

# Accurate Motion Layer Segmentation and Matting

Jiangjian Xiao

Mubarak Shah

Computer Vision Lab, School of Computer Science, University of Central Florida  
{jxiao, shah}@cs.ucf.edu

## Abstract

Given a video sequence, obtaining accurate layer segmentation and alpha matting is very important for various applications. However, when a non-textured or smooth area is present in the scene, the segmentation based on only single motion cue usually cannot provide satisfactory results. Conversely, the most matting approaches require a smooth assumption on foreground and background to obtain a good result. In this paper, we combine the merits of motion segmentation and alpha matting technique together to simultaneously achieve high-quality layer segmentation and alpha mattes. First, we explore a general occlusion constraint and design a novel graph cuts framework to solve the layer-based motion segmentation problem for the textured regions using multiple frames. Then, an alpha matting technique is further used to refine the segmentation and resolve the non-textured ambiguities by determining proper alpha values for the foreground and background respectively.

## 1 Introduction

Layer-based motion segmentation has been investigated by computer vision researchers for a long time [2, 9, 16]. Given a video sequence, motion segmentation consists of two major steps: (1) layer clustering which is to determine the number of layers in the scene and the associated motion parameters for each layer; (2) dense layer segmentation which is to assign each pixel in the image sequence to the corresponding layer and identify the occluded pixels. Currently, a number of approaches have been proposed for layer clustering problem, which have achieved good results, such as linear subspace [9] and hierarchical merging [14, 16]. However, once the initial layer clustering is achieved, how to correctly assign the pixels to different layers is a difficult problem [2, 9, 16] as shown in Fig.1. Particularly, if the images contains some non-textured regions such as the blue or white regions corresponding to the sky in Fig.1.a, the pixels in these regions may satisfy different layer motion parameters. Hence, the segmentation only using the motion cue may not provide accurate layer boundaries for these regions due to the ambiguities.

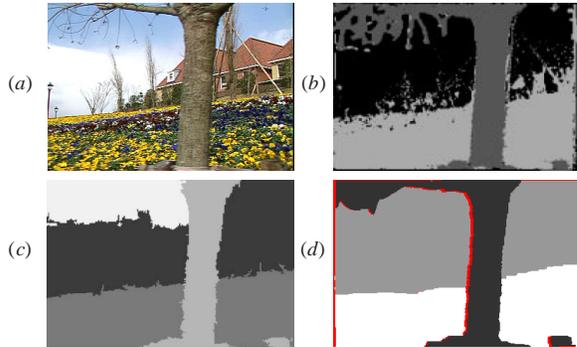


Figure 1: Previous results for flower-garden sequence. (a) One frame from the original sequence. (b) Result of Ayer and Sawhney [2]. (c) Result of Ke and Kanade [9]. (d) Result of [16], where the red pixels are occluded.

In digital matting, the observed color of the pixels around layer boundaries can be considered as a mixture of foreground and background colors, which is formulated as  $C = \alpha F + (1 - \alpha)B$ , where  $C$ ,  $F$ , and  $B$  are the observed, foreground, and background colors, and  $\alpha$  is the pixel's opacity channel. For single image matting, once a trimap is manually specified, the alpha values, foreground, and background colors of the unknown regions can be estimated under certain constraints [13, 12, 5]. However, the alpha matting technique is more suitable for smooth regions. Given a cluttered background, the performance of the existing alpha matting approaches tends to deteriorate.

In this paper, we propose an automatic approach, which combines the merits of motion segmentation and alpha matting technique together, to extract layer boundaries and alpha mattes simultaneously from a video sequence. First, a graph cuts framework is presented to solve the motion segmentation problem for the rich textured regions using multiple frames. In this framework, a novel three-node pixel graph is designed to perform multi-frame segmentation, which can correctly maintain the interactions between neighboring frame pairs and handle the general occlusion relationship among these frames. Based on the initial layer segmentation, we automatically estimate the trimap and the corresponding colors for the foreground and background. Then, after obtaining an initial alpha matte from this trimap,

we further enhance the mattes by employing the poisson editing technique [11]. Finally, applying the alpha mattes as a priori on multi-frame segmentation framework, a more precise layer segmentation is achieved and the occluded pixels between overlapping layers are also correctly identified.

