Moving Object Extraction with a Hand-held Camera

Guofeng Zhang[1] Jiaya Jia[2] Wei Xiong[2] Tien-Tsin Wong[2] Pheng-Ann Heng[2] Hujun Bao[1]*

[1]State Key Lab of CAD&CG, Zhejiang University
[2]The Chinese University of Hong Kong

Abstract

This paper presents a new method to detect and accurately extract the moving object from a video sequence taken by a hand-held camera. In order to extract the high quality moving foreground, previous approaches usually assume that the background is static or through only planar-perspective transformation. In our method, based on the robust motion estimation, we are capable of handling challenging videos where the background contains complex depth and the camera undergoes unknown motions. We propose the appearance and structure consistency constraint in 3D warping to robustly model the background, which greatly improves the foreground separation even on the object boundary. The estimated dense motion field and the bilayer segmentation result are iteratively refined where continuous and discrete optimizations are alternatively used. Experimental results of high quality moving object extraction from challenging videos demonstrate the effectiveness of our method.

1. Introduction

Accurate moving foreground extraction from a video is an active research in computer vision. For an applicable high-quality video editing tool, it is usually required that the moving foreground object can be robustly detected, separated, and edited. However, this is an inherently ill-posed problem due to the large number of unknowns and the possible geometric and motion ambiguities in the computation.

In order to separate the foreground object with visually plausible boundary, several bi-layer separation methods [5, 10, 15, 4] are proposed assuming that the camera is mostly stationary and the background is known or can be modeled. In [7], using the stereo video sequences and assuming the mostly static background, the object color, gradient, and displacement information are integrated to infer the foreground layer in real time. Later on, two approaches are proposed respectively in [15, 4] to separate the foreground from a single stationary web camera using different spatial and temporal priors. Most recently, an approach is proposed in [20] to infer bilayer segmentation monocularly even in the presence of distracting background motion. For all these methods, if the camera undergoes arbitrary translational and rotational motions and the background has complex geometry structures, the foreground object cannot be accurately extracted due to the following two main factors.

Motion estimation. In videos, the motion estimation errors are inevitable even using state-of-art algorithms [2, 18, 8]. The estimation inaccuracy may cause large problem in correctly constructing the background information and in modeling the background prior in segmentation.

Foreground definition. The moving camera also brings the difficulties in defining and identifying the foreground object. If the background is known, it is certain that the foreground can be detected and extracted using color constancy constraint or other more sophisticated tracking or Bayesian detection methods [19, 13]. However, if the background is unknown in the beginning, the foreground definition can be ambiguous. We shall show in this paper that labeling the pixels geometrically close to the camera or the pixels with large motions as foreground is NOT correct in many cases.

Besides bilayer separation, motion segmentation [1, 17, 6, 8] has been extensively studied during the past decades. The purpose of these methods is to group pixels that belong to similar motion, and eventually to cluster the motion to multiple layers. These methods do not aim to achieve the high quality foreground extraction and usually produce segmentation without accurate object boundaries especially when occlusion or disocclusion happen.

In this paper, we propose an automatic method to accurately detect and extract the moving object from a video sequence taken by a hand-held camera. Our method advances in several ways to improve the accuracy of the bilayer segmentation especially on the object boundary. Our method iterates between motion estimation and bilayer segmentation. In the motion estimation process, in order to minimize the errors caused by optical flow formulation, on

*Corresponding Author: Hujun Bao {bao@cad.zju.edu.cn} one hand, we introduce the occlusion parameter and propose the continuous-discrete optimization to avoid the local optimum. On the other hand, we apply the structure from motion technique to reliably recover the camera motion and sparse 3D points. These points map to different frames as reliable anchors to constrain the motion optimization.

In the bilayer segmentation, using the camera parameters, the multi-view geometry constraint is incorporated into our layer separation model. A novel appearance and structure consistency constraint in 3D warping is introduced in our approach to model the essential difference between the moving object and the background in the video. The final foreground is extracted by solving an optimization problem combining all these constraints and considering the temporal-spatial smoothness in video.

The paper is organized as follows. In Section 2, we give an overview of our method. In Section 3, the robust optical flow and dense depth estimation are described. The moving foreground detection and extraction are described in Section 4. Experimental results are shown in Section 5. We discuss and conclude our paper in Section 6.

2. Our Approach

Given a video sequence taken by a freely moving camera with n frames, our objective is to achieve a high quality foreground extraction without initially knowing the object motions. We denote $f^t(i)$ the pixel i in frame t. Then our goal in this paper is to estimate α_i^t, which is the label of segmentation for each pixel i in frame t. α has binary values. $\alpha_i^t = 1$ when the pixel belongs to the foreground moving object, $\alpha_i^t = 0$ when it is on the background. We set $\alpha = 0$ for all pixels initially.

In order to automatically compute a visually satisfying and perceptually correct foreground extraction result, our method iterates between two steps, i.e., the dense depth estimation and foreground labeling, until a stable bilayer segmentation is obtained. Table 1 gives an overview of our algorithm.

3. Dense Depth Estimation

We use the structure from motion (SFM) method proposed in [21] to recover the camera motion parameters from the given video sequence. For completeness we briefly summarize the algorithm as follows.

