Motion Layer Based Object Removal in Videos

Yunjun Zhang Jiangjian Xiao Mubarak Shah
Computer Vision Lab, School of Computer Science
University of Central Florida, Orlando, Florida 32816, USA

Abstract

This paper proposes a novel method to generate plausible video sequences after removing relatively large objects from the original videos. In order to maintain temporal coherence among the frames, a motion layer segmentation method is applied. Then, a set of synthesized layers are generated by applying motion compensation and region completion algorithm. Finally, a new video, in which the selected object is removed, is plausibly rendered given the synthesized layers and the motion parameters. A number of example videos are shown in the results to demonstrate the effectiveness of our method.

1. Introduction

The ability to remove large objects in videos is critical to many applications, such as video editing and film post-production. Given an input video, the goal is to remove the undesired objects and reconstruct the corresponding unknown regions in the entire video sequence based on the motion information. However, most of the current approaches [6, 1, 2, 5, 8, 11, 7, 13] only focus on the region completion in a single image. In this paper, we propose a novel approach to solve this problem, for the video sequences containing several planar motion layers. Our method based on the assumptions that the overlapping order of the motion layers in each frame is maintained the same throughout the input videos, and there has no cross occlusion between the layers in the video. For example, given that a video contains three layers 1, 2, and 3, if 1 occludes 2, and 2 occludes 3, 3 cannot occlude 1.

Based on this assumption, we first apply a level set representation and graph cut approach to achieve motion layer extraction. By exploiting the occlusion order constraints on multiple consecutive frames, the occluded pixels and the layer ordering are also explicitly determined. Then we remove the undesired layer (the large object) and locate the corresponding unknown areas in other layers for every frame. After selecting the reference frame, we apply the motion compensation to partially or even fully fill the unknown region in each layer. For the layers where some regions are still missing, we develop a graph cut based region completion algorithm to complete the missing data with the perceptually correct color-texture information. Finally, based on the layer motion parameters, we project the synthesized layers to render each new frame. Figure 1 illustrates our algorithm including the intermediated steps.

This paper is organized as follows. Section 2 reviews the previous work related to region completion. Section 3 addresses the details of our video completion algorithm in three steps. In Section 4, we demonstrate three sets of results obtained by our approach.

2 Previous Work

Most of the previous work for missing data recovery has been focused on single image completion. There are two primary categories of the work in this area. One was introduced by Bertalmio [1], who used PDE based method to repair damaged images. The idea is to extend the structures around the boundaries of the damaged area, and to fill the color information properly. For an image in which only small portions are missing, this approach can achieve highly smoothed results. However, in their results, the lack of texture in the large reconstructed area is easily visible. Therefore this approach is ineffective for filling in large holes in the natural images. Levin et al. [11] extended the idea by measuring the global image statistics, so that the inpainting results are based on the prior image knowledge besides the local color information.

Some researchers have considered texture synthesis based method as a way to achieve image completion [2, 5, 6, 8, 7]. Criminisi et al. used the angle between the isophote direction and the normal direction of the local boundary to define the searching order of the patches, so that the structure of the missing region can be filled before filling in the texture [5]. Jia and Tang [8] explicitly segmented the unknown area into different homogeneous texture areas using tensor voting. Drori et al. [6] incorporated pyramid image approximation and adaptive neighborhood size together to achieve impressive results. However, this method is slow due to the high computational complexity.

Recently some researchers started to address the video repairing problem. Bornard et al. used neighborhood-frame correction to repair damaged motion pictures [3].