## 2 Related work

In motion segmentation area, most of the existing approaches have achieved reasonable results for the layer clustering step when the motion parameters are distinct [2, 9, 16]. However, the results for the dense pixel assignment are not good in most cases due to an improper energy formulation or minimization as shown in Fig.1. Recently, graph cuts approaches [10, 4, 3] are proposed to successfully minimize energy functions for various computer vision problems. Kolmogorov et al. introduce an occlusion cost into graph cuts framework to solve stereo problem [10]. Willis et al. propose the use of graph cuts to extract layers between two wide baseline images [14]. Xiao et al. start from seed correspondences to estimate layer description [15], and then assign pixels into different layers using a graph cuts framework [16].

Using the motion information in video sequence, several approaches have been proposed to reduce the manual work for trimap generation and have achieved good matting results for two-layer scenes [6, 1, 8]. In such scenes, there are only two layers corresponding to the background and foreground. In [6], after manually specifying the trimaps for key frames, Chuang et al. propose a motion tracking technique to generate the trimaps for other frames. Apostoloff and Fitzgibbon present a background subtraction approach to automatically obtain the trimaps and recover the background [1]. In their approach, they require some prior knowledge to train the foreground model.

## 3 Multi-frame layer segmentation

In this section, we will address the following problem: Given an initial layer clustering, how to assign labels to pixels in a multiple labeling system in order to achieve an accurate layer segmentation. Before starting the discussion on the multi-frame layer segmentation issue, we first go through the relevant graph model and multi-way cuts framework. Then, we indicate the limitation of the existing graph model for multi-frame segmentation. Finally, we present a new model to remove the limitation.

### 3.1 Multi-way cuts and pixel graph

We formulate the multiple labeling problem into a multi-way graph cuts framework. In order to handle this NP-complete problem of the multi-way graph cuts [7], an  $\alpha$ -

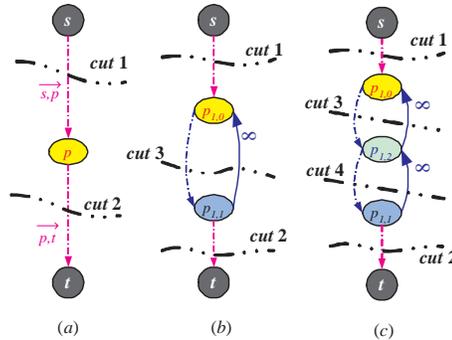


Figure 2: Pixel graphs. (a) A one-node pixel graph has two states which correspond to cut 1 ( $l_\alpha$ ) and 2 ( $l_o$ ) respectively. (b) A two-node pixel graph has three states which correspond to cut 1 ( $l_\alpha$ ), 2 ( $l_o$ ), and 3 ( $l_c$ ) respectively. (c) A three-node pixel graph designed for general occlusion constraint of multi-frame problem. It has four states corresponding to cut 1 ( $l_\alpha$ ), 2 ( $l_o$ ), 3, and 4 respectively. In both states 3 and 4, the pixel is assigned occlusion label ( $l_c$ ).

expansion algorithm is employed to iteratively minimize the total energy [4].

Traditionally, each node in the graph is only associated with one image pixel. Thus, each pixel has one individual two-state pixel graph as shown in Fig.2.a. In this case, during each step of  $\alpha$ -expansion, each pixel is either assigned a new label  $l_\alpha$  or it keeps its original label  $l_o$ , which can be naturally represented by state [0] or state [1] of each node respectively. However, in motion segmentation application, a pixel can be assigned one of three labels during one step of  $\alpha$ -expansion. These three labels are the new label  $l_\alpha$ , original label  $l_o$ , and occlusion label  $l_c$ , where the label set  $\mathcal{L}$  will be the union of occlusion and other layer labels:  $\mathcal{L} = \{l_1, l_2, \dots, l_k\} \cup \{l_c\}$ . Therefore, we need to introduce two nodes into the pixel graph which provides four states for a single pixel. After setting the weight of link  $\overrightarrow{p_{1,1}, p_{1,0}} = \infty$  as shown in Fig.2.b, only three possible combination ([0,0], [1,1], [0,1]) of these two nodes are available for this pixel, which correspond to three states of one step of  $\alpha$ -expansion and the state [1,0] is disabled.

