LAYER EXTRACTION USING GRAPH CUTS AND FEATURE TRACKING

Vardhman Jain [1], **P. J. Narayanan** [1]

[1] Center for Visual Information technology, IIIT Hyderabad, India
{vardhman@students., pjn@}iiit.net

Keywords: Layer Extraction, Video segmentation, Graph Cuts, Tracking, Interactive

Abstract

In this paper we present a new method for layer extraction by tracking a non-rigid body with no fixed motion model, in a video. The method integrates the graph cuts approach with robust point based tracking to achieve good tracking of the whole object over frames of a video. With the help of a little user interaction our method can perform fine layer extraction over irregular motion and difficult object boundaries. To achieve this we apply the 3D graph cuts on a pair of frames and propagate the labels obtained in the earlier frame to new frame by use of robust tracking method. The user is shown the results of the layer extraction and can provide extra strokes to improve the results.

1 Introduction

Layer extraction has been a topic of research in recent years. Many techniques have been proposed for automatic segmentation of layers [6, 13, 19, 20]. Though automatic segmentation of video is useful in many application like compression, coding, recognition etc. [20], Interactive segmentation of images [7, 11] and videos [8, 18] has developed recently. The superior quality they achieve with minimal user interaction makes them very attractive. These approaches have objectives similar to those of layer extraction. The extracted layers can be used in many applications of advanced video editing including Matting and Composition. The problem is also closely related to the object tracking problem which in itself has received lot of attention over the years.

The method we propose in this paper is based on the generally valid assumption that objects in the videos usually exhibit small motions over frames and also that the frames are temporally related.

There are certain issues which discourage the use of techniques which work one frame at a time and then combine the frames:

1. The object's segmentation over individual frames may not provide temporal continuity.
2. The information of segmentation obtained in earlier frames is not used.
3. The technique becomes very due to huge amount of recomputation at every frame.

In our method we try to address these problems. First we use a multi-frame graph which helps maintain temporal continuity and leverage the segmentation obtained in one frame to the other frame. We also effectively prune a large part of the image from being a part of the minimization process and thus making the graph smaller in terms of number of nodes and edges by making use of the assumption of trackability. Due to the use of robust tracking we are able to automatically provide hard constraints in the target frame which act as good seeds for the graph cuts minimization.

The layer obtained by our approach can then be used for variety of other applications like video cutout, matting, composition and object removal etc.

The paper is organized as follows. Section 2 describes the related work. Section 3 describes our approach in details. Results are demonstrated in Section 4.

2 Related Work

Layer extraction problem is closely related to various other problems like image and video segmentation, image and video matting and interactive image editing. Besides there are many applications of video segmentation including advanced video editing and object removal [21].

Image Segmentation: The problem of image segmentation has been around for a very long time. Earlier the techniques were based on clustering the image pixels based on some similarity criteria, which included intensity similarity or color similarity and spatial coherence [4, 17].

Later methods like image snapping [5] and intelligent scissors [9] tool in Adobe Photoshop which allowed user to obtain a contour around the object boundary by roughly tracking the object's boundary with the mouse, rather than requiring to drag the mouse precisely around the boundary were developed. These methods rely on local features like gradient information and Laplacian zero crossing measures and therefore they do not perform very well on highly textured regions where they can easily choose the wrong directions [11]. These methods could be termed as semi-interactive.

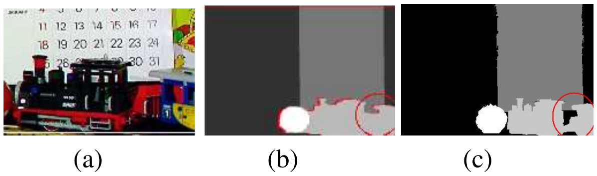
Interactive Image Segmentation: Recently techniques like Interactive Image segmentation [1], GrabCut [11] and Lazy Snapping [7] have demonstrated that with some small user input the segmentation of an image can be driven according to higher level context rather than the automatic color based segmentation techniques like watershed [17] or means shift segmentation [4] The interactive segmentation methods provide an easy way of segmenting complex objects from the image which would otherwise require tedious boundary selecting.

