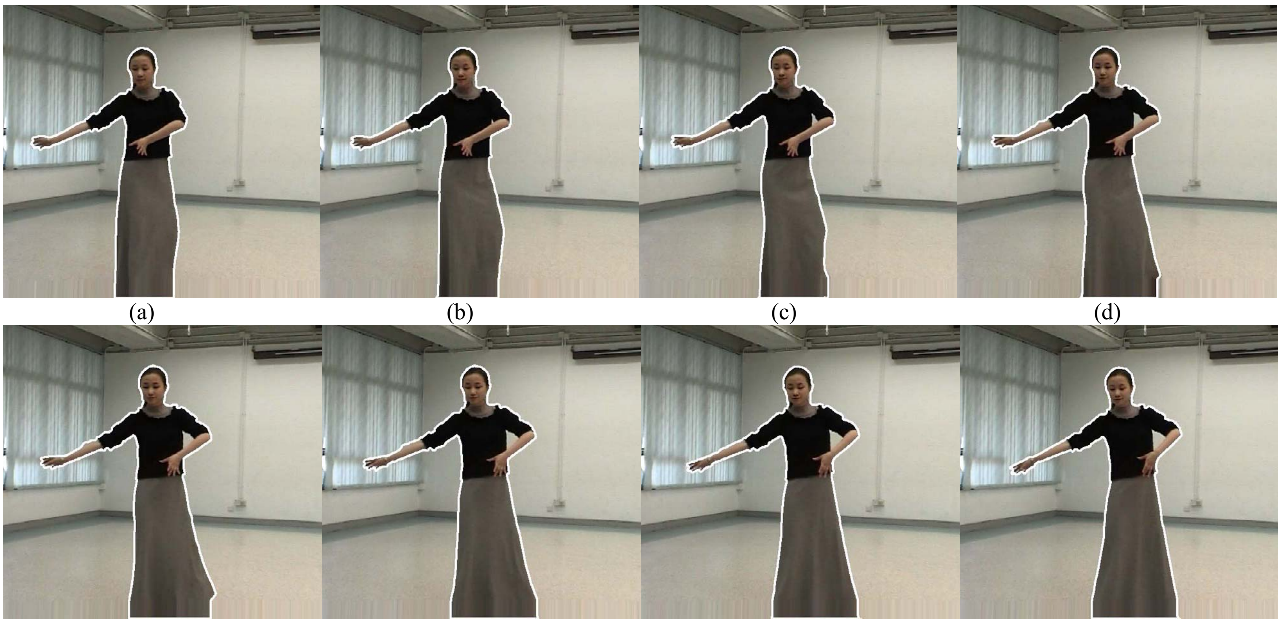


Figure 6. Compression results for IBR objects *Dance*.