### 3.2 General occlusion constraint

In order to overcome the limitation of the occlusion order constraint stated in [16], we need to revisit the occlusion problem in multiple frames. In general case, if the object is moving back and forth or the object is thin and moving fast, the occlusion area may not keep increasing and the occluded pixels may also re-appear in certain frames as shown in Fig. 3. After setting the first frame as the reference image, we can see that the pixels occluded between frame pair (1, 2) may re-appear and be assigned a layer label  $l_x \neq l_c$  in the other frame pair (1, j). If we use the occlusion order constraint of [16] to perform the segmentation, the incorrect results are obtained as shown in Fig. 3.a. Fig. 3.b shows

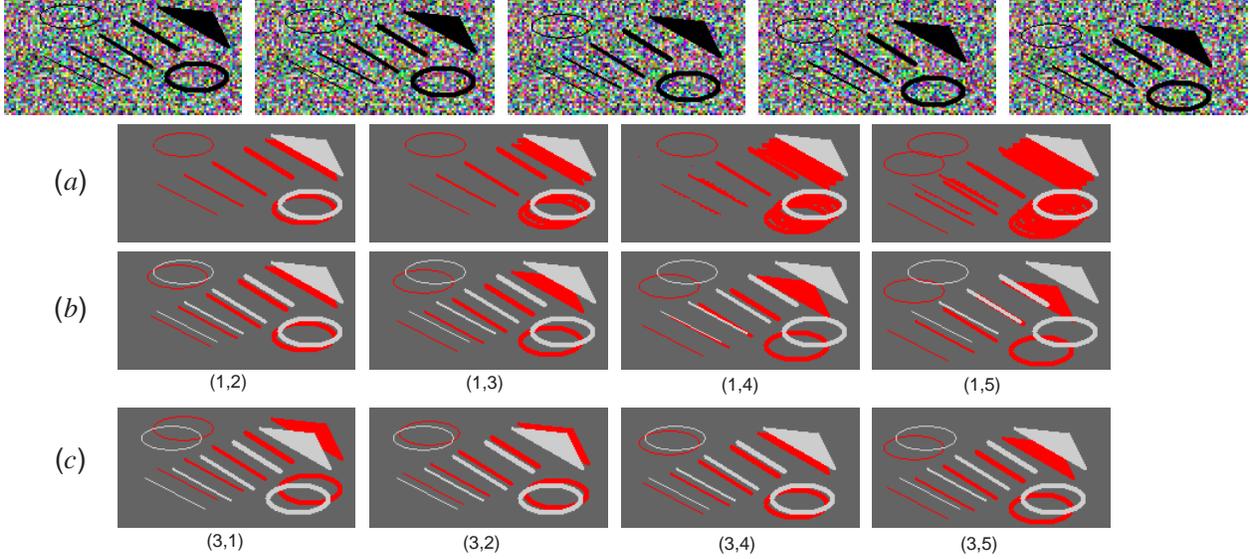


Figure 3: Multi-frame segmentation for a two-layer synthetic image sequence. Top: several black lines, two ellipses, and one triangle is moving from top-right to bottom-left on a random fixed background. (a) The results obtained using the occlusion order constraint [16], where the pixels (red) occluded between frame 1 and 2 will be always occluded between frame 1 and  $j$  ( $j > 2$ ). (b) The correct results by our approach using frame 1 as the reference image. (c) The correct results by our approach using frame 3 as the reference image.

that the occluded pixels are correctly detected by using our novel general occlusion constraint even for the lines with single pixel width.

**General Occlusion Constraint:** If a pixel  $p$  is assigned a layer label  $l_x \neq l_\zeta$  between frames  $i$  and  $j$ , then pixel  $p$  should either keep the same label  $l_x$  or be assigned the occlusion label  $l_\zeta$  between frames  $i$  and  $k$  where  $k \neq j$ .

According to this constraint, the reference image is not necessary to be set to frame 1. We can set any frame as the reference frame as shown in Fig. 3.c, where frame 3 is the reference frame.