Matting: Matting is the process of obtaining accurate alpha values at the boundaries of the object, called the alpha matte of an object. Various techniques of matting have been proposed recently. We only discuss works related to natural image matting. Bayesian Matting [3] models color distributions probabilistically and alpha is obtained using MAP, which was also proposed earlier by Ruzon and Tomasi [12]. In most matting systems the user specifies a trimap to the system specifying pixels which are 100% foreground ($\alpha = 1$), 100% background ($\alpha = 0$) and for which the alpha is to be determined. The system then estimates the α values for the unknown region. Poisson Matting [15] provides good matting results but can need substantial application of the manual brushing tool or local Poisson matting. The problem solved by matting is quite similar to one of object segmentation, but with precise boundaries. The main requirement for most matting systems is the specification of proper trimap input. Matting techniques can be applied in cascade to our layer extraction method to obtain fine mattes.

Advances in the methods for image segmentation and matting [3, 15], and cutout have motivated the researchers to provide similar techniques for videos. Chaung *et al.* proposed video matting [2], where they propagate the user given trimaps for the key frames to the intermediate frames and apply image matting technique on each frame. As discussed by Li *et al.* [8] the dense optical flow can not be accurate determined for all the pixels and therefore errors creep in. Other advanced techniques like Interactive Video Cutout [18], Video Object Cut and Paste [8] allow extraction of the object from a video.

Layer Extraction: Layer extraction is an active area of research. Layer extraction methods usually rely on motion model estimation for set of regions followed by clustering techniques to cluster regions with similar motion models. In one of the earliest work on layer extraction, Adelson and Wang [19] proposed the patch-wise motion model estimation followed by clustering of patches with similar motion model. Ke and Kanade [6] formulated the problem of layer extraction by first expanding the seed region in to initial layers and then clustering these initial layer in lower dimensional subspace.

Xiao *et al.* [20] proposed a technique for layer extraction by first obtaining regions of seed correspondence and then growing them to arbitrary shapes using the graph cuts approach integrated with level sets based formulation. The reader is suggested to refer to [20] work for a more detailed survey of layer extraction work.



(a) (b) (c)

Figure 1: Advantage of interactive segmentation, shadow of train on calendar (a) can be regarded as part of the background layer in our case (c), unlike case the automatic case [20] (b). (marked by red circle)

Most of these techniques [6, 19, 20] target at automated layer extraction and in theory assume the existence of a prominent single motion model for a layer. In practice the object that we want to segment from the video may not show consistency in motion model across its spread, for example human motion.

Interactive methods are sometimes more suitable because user can guide the output more close to desired. For instance, the shadow of the object may possess the same motion model as the object but the user might like to exclude it from the foreground layer. Purely automatic techniques find this case difficult to handle as shown in Figure 1. The method we propose is suitable for handling relatively fast inter-frame motion for an object. The point based tracking ensures that the seeds are available over frames even if the layer's shape is changing quite often. This setup would require large number of key frames in the usual 3D graph cuts setting [1].

3 Layer Extraction Using Graph Cuts and Tracking

A block diagram of our system is shown in Figure 2. The steps of our approach are the following. The user first selects a set of key frames from the video on which using interactive image segmentation technique (s)he provides precise foreground/background segmentation (We use the term foreground to mean the layer to be extracted and background to all the pixels in the image which are not part of this layer.)

Using the segmentation provided in the key frame(s) and a robust tracking approach the seed points are generated for the intermediate frames. Our algorithm can proceed with just one key frame, the first frame. We build a 3D graph for each pair of frames using individual pixels as nodes of the graph. The N-D graph cuts technique [1] is then applied and the segmentation is achieved for the new frame. This is continued for all the frames in the video.

As the segmentation obtained automatically may not be satisfactory, the user can manually inspect the segmentation