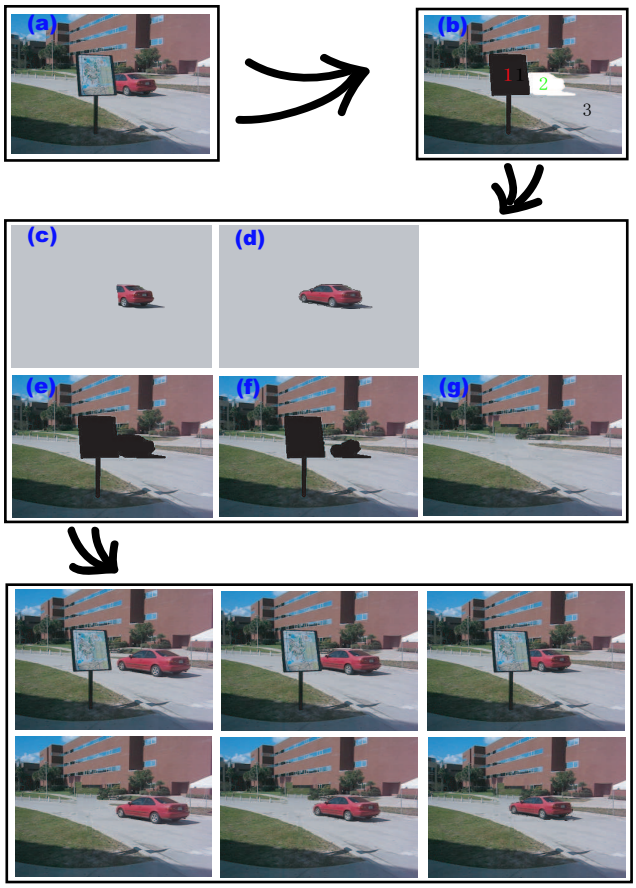

Figure 1: (a) One frame of the original video. (b) The result of the motion layer extraction. The layers are numbered based on their overlapping order. (c) and (e) All the layers except the one we want to remove. (d) and (f) Layers synthesized by applying motion information of all the frames in the video. (g) Pixels which are still unknown are filled with our region completion method. Bottom: Selected frames from the original and the synthesized video.

Bertalmio also mentioned video repairing in [2]. Wexler et al. filled the missing video portions by sampling spatio-temporal patches from other video portions, while enforcing global spatio-temporal consistency [14]. Another interesting work has been done by Jia et al. [9]. However, most of existing repairing methods do not use the motion layers and their orders in the videos. By explicitly extracting the motion layers and the order between the layers, our method can achieve very precise completion results.

3 Object Removal in Videos

Given an input video sequence, our goal is to remove relatively large objects in the video and fill in the removed area with reasonable color-texture information in all of the frames. Our algorithm consists of three main steps: (1) The video sequence is segmented into different motion layers, and the overlapping order among the layers is then determined. (2) After removing undesired object (one of the layers) in the video, the missing region in each layer of the reference frame is completed by motion compensation and region completion. (3) The completed layers in the reference frame is warped into every video frame to fill in the missing region of the frames.

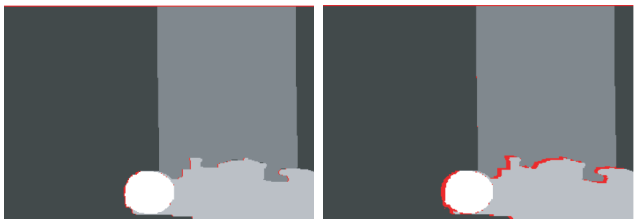

Figure 2: The final motion segmentation results using our multi-frame graph cut approach of two frames in mobile-calendar sequence. The red pixels are the occluded pixels.

3.1 Motion Layer Extraction

Given an input video, our segmentation algorithm can determine the number of the motion layer, and extract the accurate layers based on the motion parameters (affine or projective transformation).

In our approach, we first detect the seed correspondences over a short video clip [15]. Each patch around the seed correspondence is considered as an initial planar layer in the scene. Then, the region's boundary is gradually propagated along the normal direction using bi-partitioning graph cut algorithm integrated with the level set representation. Therefore, we can effectively filter out the bad seed regions. Then we design a two-step merging algorithm to merge these initial layers to into several groups, such that each group shares a single motion transformation. However, this layer merging method may not provide a correct segmentation of the scene, and the non-textured areas may belong to several layers due to their ambiguities, such as the white paper at the lower part of the calendar in the mobile-calendar sequence.

A graph cut algorithm is applied with an occlusion order constraint on multiple consecutive frames to obtain accurate layer segmentation [16]. At the same time, we also explicitly determine the occluded pixels. In our graph cut framework, this multi-frame motion segmentation problem is formulated as an energy minimization problem of the fol-

lowing function:

$$E = \sum_{j=1}^{n-1}(E_{smooth}(f) + E_{data}(f) + E_{occ}(f)) + \sum_{j=1}^{n-2} E_{order}(f),$$

where j is frame number, and n is the total number of frames (n usually is set to $3-5$). In this equation, E_{smooth} and E_{data} are standard terms in graph cut algorithm [4, 10], which correspond to piecewise smoothness penalty and data error penalty respectively. The other two terms are related to occlusion energy. The first one is $E_{occ}(f)$, which is used to impose the occlusion penalties for the occluded pixels between frames 1 and $(j+1)$. The second one is $E_{order}(f)$, which is used to impose occlusion order penalties for maintaining the occlusion order constraint on each consecutive pair of image pairs. After applying the graph cut algorithm, we extract the precise motion layers and explicitly identify the occlusion pixels between the overlapping layers as shown in Figure 2.