Base on the general occlusion constraint, we design a multi-frame graph to solve the occlusion problem as shown in Fig.4, where a new three-node pixel graph (Fig.2.c) is employed. Compared to the two-node pixel graph (Fig.2.b), an additional node,  $p_{1,2}$ , is introduced to make the interactions between the pixels of the consecutive frame pairs, such as (1, 2) and (1, 3). In a single three-node pixel graph, there are four possible cuts, and two of them correspond to the occlusion case as shown in Fig.2.c. In this way, we introduce a new concept for the pixel graph design which breaks the mapping from traditional one-to-one (one pixel graph to one node) to one-to-multiple. Therefore, we can design more flexible graph primitives to implement a sophisticated graph for different labelling problems.

In Figure 4, each image pair is separated by the red dotted lines. In each image pair (1,  $j + 1$ ),  $j > 1$ , only the pixels in the reference image (e.g. frame 1) are assigned pixel graphs. For each pixel  $p_j^1$ , a pixel graph is cre-

<sup>1</sup> $p_j$  refers to the pixel in the reference frame of frame pair (1,  $j + 1$ ).

ated with three nodes  $p_{j,0}$ ,  $p_{j,1}$ , and  $p_{j,2}$ . In each image pair (1,  $j + 1$ ),  $p_j$  belongs to a pixel set,  $\mathcal{P}_j$ , where  $\mathcal{P}_j$  is the set of pixels in the reference image for the image pair (1,  $j + 1$ ).

After assigning the link weights using Table 1, the total energy of this multi-frame graph can be minimized by seeking a labeling system  $l$  such that

$$E = \sum_{j=1}^{n-1} (E_{sm_j}(l) + E_{d_j}(l) + E_{oc_j}(l)) + \sum_{j=1}^{n-2} E_{g_j}(l), \quad (1)$$

where  $E_{sm_j}(l)$  is smoothness energy,  $E_{d_j}(l)$  is data energy,  $E_{oc_j}(l)$  is occlusion energy in each frame pair, and  $E_{g_j}(l)$  is the occlusion energy between frame pairs.  $E_{sm_j}(l)$  is defined as

$$E_{sm_j}(l) = \sum_{i=0}^1 \left( \sum_{(p_j, q_j) \in \mathcal{N}} V(p_{j,i}, q_{j,i}) \cdot T(f_{p_j} \neq f_{q_j}) \right), \quad (2)$$

where  $\mathcal{N}$  is a 4-neighbor system,  $V(p_{j,i}, q_{j,i})$  is designed to more likely maintain the same label for  $p_{j,i}$  and  $q_{j,i}$  if they have similar intensities, and  $T(\cdot)$  is 1 if its argument is true and 0 otherwise. The remaining terms in Eq.1 are defined as:

$$E_{d_j}(l) = \sum_{p_j \in \mathcal{P}_j} (D(p_j, l_{p_j}) \cdot T(l_{p_j} \neq l_\zeta)), \quad (3)$$

where  $l_{p_j}$  is the label of pixel  $p_j$  and  $l_{p_j} \in \mathcal{L}$ ,  $D(p_j, l_{p_j})$  is the data penalty function of  $p_j$ . The occlusion energy

$$E_{oc_j}(l) = \sum_{p_j \in \mathcal{P}_j} (D_\zeta \cdot T(l_{p_j} = l_\zeta)), \quad (4)$$

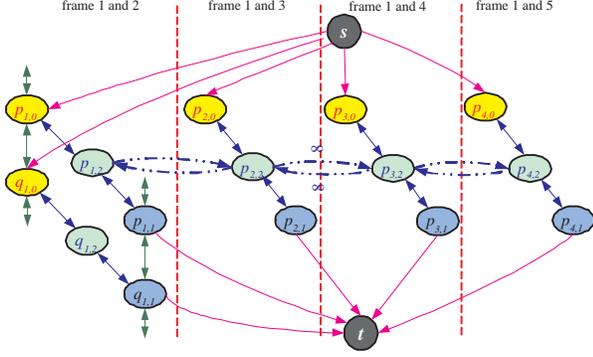


Figure 4: This graph is constructed using five consecutive frames, which have four image pairs related to the reference image. The red lines separate each pair of images into one block. The blue dotted  $n$ -links are introduced to connect  $p_{j,2}$ s to maintain the general occlusion constraint.