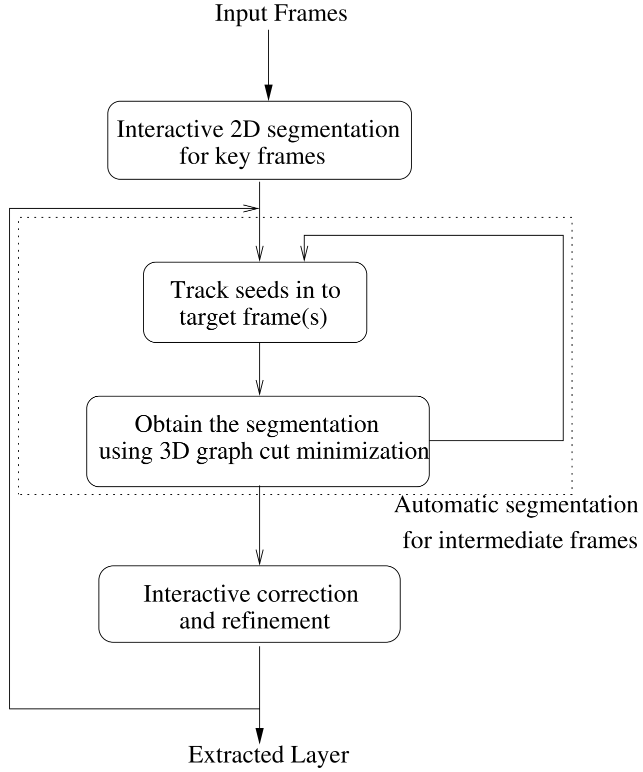

Input Frames
Interactive 2D segmentation for key frames
Track seeds in to target frame(s)
Obtain the segmentation using 3D graph cut minimization
Automatic segmentation for intermediate frames
Interactive correction and refinement
Extracted Layer

Figure 2: Overview of the system

result and provide extra strokes to improve the results. In the following subsections we provide the details of our approach:

3.1 Interactive Segmentation for Key Frames

Interactive segmentation is done for one or more key frames in the video. This step is based on the interactive segmentation method proposed by Boykov and Jolly [1]. In this step the user gives a few strokes to mark the foreground and background region in the image.

As Li *et al.* suggested in Video object cut and paste [8], other advanced tools like Lazy Snapping [7] can also be used for the purpose of key frame segmentation. In our approach which moves only in forward direction of frames, it is sufficient to start with just the first frame as the key frame. During the process, whenever the user desires, the frame can be segmented from scratch and effectively become a key frame.

3.2 Automatic Propagation of Segmentation

Various approaches [1, 8, 18] discussed the applicability of the min-cut in more than two dimensional data. A 3D graph is obtained by visualizing a set of images as planes and connecting the pixels in these images to the pixel of neighboring frames besides connecting them to the neighboring pixels in the same frame. Unlike the previous approaches of either not giving hard constraints in the intermediate frames [1, 8] or taking them through from the user [18], we propose a novel approach to obtain the hard constraints automatically. Based on the user provided segmentation of the first frame, we obtain good features points inside both foreground and background regions [14]. These features are then tracked over to the next frame where they are used for setting the hard constraints.

3.2.1 Propagation Step

The main contribution of the paper is the idea of using robust correspondence to propagate the seed points from one frame to another. Graph cuts algorithm by nature depends on the seed values. In our approach we use KLT tracking [14, 16] which tracks given feature points from one frame to another. We track two kinds of points, one set is obtained as a set of pixels which are good features to track [14] and second is a set of pixels spread evenly in the image. The KLT algorithm tracks these points in the next frame, those points can not be tracked "confidently" are ignored. Confidence is measured in terms of residual error per pixel. We use a value of 10 as threshold in our experiments. In practice any good tracking algorithm can be used for this purpose.

As shown in Figure 4, we label the points tracked from the background region in the source frame as background in the target frame and similarly for the foreground pixels.

3.2.2 3D Graph Construction

In our graph we only consider two frames at a time. The first frame is one which has been segmented either by the user manually (key frame) or by the algorithm automatically. The next frame is the frame which has to be segmented. Each pixel in the image is connected to its 8 neighbors in the same frame and also to 9 neighboring pixels in the next frame, as shown in the Figure 3. We can keep more connected graph in theory but our experiments show that this much connectivity gives good results.

Now we define the energy terms for the min-cut algorithm. The energy that needs to be minimized can be seen as the sum of three terms [8]

$$E = \sum E_1(p, f_p) + \lambda_1 \sum_{(p,q)\in V_I} E_2(p, q, f_p, f_q) + \lambda_2 \sum_{(p,q)\in V_c)} E_3(p, q, f_p, f_q) \qquad (1)$$

where f_p is the foreground/background label for the pixel p. λ_i denote the relative importance of the terms, we have used values $\lambda_1 = 10$ and $\lambda_2 = 1$ in our experiments.