1 **Structure from Motion**:
1.1 Recover the camera motion parameters $\mathbf{C}$ and sparse 3D points $\mathbf{D}$.
2 **Dense Motion and Depth Estimation**:
2.1 Estimate dense motion $\mathbf{d}$ and occlusion o for each two consecutive frames.
2.2 Estimate depth map z^t and residual error map γ^t for each frame t.
3 **Bilayer Segmentation**:
3.1 Compute the *appearance and structure consistency maps*.
3.2 Solve α by minimizing (11).
4 Repeat steps 2 and 3 for k iterations.
5 Finally, α is refined by border matting [12].

Table 1. Overview of our framework.

We first detect and track feature points over the whole video sequence. Then, we select the superior tracks and key frames, and initialize the projective reconstruction from the reference triple frames. The projective reconstruction is upgraded to a metric framework at an appropriate moment through self-calibration. For each newly added frame, the new camera parameters and 3D points are initialized, and existing structure and motion are refined. Finally, the whole structure and motion are refined through bundle adjustment.

The output of the SFM estimation includes the recovered camera parameter set $\mathbf{C}$ and a sparse 3D point set $\mathbf{D}$ mapping to the feature points in the video frames. We denote the camera parameter as $\mathbf{C}^t = \{\mathbf{K}^t, \mathbf{R}^t, \mathbf{T}^t\}$ for frame t, where $\mathbf{K}^t$ is the intrinsic matrix, $\mathbf{R}^t$ is the rotation matrix, and $\mathbf{T}^t$ is the translation vector.

3.1. Motion Estimation

The motion of each pixel is computed on consecutive frames in the video. We use a displacement vector $\mathbf{d}^{t,t+1}(i) = (d_x^{t,t+1}(i), d_y^{t,t+1}(i))$ to model the motion of pixel i between neighboring two frames t and $t+1$. In order to handle occlusions, for each frame pair f^t and f^{t+1}, we define the occlusion label $\{o_i^{t,t+1} | o_i^{t,t+1} \in \{0,1\}\}$ for each pixel i. If one pixel is occluded when mapping from frame t to frame $t+1$, $o^{t,t+1}$ is set to 1.

We define the following objective function to solve the dense displacement map:

$$\arg\min_{\mathbf{d},o} \sum_{t=1}^{n-1} (E^{t,t+1}(\mathbf{d},o) + E^{t+1,t}(\mathbf{d},o)), \qquad (1)$$

where $E^{t,t+1}(\mathbf{d},o)$ and $E^{t+1,t}(\mathbf{d},o)$ are the bidirectional energy terms representing the mapping from frame t to frame $t+1$ and mapping from frame $t+1$ to frame t respectively. Since they are similarly defined, we only give the definition of $E^{t,t+1}(\mathbf{d},o)$ as follows,

$$\begin{aligned} E^{t,t+1}(\mathbf{d},o) = \sum_{i \in f_t} [m^{t,t+1}(i) + \sum_{j \in N(i)} s^{t,t+1}(i,j)] \\ + \mathcal{D}^{t,t+1}(\mathbf{D}), \end{aligned} \qquad (2)$$

where $N(\cdot)$ denotes the set of neighborhood. The energy function has three components: (i) the data matching term $m(i)$, (ii) the smoothness term $s(i,j)$ which consists of the spatial smoothness of motion and the visibility consistency, and (iii) a prior from the recovered 3D points.

3.2. The Energy Function

Data matching term $m(i)$ is defined on the color constancy constraint between the matched pixels and is given by

$$m^{t,t+1}(i) = \begin{cases} \rho_d^{t,t+1}(i), & o_i^{t,t+1} = 0, \alpha_i^t = \alpha_{i'}^{t+1} \\ \min\{\rho_d^{t,t+1}(i), \eta_o\}, & o_i^{t,t+1} = 0, \alpha_i^t \neq \alpha_{i'}^{t+1} \\ \eta_o, & o_i^{t,t+1} = 1 \end{cases}$$

where i' in f^{t+1} is the matched pixel of i in frame t. η_o is a penalty, preventing all pixels from being labeled as occlusion, which is defined similarly as the one in [14]. $\rho_d^{t,t+1}(i)$ is a differentiable robust function:

$$\rho_d^{t,t+1}(i) = \frac{||f^t(i) - f^{t+1}(i')||^2}{\eta_d + ||f^t(i) - f^{t+1}(i')||^2}.$$

If $o_i^{t,t+1} = 0$ and $\alpha_i^t \neq \alpha_{i'}^{t+1}$, there should ideally exist occlusion. However, in our optimization process, due to the use of discrete image space and the possible estimation errors, the bilayer separation is not always accurate. The matching cost is thereby defined as $\min\{\rho_d^{t,t+1}(i), \eta_o\}$ to constrain the cost. Using optical flow, the color difference between $f^t(i)$ and $f^{t+1}(i')$ can be further written as

$$||f^t(i) - f^{t+1}(i')||^2 \approx ||f_x^t(i) \cdot d_x^{t,t+1}(i) + f_y^t(i) \cdot d_y^{t,t+1}(i) + f_t^t(i)||^2,$$

where f_x^t, f_y^t and f_t^t are image gradients in x, y and t directions respectively. The continuity of the above function is important in computing the first order derivative

$$\frac{\partial(f^t(i) - f^{t+1}(i'))}{\partial \mathbf{d}} \approx (f_x^t(i), f_y^t(i))^\top,$$

which makes it possible to apply a nonlinear continuous optimization, e.g., the steepest descent method, to estimate $\mathbf{d}$.