Giving the motion layer segmentation and the occlusion information between each pair of the layers, we use a simple approach to extract the overlapping order among the layers. For every pair of overlapping layers Γ_p and Γ_q in frame F_i, we denote the overlapped area by ρ_{pq}. If the correspondent area of ρ_{pq} in frame F_{i-1} belongs to Γ_p, Γ_p is on top of Γ_q or vice versa. Based on this scheme, every layer in the video is assigned with an order number, where the background layer is always assigned with 1. Figure 3 demonstrates the order of the layers in mobile-calendar sequences. Since the ball layer and the train layer do not overlap each other, and are on top of all other layers, they are numbered as a single layer.

3.2 Layer Compensation and Completion

Since each motion layer extracted from the previous step contains either a distinct object or the whole background, it is easy to remove the undesired object by deleting the corresponding layer. After removing the undesired layer, i, all the layers with smaller order numbers may have missing regions in some frames of the video. For each uncompleted layer, a motion model is applied to find the motion parameters between frames. Then, in each layer, a compensated reference frame is generated by warping all of the frames together with their motion parameters respectively as shown in Figure 1.d and f. In most cases, there may still be some large portions of layer missing color-texture information. In order to fill in the remaining missing regions, we propose a graph cut based single image completion method.

Before we explain the proposed method in details, few terms need to be defined. A known area is a region with all color-texture information available. An unknown area is a region with no color-texture information. A source patch is a small neighborhood area fully contained in known area.

Figure 3: This is the demonstration of the three layers in mobile-calendar sequences. The order of the layers is from the left to the right. In the real frame, the ball and and the train belong to the third layer, the calender is the second layer and background is the first layer.

A target patch has the same size of source patch, and is located on the boundary of unknown area. We denote the known area by Φ, the unknown area by Ω and the boundary of Ω by $\partial\Omega$. The source and target patches are denoted by Ψ_s and Ψ_t respectively.

Our method is based on non-parametric texture synthesis, therefore the filling order of the patches is critical to the quality of completion. In order to keep the performance at a reasonable level, we randomly select one patch from few potential patch locations on $\partial\Omega$ containing the largest known region on , and define it as the target patch, Ψ_t.

After determining the target patch, Ψ_t, a patch matching step is applied to find a source patch, Ψ_s in Φ, which has the best similarity with Ψ_t. We define the center of the previous target patch, the current target patch, the previous source patch, the current source patch as $\mathbf{x_{t1}}$, $\mathbf{x_{t2}}$, $\mathbf{x_{s1}}$ and $\mathbf{x_{s2}}$ respectively. If $\mathbf{x_{t1}}$ and $\mathbf{x_{t2}}$ are close enough, $\mathbf{x_{s2}}$ has a very high possibility to appear around $\mathbf{x_{s1}}$ due to the spatial similarity assumption. Therefore we can reduce the search space Φ_s from Φ to a neighborhood area around $\mathbf{x_{s1}}$, if the distance between $\mathbf{x_{t1}}$ and $\mathbf{x_{t2}}$ is within a threshold.

The similarity between the two patches can be expressed directly as follow:

$$\Psi_s = \arg\min_{\Psi_i \in \Phi_s} \frac{d(\Psi_t, \Psi_i)}{N_t},$$

where the distance $d(\Psi_t, \Psi_i)$ between the two patches is defined as the sum of squared difference (SSD), N_t is the number of pixels in the known area of the target patch, Ψ_t. It serves as a normalization factor.

This simple approach works well for images without projective deformation. However, for natural images, this condition may not be true. In order to handle the deforma-

tion, we estimate the projective transformation parameters between the two patches based on [12] before applying the similarity measurement.

After estimating the projective transformation parameters, the patch Ψ_s is warped to $\Psi_{\hat{s}}$ based on the motion parameters. Therefore the similarity measure can be calculated between Ψ_t and $\Psi_{\hat{s}}$. We then propose a new framework to update the patch Ψ_t using $\Psi_{\hat{s}}$. Instead of formulating this problem as a merging problem done by the previous researchers [5, 8, 6], we reformulate it as a cutting problem: giving two similar and spatially overlapping patches, where a cut should be made to separate those two patches and to maintain the seam least noticeable?