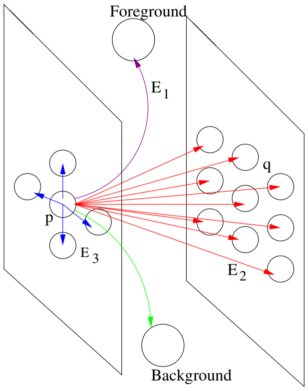


Figure 3: The 3D graph construction. Every pixel p in the graph is connected to 8 neighbors in same frame (only 4 shown, marked by blue edges), and 9 pixels in the neighboring frame, marked by red edges, and to the two terminal nodes namely the source(foreground) and sink(background) marked in cyan and green colors respectively. The energy for the three types of connections are E_2, E_3 and E_1 respectively

The term $E_1(p, f_p)$ denotes the data energy [11] term. It is the penalty of labeling the pixel p as f_p. Boykov and Jolly [1] defined this energy term based on "similarity" of the pixel intensity to the gray scale histogram for Foreground and Background. More recently RGB color space processing has been preferred and the histogram are replaced by Gaussian Mixture Models (GMMs)

GMMs have been commonly used in many recent works [3, 8, 12] to represent the foreground and backgrounds pixels. We use the method originally proposed by [10] for obtaining the approximate GMM, from the user segmented images. Let us denote the components of the foreground GMMs by (μ_m, Σ_m, w_m) for $m \in [1, M]$, where M is number of Gaussians in the model. We have used of the value of $M = 6$ in our experiments.

For a pixel color c, the distance to the foreground GMMs is defined as [11, 8]

$$d^f = \min_{m \in [1,M]} [D(w^f_m, \Sigma^f_m) + D(c, \mu^f_m, \Sigma^f_m)] \quad (2)$$

where

$$D(w, \Sigma) = -\log w + \frac{1}{2} \log |\Sigma|, \quad (3)$$

and

$$D(c, \mu, \Sigma) = \frac{1}{2}(c - \mu)^T \Sigma^{-1} (c - \mu) \quad (4)$$

Our definition of E_1 is similar to one proposed by Boykov and Jolly [1]. The term's value for a seeds points is set very high (∞) to the seed's label node (source or sink) and very small (0) to the opposite label. The value for a non seed point is set to be the distance d^f and d^b for the edge to the background and foreground respectively.

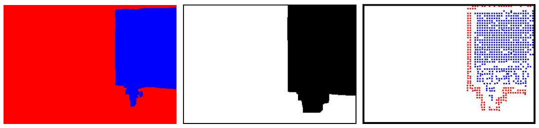


Figure 4: The tracking process: (a) The calendar layer is shown segmented in source frame, (b) The 'estimated region mask' to decide which pixels (shown in white) in the image will be included in graph cuts minimization for segmentation of next frame, (c) The seed points or hard constrains ts obtained using reliable tracking of points from the source frame (red indicates background and blue indicates background

The terms E_2 and E_3 denote the interaction penalty for intra-frame neighboring pixels and the pixels in the neighboring frame. We define these values using the well known interaction penalty measure [1]:

$$E(p, q, f_p, f_q) = |f_p - f_q|. \exp\{-\frac{||c_p - c_q||^2}{2 * \sigma^2}\}. \frac{1}{dist(p, q)}. \quad (5)$$

where $||c_p - c_q||^2$ is the Euclidean distance of the color values of pixel p and q. The term σ can be described as a parameter weighing the contrast. A high value of sigma puts a low penalty on high color difference and vice versa. The term $|f_p - f_q|$ ensures that the penalty is taken only for the boundary values [1]. We used value $\sigma = 50$ for our experiments.

To summarize the various steps for extracting a single layer in the sequence are as follows:

1. If frame is key frame skip to step 5.
2. Load the previously segmented frame on graph, all the pixels are either background or foreground the labeling of these pixels is not changed during the minimization.
3. Load the frame with needs to be segmented on the graph:
   (a) Track feature points from previous frame to current.
   (b) Set the successfully tracked points as hard constraints.
4. Run the graph cut on the 3D graph.
5. Set the current frame as previous and go to step 1.

3.3 Interactive Refinement

User interaction is needed to manually refine some of the labellings obtained in the intermediate frames during the process. In our system the user gives the corrective strokes in one of the frame and choose how for how many frames the automatic segmentation step has to be re-done. Once the segmentation is obtained for a particular frame, user can interactively modify the segmentation due to the use of efficient iterative max-flow algorithm on the original 3D graph. Unlike other approaches which have a final stage where user interaction can be applied, in our technique user can interact and improve the labellings (segmentation) at any intermediate frame. Interaction step is quite fast as we talk about in Section 4.