Smoothness term $s(i, j)$ encourages the smoothness of the motion and occlusion, and is defined as

$$\begin{aligned} s^{t,t+1}(i,j) = \beta_s \rho_s^{t,t+1}(i,j) + \beta_o |o_i^{t,t+1} - o_j^{t,t+1}| \\ + \beta_w |o_i^{t,t+1} - W_i^{t,t+1}|, \end{aligned} \tag{3}$$

where ρ_s and $|o_i^{t,t+1} - o_j^{t,t+1}|$ are the spatial smoothness constraints for the displacement and occlusion in each frame. ρ_s is a robust function, given by

$$\rho_s^{t,t+1}(i,j) = \begin{cases} \min\{||\mathbf{d}^{t,t+1}(i) - \mathbf{d}^{t,t+1}(j)||^2, \eta_s\}, & \alpha_i^t = \alpha_j^{t+1} \\ 0, & \alpha_i^t \neq \alpha_j^{t+1} \end{cases}$$

which implies if two neighboring pixels belong to different layers after segmentation, the spatial smoothness does not need to be preserved. η_s controls the upper bound of the cost. $W^{t,t+1}(i) \in \{0, 1\}$ is a binary value, indicating whether or not there exists one or more pixels i in f^t that can be matched from f^{t+1} according to the displacement value $\mathbf{d}^{t+1,t}$ [14]. The value of $W^{t,t+1}(i)$ is set to 1 if there is no corresponding pixel in f^{t+1} for $f^t(i)$.

Prior $\mathcal{D}$ imposes constraints with the recovered sparse 3D points $\mathbf{D}$ from our SFM estimation. For the 3D point $X \in \mathbf{D}$, its projections in f^t and f^{t+1} are denoted as $\mathbf{u}_X^t$ and $\mathbf{u}_X^{t+1}$ respectively where $\mathbf{u}_X^t$ can be computed by

$$\mathbf{u}_X^t = \mathbf{K}^t(\mathbf{R}^{\mathbf{t}} X + \mathbf{T}^t),$$

with the estimated camera parameters $\mathbf{K}^t$, $\mathbf{R}^{\mathbf{t}}$, and $\mathbf{T}^t$ from the SFM step. These pixels should be matched and be taken as *anchor points* in the optical flow estimation

$$\mathcal{D}^{t,t+1} = \beta_{\mathcal{D}} \sum_{X \in \mathbf{D}} \sum_{f_t, f_{t+1} \in \varphi(X)} ||\mathbf{d}^{t,t+1}(\mathbf{u}_X^t) - (\mathbf{u}_X^{t+1} - \mathbf{u}_X^t)||^2, \tag{4}$$

where $\varphi(X)$ is the frame set in which X has corresponding image feature points. The weight $\beta_{\mathcal{D}}$ is set to a large value.

3.3. Solving the Energy Function

Combining the definition of different energy terms, we solve for a dense displacement map with the consideration of the occlusion bi-directionally.

The occlusion o is initially set to all zeros. With the recovered 3D point set $\mathbf{D}$, we are able to determine the displacement of the sparse anchor points which correspond to the 3D points in $\mathbf{D}$. The motions of other pixels are initialized using our motion interpolation. Specifically, in each frame, we produce a 2-D triangulation of the sparse anchor points. Then the motion vectors for pixels inside each triangle are initialized using triangular interpolation. Our motion optimization algorithm alternates between the following two steps:

1. Fix o, and estimate $\mathbf{d}$ by minimizing (1).

2. Fix $\mathbf{d}$, and estimate o by minimizing (1). Since the occlusion o has binary values, we use graph cut [3] to compute it.

In step 1, nonlinear continuous optimization methods, e.g., the steepest descent algorithm, can be used to estimate $\mathbf{d}$. However, it requires a good start point and is easily stuck in a local minimum. To overcome this problem, we propose a *continuous-discrete optimization* process.

We first apply the steepest descent algorithm to estimate a displacement map for each frame pair. In this step, the result may be only in a local minimum point given the high dimension of the solution space. In order to pull the result out of a local minimum point, we apply scalar quantization on the displacement $\mathbf{d}$ in x and y directions in the range of $[d_x - \eta, d_x + \eta]$ and $[d_y - \eta, d_y + \eta]$ respectively, where η is a constant value and is set to 5 in our experiments. Then, in the discrete space, loopy belief propagation [16] is applied to compute a better solution $\mathbf{d}'$. The continuous and discrete optimizations alternate and rapidly converge in our experiments.

3.4. Depth Estimation and Geometric Constraint

Once the dense motion vectors are computed, we link each pixel forward and backward in the neighboring frames according to the pixel displacement. This process eventually form dense motion *tracks*. Assuming that the estimated optical flow is not always accurate and the errors may be accumulated in constructing the tracks, we break a link between the connected pixels $f^t(i)$ and $f^{t+1}(i')$ in a track if one of the following happens: 1) i' or i is labeled “occluded”. 2) The optical flow consistency error

$$e_{flow}^{t,t+1}(i) = ||\mathbf{d}^{t,t+1}(i) + \mathbf{d}^{t+1,t}(i')|| \tag{5}$$

is larger than a threshold (2 pixels in our experiments). After the above process, the lengths of all tracks are limited to no more than N frames (30 in our experiments).