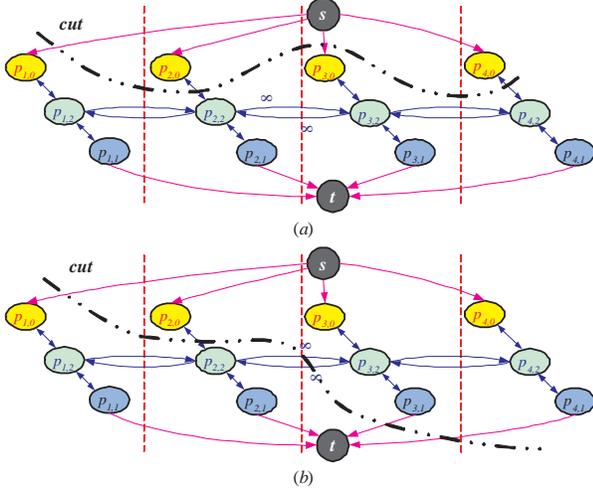


Figure 5: Valid and invalid cut of multi-frame graph model in one step  $\alpha$ -expansion. (a) a valid cut, where  $p_1$  and  $p_3$  will be assigned new label  $l_\alpha$ , and  $p_2$  and  $p_4$  will be assigned occlusion label  $l_\zeta$ . (b) an invalid cut that attempts to cut a  $n$ -link with weight  $\infty$ .

where the occlusion penalty,  $D_\zeta$ , is an empirical constant. In Eq.1, the last term is used to maintain the general occlusion constraint between frame pairs, which is defined as

$$\sum_{j=1}^{n-2} E_{g_j}(l) = \sum_{(p_j, p_k) \in \mathcal{P}} (\infty \cdot T(l_{p_j} \neq l_\zeta \wedge l_{p_k} \neq l_\zeta \wedge l_{p_j} \neq l_{p_k} \wedge j \neq k)),$$

where  $\mathcal{P} = \{\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_{n-1}\}$ . After adding  $\infty$   $n$ -links to connect the corresponding  $p_{j,2}$ s between the neighboring frame pairs, some cuts are disabled such as shown in Fig.5.b and c. Therefore, for the pixels at the same location in different frame pairs, such as  $p_1, p_2, \dots, p_{n-1}$ , no more than one kind of layer label will be assigned to them. For example, if pixel  $p_1$  in frame pair (1, 2) is assigned a layer label  $l_x$ , then pixel  $p_j$  in frame pair (1,  $j+1$ ) will be assigned

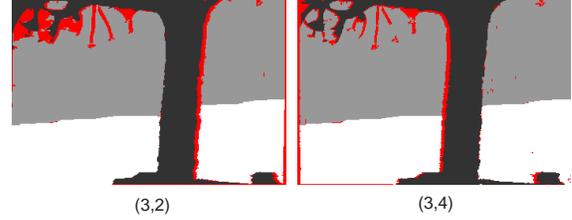


Figure 6: Segmentation results of flower-garden sequence. Here frame 3 is the reference image, two segmented images of frame pairs (3, 2) and (3, 4) are shown.

either the same label  $l_x$  or the occlusion label  $l_\zeta$ .

Fig.5.a shows one valid cut in one step of  $\alpha$ -expansion that is consistent with our general occlusion constraint. Fig.5.b shows one invalid cut, where the cut attempts to cut  $n$ -link  $\overrightarrow{p_{3,2}, p_{2,2}}$  with weight  $\infty$ . If this cut is allowed,  $p_1$  will be assigned a new label  $l_\alpha$ ,  $p_2$  will be occluded,  $p_3$  and  $p_4$  will keep the original label  $l_o$ . Here two possible layer labels  $l_\alpha$  and  $l_o$  may be assigned to the pixels at the same location in different frame pairs. This violates the observation that no more than one type of layer label should be assigned to the pixels at the same location in different frame pairs.

Fig.6 shows the segmentation results by using the proposed framework, where some of tree branches are segmented. Compared to Fig.1, our results are indeed better than those obtained by the other approaches. However, there are still a number of artifacts in the segmentation results, particularly in the non-textured region due to the motion ambiguities.

## 4 Alpha matting

With the aim of eliminating the segmentation artifacts, in this section we combine the merits of motion cue with the alpha matting techniques to automatically extract accurate matting and refine the layer segmentation. Fig.7 shows the block diagram of our video matting approach.