3.4 Speeding Up The Segmentation

A typical graph cut on the whole video could be quite slow due to the large number of pixels over which optimization is to be applied. As pointed out in Section 1 one of the main emphasis of our approach is to make the 3D graph cuts more efficient using the temporal and spatial continuity.

Consider the first frame of the 3D graph is segmented and we obtain the object mask. For the second frame the object's position can only be in some certain range of its previous position, called the estimated region mask. Based on this assumption we prune all the pixels which are not there in the union of the original mask and estimated region mask (Figure 4). The estimated region mask can be computed based on the estimated motion of the object and knowledge of motion model might be helpful. In our experiments we used a radial disc around the previous position as the estimated region mask. This prunes out large part of the image from the graph and boosts the efficiency by both avoiding the calculation of the energy terms and the actual running of the minimization algorithm. We also get hard constraints carried over the frames, this brings further efficiency as we can avoid calculating the complex energy values for these pixel positions. Finally we also use an iterative graph cuts algorithm and avoid the expensive from scratch optimization whenever possible.

4 Results

We show the layers extracted from the flowergarden sequence and the mobile calender (Figure 5 sequence). As can be seen in the Figure, Our algorithm does a good job of extracting the ball from the surrounding object many of which have similar colors. It should be noted that the ball's motion doesn't follow any specific motion. The train's shadow was also easily declared as part of the background layer as can be seen in Figure 1. Figure 6 shows another example where we segment the football and player as a single layer from the video.

In case of the flowergarden sequence for tree layer extraction the foreground tree matches in color with some of the background regions. In this case more user interactions were required to un-mark the spilling-in of the background in foreground regions and vice versa, but average interactive processing time was not more than a second per frame. The time required for interactive correction depends on boundary smoothness. The garden-house frame separation for example required just 3-5 strokes after the first key frame.

The time required for the segmentation depends on the object size as the graph size is dependent on it. For a small object like ball in the mobile-calendar sequence time taken on each iteration of 3D graph cuts is approx 1 sec, while for the calendar its around 2 sec. Iterative improvements on the graph are very fast and take less that 0.1 sec per optimization. All the experiments are performed on a Athlon 2600+ Machine, with 256MB RAM, the sequence had the frame size of 320x240. The overall processing time for one layer comes to around 2.5-4 seconds including the interaction. Therefore a 50 frames video can be processed in 3-4 minutes. Our approach has the advantage of allowing precise user inputs while performing 3D graph cuts on individual pair of frames.

5 Conclusions and Future Work

In this paper we proposed the idea of integrating robust feature tracking to seed the hard constraints in a 3D graph cuts minimization. This method can be used for a variety of purposes like video matting and layer extraction. Our method is currently limited to binary labeling. In our future work we would be investigating the feasibility of multi-label segmentation. Also we would like to develop faster algorithms for the same purpose by use of regions as primitives instead of current pixel level processing.

References

[1] Y.Y. Boykov and M.P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. In *ICCV01*, pages I: 105–112, 2001.

[2] Y. Chuang, A. Agarwala, B. Curless, D.H. . Salesin, and R. Szeliski. Video matting of complex scenes. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 243–248, New York, NY, USA, 2002. ACM Press.

[3] Y.Y. Chuang, B. Curless, D.H. Salesin, and R. Szeliski. A bayesian approach to digital matting. In *Proceedings of IEEE CVPR 2001*, volume 2, pages 264–271. IEEE Computer Society, December 2001.

[4] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(5):603–619, 2002.

[5] M. Gleicher. Image snapping. In *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 183–190, New York, NY, USA, 1995. ACM Press.