For a track p expanding the frames from f^l to f^r, according to the definition of motion vectors, the pixels along track p in different frames should correspond to a same 3D point X_p in the scene. Denoting the pixel in track p in frame t as $\mathbf{x}_p^t$, ideally, we should have

$$\mathbf{x}_p^t = \mathbf{K}^t(\mathbf{R}^t X_p + \mathbf{T}^t),$$

where $\mathbf{K}$, $\mathbf{R}$, and $\mathbf{T}$ are the estimated camera parameters. In real examples, the above equation does not hold and there always exist residual errors if we compute $||\mathbf{x}_p^t - \mathbf{K}^t(\mathbf{R}^t X_p + \mathbf{T}^t)||$. Therefore, we estimate X_p by minimizing the root mean squared error (RMSE)

$$\arg\min_{X_p} \sqrt{\frac{1}{r-l+1}\sum_{t=l}^{r} ||\mathbf{x}_p^t - \mathbf{K}^t(\mathbf{R}^t X_p + \mathbf{T}^t)||^2}. \tag{6}$$

In our approach, the method of solving Equation (6) is similar to those in [11]. After obtaining a set of X's, a depth map z^t for each frame t, can be computed by storing the depth value z_p^t in $[x_p^t, y_p^t, z_p^t] = \mathbf{R^t} X_p + \mathbf{T}^t$. In the meantime, we record the RMSE for all pixels in frame t using a residual error map γ^t.

If one pixel maps to a 3D point in the background, its residual error should be small to satisfy the *multiple-view geometry*. Therefore, if the residual error for one pixel is large, it is quite possible that this pixel maps to the foreground since it does not satisfy the *geometric constraint*.

4. Moving Object Extraction

Although the residual error map γ^t and the depth map z^t contain important information to detect the moving object, they are still insufficient to correctly identify the foreground pixels due to the following reasons.

First, the residual error and depth rely greatly on the accuracy of motion estimation. Their values are not reliable on the moving object boundary, as shown in Figure 1 (a).

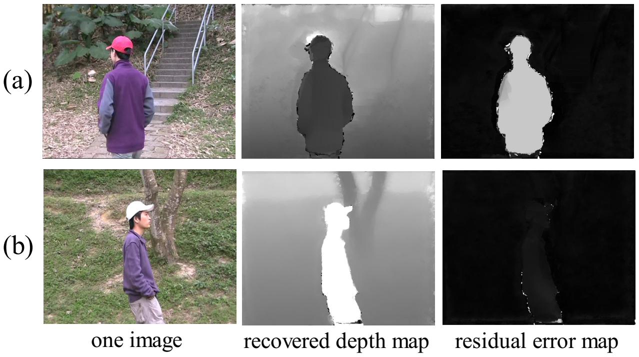


Figure 1. The problems of geometric constraint. (a) The residual errors near object boundary are noisy. (b) The residual errors of moving object are small, and the estimated depth values of moving object are very large, which contradict the ground truth.

Second, it is possible that the residual error on moving object is very small, which satisfies the geometric constraint. For instance, if in a video capturing, the camera undergoes the same motion as the foreground object in order to keep it in center of all frames, the computed residual errors of the pixels on moving object may be very small. This makes these pixels to be identified as the static background. It is interesting to note here that using the depth value also does not help to correct the errors in this situation. According to the multiple-view geometry, the recovered depth values of the moving object pixels are much larger than those of the true background pixels, as shown in Figure 1 (b). So even using the depth information, the pixels in the moving object will still be labeled as background!

According to the fact that the near objects occludes the far ones, we propose a method based on *appearance and structure consistency constraint* in 3D warping, which can appropriately address all these problems.

4.1. Appearance and Structure Consistency

Since the depth map is computed, 3D warping techniques [9] can be used to render new views by projecting the pixels in one frame to their 3D locations and re-projecting 3D points onto other frames. In our method, for frame t, we select its neighboring $2l$ frames, i.e. $\{f^{t-l}, ..., f^{t+l}\}$. Then we warp these frames to f^t using depth information. The pixels whose residual error is larger than a threshold (3.0 pixels in our experiments) are quite likely on the moving object. So we exclude them in the warping. The image warped from $f^{t'}$ to f^t is denoted as $\hat{f}^{t,t'}$. One illustration is shown in Figure 2. The red pixels are those receiving no projection during the warping.

Due to the accumulation error, the warped point may deviate from its correct position. We thus apply the following algorithm to locally search the best match using windows. The appearance error of pixel i with respect to f^t and $\hat{f}^{t,t'}$

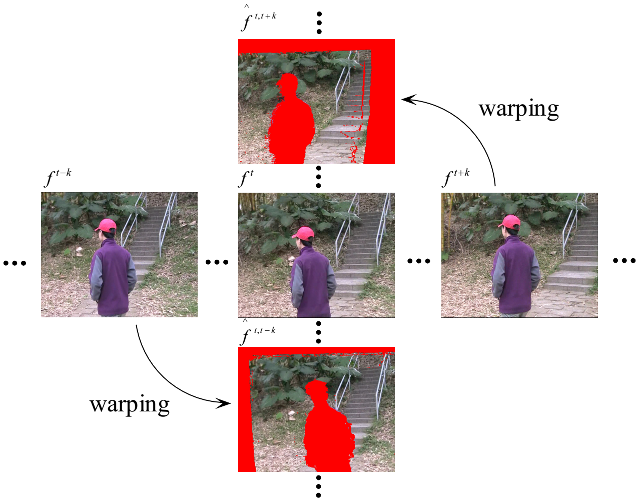


Figure 2. 3D warping. The neighboring frames of f^t are warped to f^t. The red pixels are those receiving no projection during the warping due to the large residual error or occlusion.

is given by

$$\mathcal{A}^{t,t'}(i) = \frac{1}{|W|} \min_j \sum_{k \in W} ||f^t(i+k) - \hat{f}^{t,t'}(i+j+k)||^2, \quad (7)$$

where W is a window, and $j = (d_x, d_y)$ is the searching index. It is illustrated in Figure 3. In our experiments, the size of W is set to 7×7, and the values of d_x and d_y are in $[-7, 7]$.