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

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

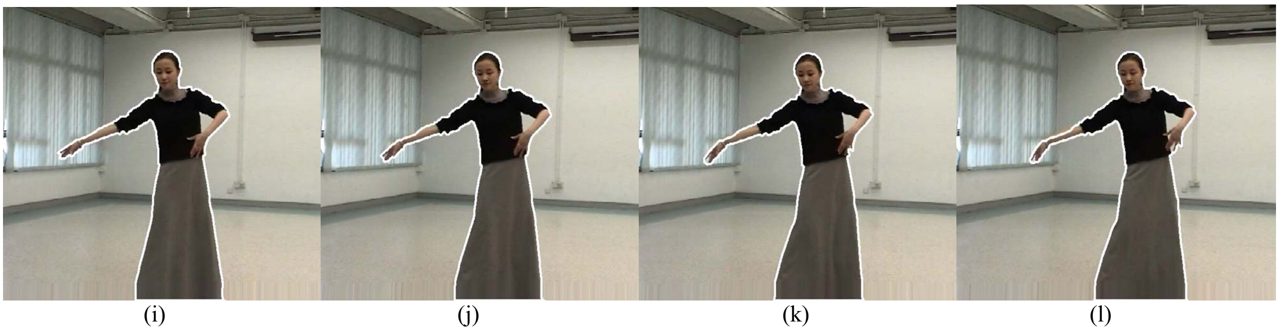
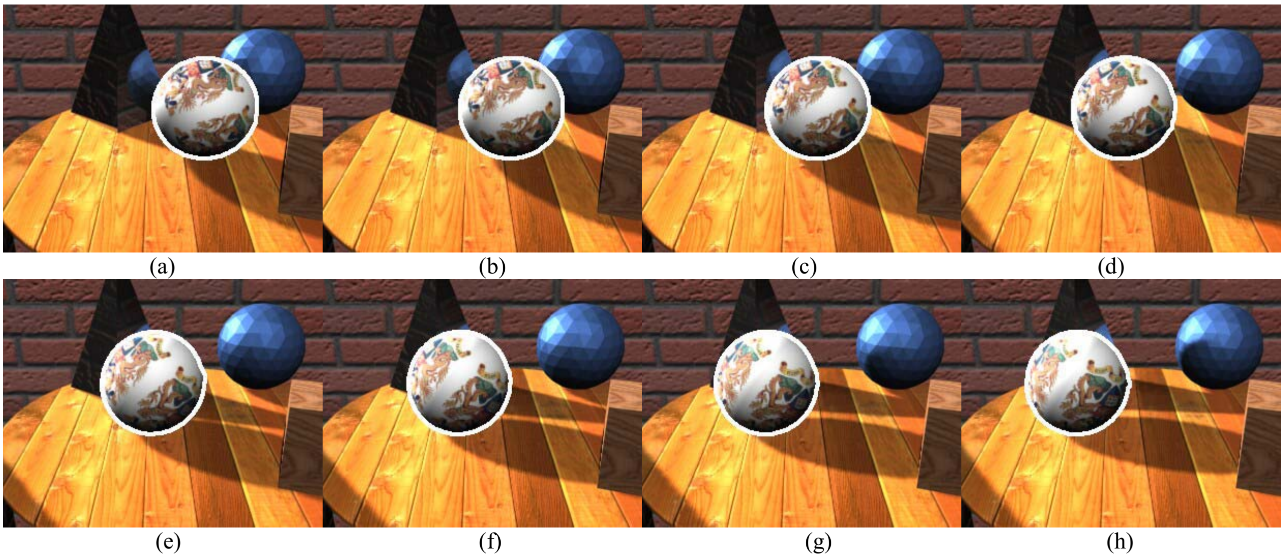

(i) (j) (k) (l)

Figure 3. (a)-(l) Tracking on the *Dance* sequence using our method. Image size: 720×576 .


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

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

Figure 4. (a)-(h) Tracking on the *Ball* sequence using our method. Image size: 320×240 .

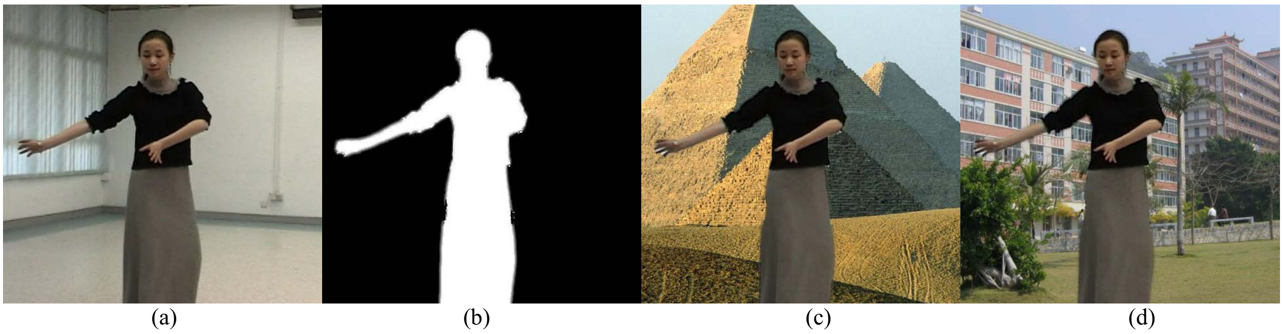

(a) (b) (c) (d)

Figure 5. (a) Input image. (b) alpha map. (c)-(d) New images of compositing extracted foreground over other background scenes.