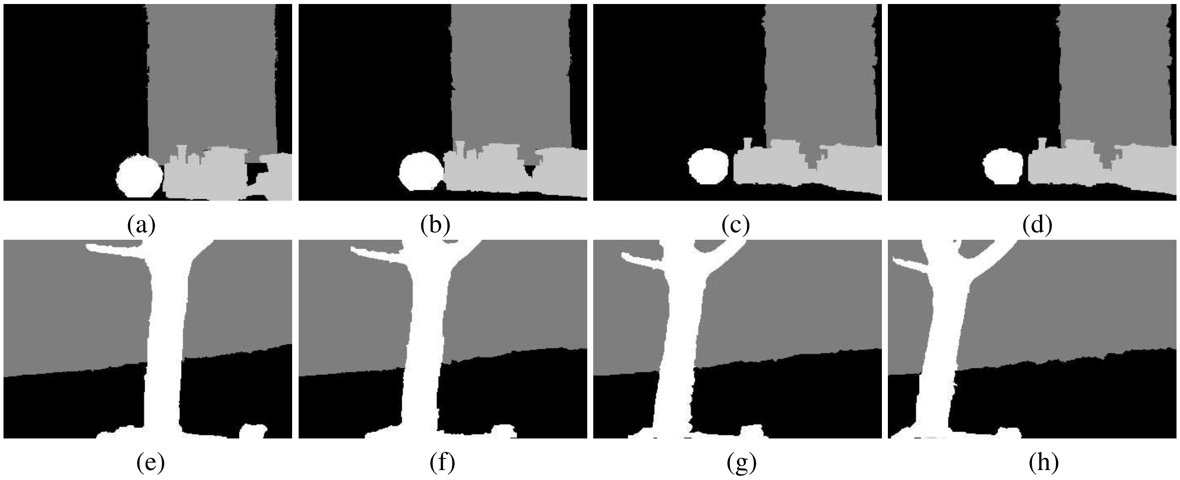


Figure 5: (a-d) Layers obtained by segmenting frames of the mobile-calendar sequence using our technique. (e-f) Layers obtained for the flower-garden sequence

[6] Q. Ke and T. Kanade. A subspace approach to layer extraction. In *CVPR*, 2001.

[7] Y. Li, J. Sun, Tang C.K., and H.Y. Shum. Lazy snapping. *ACM Trans. Graph.*, 23(3):303–308, 2004.

[8] Y. Li, J. Sun, and H. Y. Shum. Video object cut and paste. *ACM Trans. Graph.*, 24(3):595–600, 2005.

[9] E.N. Mortensen and W.A. Barrett. Interactive segmentation with intelligent scissors. *GMIP*, 60(5):349–384, September 1998.

[10] M.T. Orchard and C.A. Bouman. Color Quantization of Images. *IEEE Transactions on Signal Processing*, 39(12):2677–2690, 1991.

[11] C. Rother, V. Kolmogorov, and A. Blake. "grabcut": Interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23(3):309–314, 2004.

[12] M.A. Ruzon and C. Tomasi. Alpha estimation in natural images. In *CVPR*, pages 1018–1025, 2000.

[13] J. Shi and J. Malik. Motion segmentation and tracking using normalized cuts. In *ICCV*, pages 1154–1160, 1998.

[14] J Shi and C. Tomasi. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94)*, Seattle, June 1994.

[15] J. Sun, J. Jia, C.K. Tang, and H.Y. Shum. Poisson matting. *ACM Trans. Graph.*, 23(3):315–321, 2004.

[16] C. Tomasi and T. Kanade. Detection and tracking of point features. Technical Report CMU-CS-91-132, Carnegie Mellon University, April 1991.

[17] L. Vincent and P. Soille. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE PAMI, 1991*, 13(6):583–598, 1991.

[18] J. Wang, P. Bhat, A. Colburn, M. Agrawala, and Michael F.C. Interactive video cutout. *ACM Trans. Graph.*, 24(3):585–594, 2005.

[19] J.Y.A. Wang and E.H. Adelson. Layered representation for motion analysis. In *CVPR93*, pages 361–366, 1993.

[20] J. Xiao and M. Shah. Motion layer extraction in the presence of occlusion using graph cuts. In *CVPR04*, pages II: 972–979, 2004.

[21] Y. Zhang, J. Xiao, and M. Shah. Motion layer based object removal in videos. In *WACV05*, pages I: 516–521, 2005.

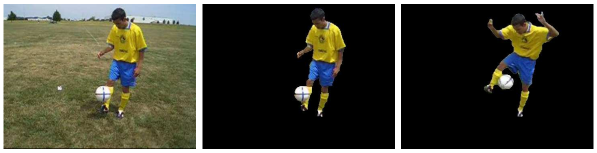


Figure 6: The football and player can be extracted as single layer by our algorithm even though their motions do not have any common motion model