We also construct the warped depth maps $\hat{z}^{t,t'}$ similar to the warped frames. A structure error measurement is proposed to search local best match defined on the depth map. The structure error of pixel i with respect to f^t and $\hat{f}^{t,t'}$ is defined by

$$\mathcal{S}^{t,t'}(i) = \frac{1}{|W|} \min_j \sum_{k \in W} ||\frac{1}{z^t(i+k)} - \frac{1}{\hat{z}^{t,t'}(i+j+k)}||^2, \quad (8)$$

where z^t is the recovered depth map in frame f^t.

After computing (7) and (8), each pixel i in frame t has several appearance and structure error measurements. They can be used to basically represent the probability that one pixel is in foreground or background. For instance, if the residual error $\mathcal{A}^{t,t'}(i)$ is large, pixel i has high chance to be in the moving foreground. For each pixel, we apply the median filter to all $\mathcal{A}^{t,t'}(i)$ and $\mathcal{S}^{t,t'}(i)$ where $t' \in \{t-l, ..., t+l\}$, and compute the median values

$$\overline{\mathcal{A}}^t(i) = \text{median}\{\mathcal{A}^{t,t-l}(i), ..., \mathcal{A}^{t,t+l}(i)\},$$

$$\overline{\mathcal{S}}^t(i) = \text{median}\{\mathcal{S}^{t,t-l}(i), ..., \mathcal{S}^{t,t+l}(i)\}.$$

The appearance consistency term is defined as

$$\mathcal{C}^t_{\mathcal{A}}(i) = e^{-\frac{\overline{\mathcal{A}}^t(i)}{2\delta^2_{\mathcal{A}}}},$$

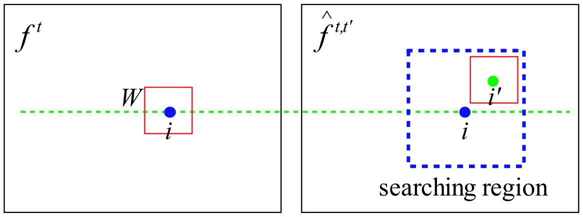


Figure 3. Locally searching the best match using windows. For pixel i in f^t, its best matching point in $\hat{f}^{t,t'}$ is i', which deviates from the true position. The red solid rectangle is the matching windows W, and the blue dash window shows the searching region.

where $\delta_{\mathcal{A}}$ is the standard deviation. Since $\overline{\mathcal{S}}^t$ and γ^t are defined on depth, we combine them in defining the structure consistency term

$$\mathcal{C}^t_{\mathcal{S}}(i) = \sqrt{e^{-\frac{\overline{\mathcal{S}}^t(i)}{2\delta^2_{\mathcal{S}}}} e^{-\frac{(\gamma^t(i))^2}{2\delta^2_{\gamma}}}},$$

where $\delta_{\mathcal{S}}$ and δ_γ are two standard deviations. In most of our experiments, $\delta_{\mathcal{A}} = 12$, $\delta_\gamma = 1.5$, and $\delta_{\mathcal{S}} = 0.05(z^{-1}_{\text{min}} - z^{-1}_{\text{max}})$. Here, $[z_{\text{min}}, z_{\text{max}}]$ is the depth range of the scene, which can be estimated using the recovered sparse 3D points $\mathbf{D}$.

Combining both the appearance and structure consistency terms, the data likelihood term is given by

$$\mathcal{L}^t(i) = \begin{cases} 1 - \mathcal{C}^t(i) & \alpha^t_i = 0 \\ \mathcal{C}^t(i) & \alpha^t_i = 1 \end{cases} \quad (9)$$

where $\mathcal{C}^t$ is the *appearance and structure consistency term*, a combination of $\mathcal{C}^t_{\mathcal{A}}$ and $\mathcal{C}^t_{\mathcal{S}}$, given by

$$\mathcal{C}^t(i) = w^t_{\mathcal{A}} \mathcal{C}^t_{\mathcal{A}}(i) + (1 - w^t_{\mathcal{A}}) \mathcal{C}^t_{\mathcal{S}}(i), \quad (10)$$

where $w_{\mathcal{A}}$ is a factor balancing the two terms.

With the definition of our likelihood, we give an analysis that the proposed model can address the problems described in Section 4. First, our model is not that sensitive to both the accumulation error and boundary artifacts using 3D warping, as demonstrated in Figure 4. Second, our model can faithfully represent the occlusion. For instance, if a moving object has the same motion with the camera, according to multiple-view geometry, its recovered depth value will be very large. We use frame t' close to the reference frame t, to produce $\hat{f}^{t,t'}$ by 3D warping. Since the moving object pixels have larger "depth" values than the background, the warped moving pixels in $\hat{f}^{t,t'}$ will be occluded by the background. Therefore, the moving pixels, no matter what their depth values are in frame t, will have large appearance and structure consistency error, and the likelihood term (9) can be used to correctly model the moving object.