Once the initial segmentation of the reference image is

Table 1: Weights of the links.

Link	Weight	for
$\overrightarrow{p_{j,1}, t}$	$D(p_j, l_o)$	$p_j \in \mathcal{P}_j \wedge l_o \neq l_\zeta \wedge l_o \neq l_\alpha$
$\overrightarrow{p_{j,1}, t}$	$\infty$	$p_j \in \mathcal{P}_j \wedge (l_o = l_\zeta \vee l_o = l_\alpha)$
$\overrightarrow{s, p_{j,0}}$	$D(p_j, l_\alpha)$	$p_j \in \mathcal{P}_j$
$\overrightarrow{p_{j,0}, p_{j,2}}$ $\overrightarrow{p_{j,2}, p_{j,1}}$	$D_\zeta$	$p_j \in \mathcal{P}_j$
$\overrightarrow{p_{j,2}, p_{j,0}}$ $\overrightarrow{p_{j,1}, p_{j,2}}$	$\infty$	$p_j \in \mathcal{P}_j$
$\overrightarrow{p_{j,0}, q_{j,0}}$ $\overrightarrow{q_{j,0}, p_{j,0}}$	$V(p_{j,0}, q_{j,0})$	$\{p_j, q_j\} \in \mathcal{N} \wedge \{p_j, q_j\} \in \mathcal{P}_j$
$\overrightarrow{p_{j,1}, q_{j,1}}$ $\overrightarrow{q_{j,1}, p_{j,1}}$	$V(p_{j,1}, q_{j,1})$	$\{p_j, q_j\} \in \mathcal{N} \wedge \{p_j, q_j\} \in \mathcal{P}_j$
$\overrightarrow{p_{j,2}, p_{(j+1),2}}$ $\overrightarrow{p_{(j+1),2}, p_{j,2}}$	$\infty$	$p_j \in \mathcal{P}_j \wedge p_{(j+1)} \in \mathcal{P}_{(j+1)}$

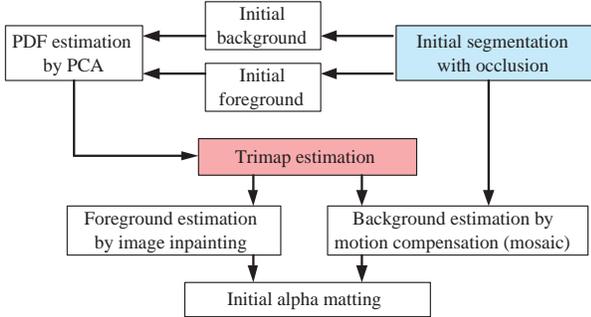


Figure 7: The block diagram of alpha matting process.

obtained, an initial trimap is directly determined, where the tree layer is the definite foreground,  $F$ , sky-house and flower layers are the definite background,  $B$ , and the occlusion areas are the unknown regions,  $U$ , as shown in Fig.8.c. It is obvious that this trimap is not good due to the imprecise motion segmentation.

Therefore, we have to refine  $F$  and  $B$  using their color distributions. To estimate the color distribution, we associate a 4-neighborhood system to each pixel and then form one data vector for this pixel, where the dimension of this data vector is  $5 \times 3$  (RGB). Instead of using one pixel, five neighboring pixels will provide more robust results typically for the noisy motion imagery. Next, we apply Principal Component Analysis (PCA) to all the data points and only the first few significant components are retained. After projecting the data on the new coordinates, a probability density function (PDF) can be obtained for  $F$  and  $B$  respectively by applying a proper Gaussian convolution. Fig.8.a – b show the PDFs using the first two components of PCA. Based on the PDFs, we can detect the pixels with low probability in both regions and mark them with the unknown label. Then, a shrinking process is applied to both  $F$  and  $B$  to further enlarge the unknown regions. As a result, we obtain a good trimap (Fig.8.d) for the next step of the matting process.