In our case, the patches can be cut only in the overlapping region, Ψ_o, where both patches have known information. We define each location in the overlapping region as a vertex v_i. Let $C_t(v_i)$ and $C_s(v_i)$ be the color value at the location v_i in Ψ_t and Ψ_s respectively. The bi-partitioning problem can be solved by minimizing the energy

$$\begin{aligned} E &= E_{smooth}(f) + E_{data}(f) \\ &= \sum_{(v_i,v_j)\in\mathcal{N}} W(v_i,v_j) + \sum_{v_i\in\Psi_o} D_p(f_i), \end{aligned}$$

where D_p is set to constant and the weight function, $W(v_i, v_j)$, between vertices v_i and v_j is defined as follows:

$$W(v_i,v_j) = \begin{cases} (\|C_t(v_i) - C_s(v_i)\| + \|C_t(v_j) - C_s(v_j)\|) \\ \qquad\qquad if\{v_i,v_j\} \in \mathcal{N}, \\ \infty \qquad\qquad otherwise \end{cases}$$

where function $\|\cdot\|$ denotes the Euclidean distance between color values, and $\mathcal{N}$ is a 4-connected neighborhood. After defining the weight function as above, the minimal cut can be easily computed by standard graph cut algorithm. A small weight means that if the cut runs between the pair of vertices, the four resulting color pairs $C_t(v_i)$ and $C_s(v_j)$, $C_s(v_i)$ and $C_t(v_j)$, $C_t(v_i)$ and $C_t(v_j)$, $C_s(v_i)$ and $C_s(v_j)$ do not have much difference. Therefore, the cut gives the least noticeable seam. On the contrary, the large weight between two vertices implies that a seam between the two vertices is more noticeable. Figure 4 shows two results for our region completion approach.

3.3 Frame Composition

After the layer compensation and completion, all of the color-texture information for each synthesized layer is available in the reference frame. The next step is to project the synthesized layers to render each new frame based on the layer motion parameters, which can effectively maintain the temporal consistency for all the frames in the video.

For each frame, F_i, in the video, the motion parameters with respect to the reference image can be computed by accumulating motion parameters of consecutive frames.

Figure 4: Left column: Layers with missing regions. Right column: The region completion results obtained by our method.

Therefore, the projection of the synthesized layers can be easily implemented by warping it into the corresponding position in the target frames. As a result, the missing region in the target layer is completed by the information from the synthesized layer.

4 Experiments

We tested our approach on three video sequences, mobile-calendar, car-map and statue-road, to demonstrate the effectiveness of the method. Figure 6 shows the results of five selected frames of the well known mobile-calendar sequence. Layer motion parameters are computed by using affine model. Each layer has its own distinctive motion.(calendar moving down, background moving right and ball and train moving left). Our results fully demonstrate the advantage of incorporating motion layer segmentation into the video completion framework.

Figure 7 shows the results selected from a car-map sequence. This sequence challenges our approach in several aspects: (1) The background has a strong perspective projection deformation. (2) The map board not only occludes the background but also occludes the moving car. A large portion of the body of the car is covered by the board in few frames in the sequence. In this case, directly computing the motion parameter gives noticeable misalignment after warping, since the car layer in different frames have different available area. The left bottom image in Figure 5 shows the results. In order to refine the result, a motion prediction approach is applied. We use a constant velocity assumption and apply it as the initial condition for the motion estima-

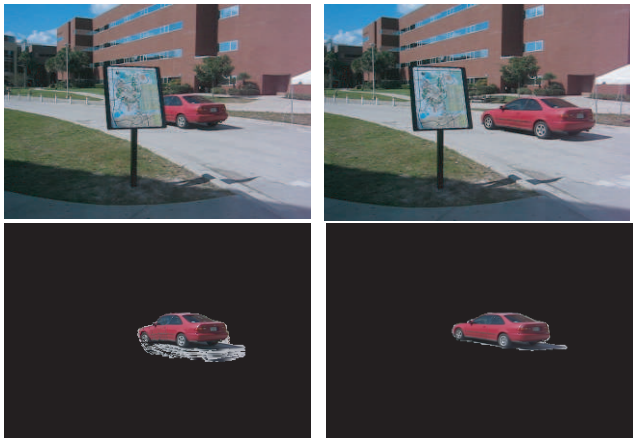

Figure 5: Left Top: A target frame in which a large region of the car layer is covered by the map board. Right Top: The reference frame in which the car layer is fully visible. Left Bottom: synthesized car layer by warping the car images based on the directly calculated motion parameters. Right Bottom: Final result of our method by applying the constant velocity constraint.

tion. By doing so, much better result is achieved and shown in right bottom image in Figure 5. Besides the compensation, our method successfully recovers the background and the full car body in every frame of the sequence. We did not remove the shadow of the map board because it does not interrupt the perceptual appearance of the output video. It can be easily removed by our method if needed.

The last results in Figure 8 are selected from a statue sequence. The original sequence is taken by a hand-held camera. The motion between frames is not smooth. Our method can still remove the statue in the scene in an unnoticeable manner.