4.2. The Model for Bilayer Segmentation

Only using the likelihood term (9) in segmentation does not necessarily preserve boundary smoothness. We introduce the following foreground/background separation energy function

$$E_B = \sum_{t=1}^{n}\sum_{i\in f^t}(\mathcal{L}^t(i) + \lambda_{\mathcal{T}}\mathcal{G}_{\mathcal{T}}^t(i) + \lambda_{\mathcal{S}}\sum_{j\in N(i)}\mathcal{G}_{\mathcal{S}}^t(i,j)). \tag{11}$$

There are two components: the data term $\mathcal{L}^t(i)$, and the smoothness term which consists of spatial smoothness $\mathcal{G}_{\mathcal{S}}^t(i,j)$ and temporal consistency $\mathcal{G}_{\mathcal{T}}^t(i)$. $\lambda_{\mathcal{S}}$ and $\lambda_{\mathcal{T}}$ are the relative weights. We use $\lambda_{\mathcal{S}} = \lambda_{\mathcal{T}} = 1$ in our experiments.

In our experiments, we found that in map $\mathcal{C}^t$ computed on all pixels in frame t, there is large contrast around the moving object boundary. So the spatial smoothness term should encourage the foreground boundary to lie on pixels with large contrast in map $\mathcal{C}^t$ as well as in image f^t. We compute the contrast on map $\mathcal{C}^t$ by

$$g_{\mathcal{AS}}^t(i,j) = G_\sigma \otimes ||\mathcal{C}^t(i) - \mathcal{C}^t(j)||,$$

where $\otimes$ is the operation of convolution and G_σ is a Gaussian smoothing filter with a characteristic width of σ.

We also attenuate the background contrast to encourage smoothness in the background. In computing the appearance consistency error for each pixel i in frame t, suppose $\overline{\mathcal{A}}^t(i)$ is found in $\hat{f}^{t,t'}$ (i.e. $\overline{\mathcal{A}}^t(i) = \mathcal{A}^{t,t'}(i)$), and its corresponding best matching point is i', we consider $\hat{f}^{t,t'}(i')$ as the background color for pixel i. Thus we estimate a background image for f^t, and employ the method proposed in [15] to attenuate the background contrast. The attenuated color contrast is denoted as $g_c^t(i,j)$.

Finally, we combine $g_c^t(i,j)$ and $g_{\mathcal{AS}}^t(i,j)$ to define the spatial smoothness term

$$\mathcal{G}_S(i,j) = \begin{cases} \exp(-\frac{g_{\mathcal{AS}}^t(i,j)\cdot g_c^t(i,j)}{2\sigma_s}) & \text{if } \alpha_i^t \neq \alpha_j^t \\ 0 & \text{if } \alpha_i^t = \alpha_j^t \end{cases} \tag{12}$$

where $\sigma_s = 0.04 \cdot 15$ in our experiments.

The temporal consistency term is bidirectional, given by

$$\mathcal{G}_T^t(i) = \mathcal{G}_T^{t,t+1}(i) + \mathcal{G}_T^{t,t-1}(i).$$

Let i' in f^{t+1} be the corresponding point to pixel i in f^t using motion estimation, the $\mathcal{G}_T^{t,t+1}(i)$ is defined as

$$\mathcal{G}_T^{t,t+1}(i) = \begin{cases} w_{flow}^{t,t+1}(i) & \text{if } \alpha_i^t \neq \alpha_j^t \\ 0 & \text{if } \alpha_i^t = \alpha_j^t \end{cases}$$

where $w_{flow}^{t,t+1}(i)$ measures the optical flow consistency error defined in (5) and color difference in motion estimation. It is given by

$$\begin{aligned} w_{flow}^{t,t+1}(i) &= \exp(-\frac{(e_{flow}^{t,t+1}(i))^2}{2\delta_{flow}^2})\cdot \\ &\quad \exp(-\frac{||f^t(i) - f^{t+1}(i')||^2}{2\delta_{color}^2}), \end{aligned} \tag{13}$$

η_d	η_o	η_s	β_s	β_o	β_w	$\beta_{\mathcal{D}}$	$\delta_{\mathcal{A}}$	δ_γ
400	0.5	4	0.1	0.21	0.6	100	12	1.5

$\delta_{\mathcal{S}}$	$w_{\mathcal{A}}$	$\lambda_{\mathcal{T}}$	$\lambda_{\mathcal{S}}$	σ_s	δ_{flow}	δ_{color}
$0.05(z_{\text{min}}^{-1} - z_{\text{max}}^{-1})$	0.5	1	1	0.6	1.0	10

Table 2. Parameter configuration in our experiments.

where $\delta_{flow} = 1.0$ and $\delta_{color} = 10$ in our experiments. $\mathcal{G}_T^{t+1,t}(i)$ is symmetrically defined.

4.3. Iterative Optimization

To reduce the complexity in computing the appearance and structure consistency map for each frame t, we only select 20-30 frames from its neighboring frames to perform 3D warping. After the appearance and structure consistency maps are computed, we apply graph cut method to compute α by minimizing $E_B(\alpha)$. If we take all frames in computing E_B, the process can be very time-consuming. It is also unnecessary since the temporal smoothness may not hold if two frames are not close in the video. In our method, we simultaneously solve 10 frames each time from the head to the tail in the video.