After determining the trimap, we need to estimate foreground color,  $U_F$ , and background color,  $U_B$ , for each pixel in the unknown region,  $U$ . For  $U_F$ , there is no way to find the pixels from any frames since they do not exist in this region. In this case, an inpainting technique is a good choice to propagate the color information from the definite foreground,  $F$ , to the unknown region. Fig.8.e shows using this approach we even can fill the whole image and not only restrict to  $U$ . To estimate  $U_B$ , we first apply motion compensation (mosaic) approach to fill some parts of  $U$  using the pixels from its neighboring frames. Then for the remaining uncovered holes, similarly we apply the inpainting approach to fill them as shown in Fig.8.f. After that, an initial alpha value,  $\alpha_0$ , in  $U$  can be computed by

$$\alpha_0 = \frac{(U_C - U_B) \cdot (U_F - U_B)}{\|U_F - U_B\|^2}, \quad (5)$$

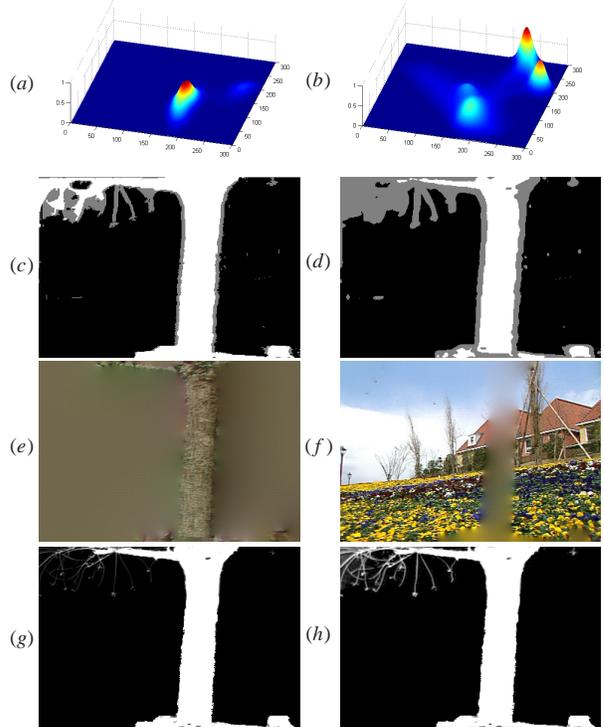


Figure 8: Alpha matting process. (a) and (b) are PDFs of  $F$  and  $B$  respectively after PCA. (c) and (d) are trimaps before and after using PDF estimation. (e) and (f) are the color estimations of foreground,  $U_F$ , and background,  $U_B$ , for the whole image. (g) and (h) are alpha mattes before and after the enhancement.

where  $U_C$  is the observed color of the pixel in  $U$ . Fig.8.g shows the initial alpha matting results. Then, we enforce a boundary condition and use  $\nabla\alpha_0$  as a guidance field to re-estimate the alpha value,  $\alpha$ , such that

$$\min_{\alpha} \iint_U |\nabla\alpha - \gamma\nabla\alpha_0|^2 \quad \text{with } (\alpha|_{\partial F} = 1 \wedge \alpha|_{\partial B} = 0),$$

where  $(\alpha|_{\partial F} = 1 \wedge \alpha|_{\partial B} = 0)$  is to enforce the boundary condition, and  $\gamma$  is a constant coefficient which can be used to scale the guidance field and enhance the new alpha mattes. Fig.8.h shows the final alpha matting results where the details of the branches are more clear than before and the noise is reduced.

## 5 Experiments

In the experiments, we tested our approach on both synthetic and real data. For the synthetic data, we achieved high-quality segmentation even for the thin objects with one pixel width as shown in Fig.3. For the real data, we demonstrate our results on two standard motion sequences and some additional sequences. *Note: More results are available on our web site [17].*



Figure 9: Left: segmentation results of three layers where the occluded pixels are marked by red. Middle: alpha mattes of the foreground layer. Right: new composited frames after superimposing the foreground of flower-garden in another video sequence.

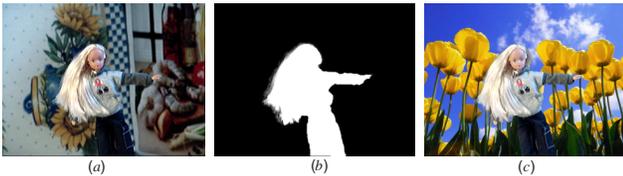


Figure 10: (a): One original frame from the “doll” sequence. (b): Alpha mattes of the doll. (c) Video composition on a still background.