5 Conclusions

A novel method is presented in this paper to solve the problem of object removal in videos. Our contributions mainly focus on three areas: (1) Incorporating the motion layer segmentation method into our framework. It not only segments the motion layers, but also retrieves the overlapping order among the layers. This is very crucial for correctly rendering the synthesized layers in the missing regions. (2) Introducing graph cut in single image completion to improve the quality of the completion results. (3) Applying layer motion compensation to maintain the completion consistency in the video sequences.

In the future, we will apply video matting approach to refine the composition quality and investigate the possibility of using statistical models in maintaining the consistency for the completed area in every frame.

References

[1] M. Bertalmio and G. Sapiro. Image inpainting. *SIGGRAPH*, 2000.

[2] M. Bertalmio, L. Vese, and G. Sapiro. Simultaneous structure and texture image inpainting. *CVPR*, 2003.

[3] R. Bornard, E. Lecan, L. Laborelli, and J.-H. Chenot. Missing data correction in still images and image sequences. In *Proceedings of the tenth ACM international conference on Multimedia*, 2002.

[4] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE PAMI*, 2001.

[5] A. Criminisi, P. Pérez, and K. Toyama. Object removal by exemplar-based inpainting. *CVPR*, 2003.

[6] I. Drori, D. Cohen-Or, and H. Yeshurun. Fragment-based image completion. *ACM Transactions on Graphics*, 2003.

[7] H. Igehy and L. Pereira. Image replacement through texture synthesis. In *ICIP*, 1997.

[8] J. Jia and C.-K. Tang. Image repairing: Robust image synthesis by adaptive nd tensor voting. *CVPR*, 2003.

[9] J. Jia, T.-P. Wu, W.-W. Tai, and C.-K. Tang. Video repairing: Inference of foreground and background under severe occlusion. *CVPR*, 2004.

[10] V. Kolmogorov and R. Zabih. Multi-camera scene reconstruction via graph cut. *ECCV*, 2002.

[11] A. Levin, A. Zomet, and Y. Weiss. Learning how to inpaint from global image statistics. *ICCV*, 2003.

[12] S. Mann and R. Picard. Video orbits of the projective group: a new perspective on image mosaicing. *Technical Report 338, MIT Technical Report*, 1995.

[13] J. Shen. Inpainting and the fundamental problem of image processing. *SIAM news*, 2003.

[14] Y. Wexler, E. shechtman, and M. Irani. Space-time video completion. *CVPR*, 2004.

[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.

Figure 6: Top row: Five selected frames from the original mobile-calendar sequence. Bottom row: The correspondent frames obtained by our algorithm, in which the train and the ball are removed.

Figure 7: Top row: Five selected frames from the original car-board sequence. Bottom row: The corresponding frames obtained by our algorithm, in which the map board is removed.

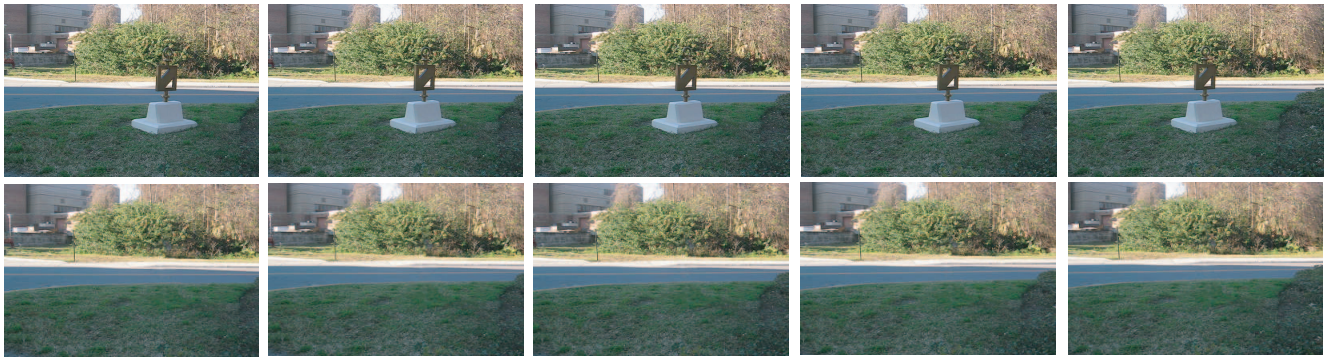

Figure 8: Top row: Five selected frames from the original statue sequence. Bottom row: The corresponding frames obtained by our algorithm, in which the statue is removed.