After estimating α, we further refine the motion parameters $\{\mathbf{d}, o\}$ described in Section 3 and use them to optimize α again. Two iterations in alternating the two steps are sufficient in our method. Finally, the binary foreground maps are refined using border matting [12]. Table 2 lists the parameter values used in our experiments.

5. Results

We validate our algorithm using several challenging video sequences taken by a hand-held camera. Table 3 lists some statistical information of the video sequences we use. Strong vibration can be observed in all the video sequences. Also, due to the deinterlacing process, there is color mixing around object boundary. These factors bring difficulties to perform accurate foreground separation.

Figure 4 shows that our method can successfully extract the moving object. The foreground can not be extracted only using depth maps and residual error maps due to the ambiguity in the geometric constraint. Figure 4 (b) and (c) show that the residual errors of moving object are very small and the recovered depth has large value in foreground, which contradicts the ground truth. Figure 4 (d) shows the appearance and structure consistency map defined in (10). The contrast map of appearance and structure consistency and the attenuated color contrast map are shown in Figure 4(e) and (f) respectively, using which we generate the spatial smooth term map shown in Figure 4(g). Figure 4(h) shows our binary segmentation result, and (i) shows our foreground extraction result refined by matting method.

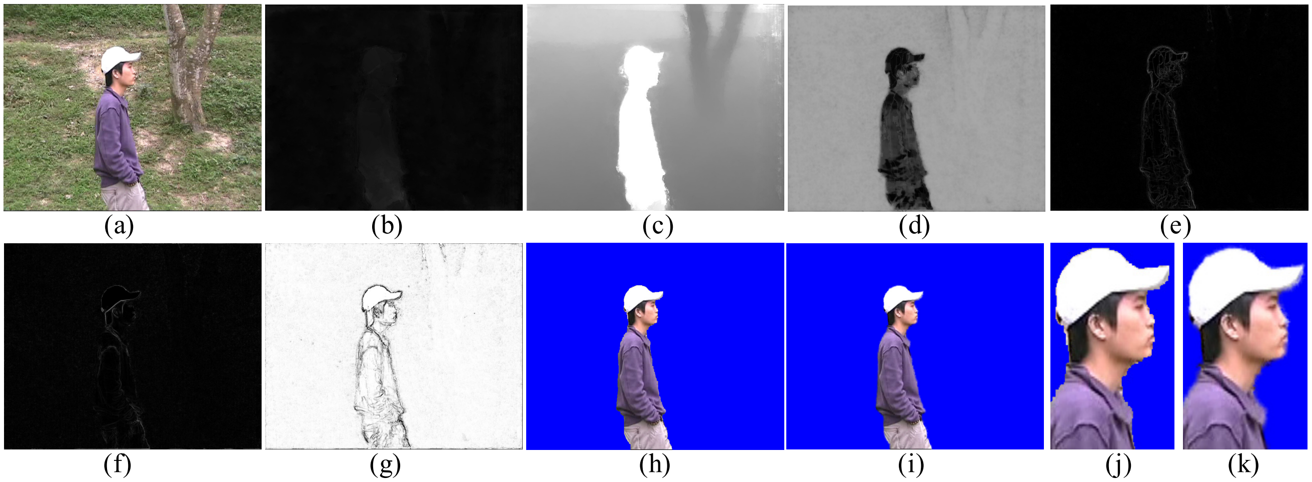


Figure 4. (a) One frame of Tree Sequence. (b) Residual error map. (c) Recovered depth map. (d) Appearance and structure consistency map defined in (10). (e) Contrast map of appearance and structure consistency. (f) Attenuated color contrast map. (g) The spatial smoothness term map defined in (12). (h) Foreground segmentation result. (i) Foreground extraction result after matting. (j) Magnified region of (h). (k) Magnified region of (i).

sequence	Tree (Fig. 4)	Stair (Fig 6(a))	Path (Fig 6(b))
frames	150	200	110
frames/sec	25	25	25

Table 3. Video lengths of the three tested sequences.

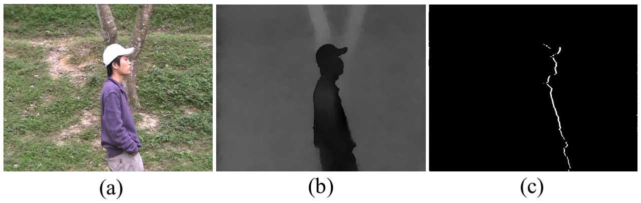


Figure 5. Optical flow. (a) One frame. (b) Optical flow map. (c) Occlusion map.

Figure 5 shows an estimated forward optical flow map with occlusion for the “Tree” sequence. In (b), the displacement for each pixel i is computed as $||\mathbf{d}(i)||$. Our recovered occlusion map in Figure 5(c) is close to the ground truth.

Figure 6 shows more results to demonstrate our moving object extraction system. Please refer to the supplementary video for the complete frames.