Figure 11: (a) One original frame from the “mom-daughter” sequence. (b) Alpha mattes of the non-rigid foreground. (c) Video composition on a still background.

Fig.9 shows that after alpha matting step, we can further refine the segmentation by integrating the matting information. Fig.9 shows the segmentation and matting (or called soft segmentation) results from “flower-garden” sequence. At the bottom of Fig.9, we show one promising application of video composition, where the tree layer of “flower-garden” is transferred to another video sequence. Fig.10 shows one real sequence taken by a hand-held moving video camera (Fig.10). In this static scene, a doll is located in front of a chaotic background. Due to the depth variations, the motion parameters of the doll are different from the background. We have also extended our approach to background subtraction where the non-rigid foreground is considered as the occluded region of the background. Fig.11 shows the results for another standard sequence, “mom-daughter”. After obtaining the initial segmentation for each frame, we can generate a background mosaic using all background layers from the video sequence. Then, the proposed matting process is further used to estimate alpha channel for each frame.

## 6 Conclusions

In this paper, we presented an effective approach to correctly segment the layers and pull accurate alpha mattes for the video sequences containing 2-D motion. Our contributions consist of: (1) The formulation of a general occlusion constraint among multiple frames. (2) A novel three-node pixel graph is designed to solve the layer-based motion segmentation problem for textured regions. (3) An alpha matting technique to improve the segmentation around the non-textured ambiguous regions. After applying our approach to several challenging real and synthetic sequences, we achieved much better results compared to the other state-of-art approaches.

## References

- [1] N. Apostoloff and A. Fitzgibbon, “Bayesian Video Matting Using Learnt Image Priors”, CVPR, 2004.
- [2] S. Ayer and H. Sawhney, “Layered Representation of Motion Video Using Robust Maximum-Likelihood Estimation of Mixture Models and Mdl Encoding”, ICCV, 1995.
- [3] S. Birchfield and C. Tomasi, “Multiway Cut for Stereo and Motion with Slanted Surfaces”, ICCV, 1999.
- [4] Y. Boykov, O. Veksler, and R. Zabih, “Fast Approximate Energy Minimization via Graph Cuts”, PAMI, 23(11), 2001.
- [5] Y. Chuang, B. Curless, D. Salesin, and R. Szeliski, “A Bayesian Approach to Digital Matting”, CVPR, 2001.
- [6] Y. Chuang, A. Agarwala, B. Curless, D. Salesin, and R. Szeliski, “Video Matting of Complex Scenes”, SIGGRAPH, 2002.
- [7] E. Dahlhaus, D. Johnson, C. Papadimitriou, P. Seymour, and M. Yannakakis, “The Complexity of Multiway Cuts”, ACM Symp. Theory of Computing, 1992.
- [8] S. Hasinoff, S. Kang, and R. Szeliski, “Boundary matting for view synthesis”, IEEE Workshop on Image and Video Registration, 2004.
- [9] Q. Ke and T. Kanade, “Robust Subspace Clustering by Combined Use of kNND Metric and SVD Algorithm”, CVPR, 2004.
- [10] V. Kolmogorov and R. Zabih, “Visual Correspondence with Occlusions using Graph Cuts”, ICCV, 2001.
- [11] P. Perez, M. Gangnet, and A. Blake, “Poisson Image Editing”, SIGGRAPH, 2003.
- [12] M. Ruzon and C. Tomasi, “Alpha Estimation in Natural Images”, CVPR, 2000.
- [13] J. Sun, J. Jia, C. Tang, and H. Shum, “Poisson Matting”, SIGGRAPH, 2004.
- [14] J. Wills, S. Agarwal, and S. Belongie, “What Went Where”, CVPR, 2003.
- [15] J. Xiao, and M. Shah, “Two-Frame Wide Baseline Matching”, ICCV 2003.
- [16] J. Xiao, and M. Shah, “Motion Layer Extraction in the Presence of Occlusion using Graph Cut”, CVPR, 2004.
- [17] [http://www.cs.ucf.edu/~vision/projects/video\\_matting/](http://www.cs.ucf.edu/~vision/projects/video_matting/).