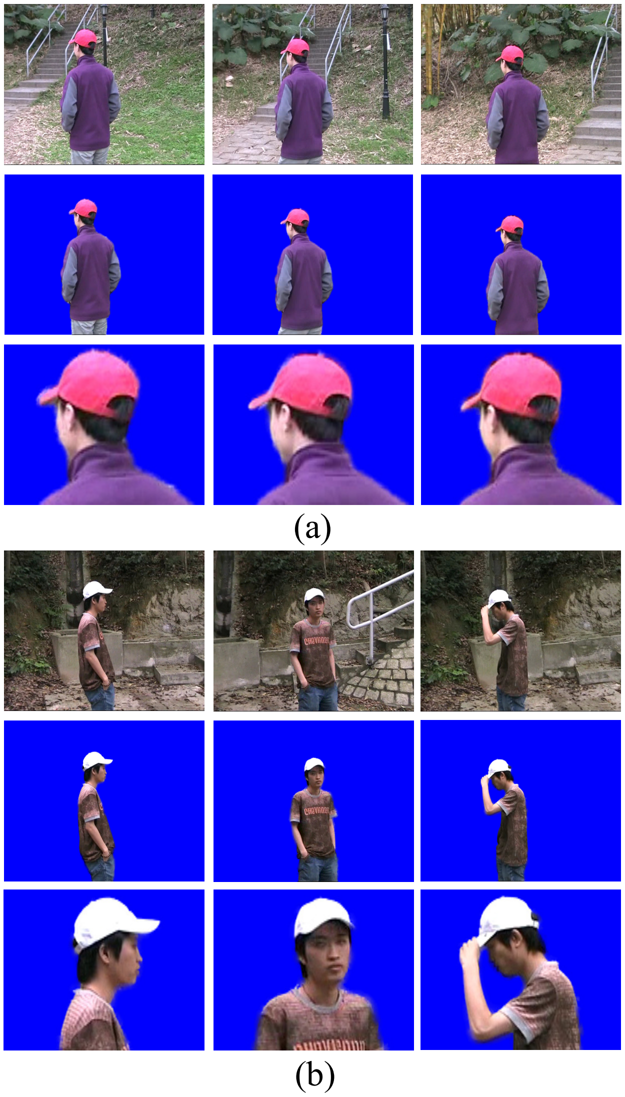
In our implementation, the total computation time is about 6 minutes for each frame averagely. The computation cost is mostly on the dense motion estimation in step 2.1 and the calculation of appearance and structure consistency maps in step 3.1.

6. Discussion and Conclusions

In this paper, we have proposed a complete bilayer separation system to accurately detect and extract the moving foreground object from a video sequence taken by a hand-held camera. Our method alternates between two major steps. In the first step, we estimate the camera motion parameters and the object motion fields which directly encode the occlusion information. We introduce the anchor points in prior to constrain the optical flow. The continuous-discrete optimization performs well in producing a globally optimal result. In the second step, we take the depth and motion information into the layer separation. It has been shown that the depth map and the geometric constraint has ambiguity in identifying the foreground object. So we introduce the appearance and structure consistency constraint to reliably detect the moving objects. Our final result is computed by optimization which combines a set of terms from motion, depth, and colors.

Our current system still has some limitations. First, if the background scenes do not have sufficient features and most regions are extremely textureless, the camera motion parameters and optical flow estimation will contain large errors, and the bilayer separation may not work well. This problem can be alleviated by incorporating segmentation into our motion estimation. Second, when foreground object contains very thin structures or small holes with respect to image size, incorrect separation may happen around these regions.

Acknowledgements. We would like to thank Fangming Liu, Liansheng Wang and Defeng Wang for their help during video capture. This work is supported by NSF of China (No.60633070), 973 program of China (No.2002CB312104), and a grant from the Research Grants Council of the Hong Kong Special Administrative Region (Project No. 412206). This work is affiliated with the Microsoft-CUHK Joint Laboratory and the Virtual Re-



(a)

(b)

Figure 6. More examples. (a) “Stair” example. (b) “Path” example. In each example, the upper row shows three frames selected from videos, the middle row shows our foreground extraction results, and the lower row shows the magnified views of extraction results.

ality, Visualization and Imaging Research Center at the Chinese University of Hong Kong.

References

[1] S. Ayer and H. S. Sawhney. Layered representation of motion video using robust maximum-likelihood estimation of mixture models and mdl encoding. In *ICCV*, pages 777–784, 1995. 1

[2] M. J. Black and P. Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding*, 63(1):75–104, 1996. 1

[3] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(11):1222–1239, 2001. 3

[4] A. Criminisi, G. Cross, A. Blake, and V. Kolmogorov. Bilayer segmentation of live video. In *CVPR (1)*, pages 53–60, 2006. 1

[5] A. M. Elgammal, D. Harwood, and L. S. Davis. Nonparametric model for background subtraction. In *ECCV (2)*, pages 751–767, 2000. 1

[6] S. Khan and M. Shah. Object based segmentation of video using color, motion and spatial information. In *CVPR (2)*, pages 746–751, 2001. 1

[7] V. Kolmogorov, A. Criminisi, A. Blake, G. Cross, and C. Rother. Bi-layer segmentation of binocular stereo video. In *CVPR - Volume 2*, pages 407–414, 2005. 1

[8] M. P. Kumar, P. H. S. Torr, and A. Zisserman. Learning layered motion segmentation of video. In *ICCV*, pages 33–40, 2005. 1

[9] W. R. Mark, L. McMillan, and G. Bishop. Post-rendering 3d warping. In *SI3D*, pages 7–16, 180, 1997. 4

[10] A. Monnet, A. Mittal, N. Paragios, and V. Ramesh. Background modeling and subtraction of dynamic scenes. In *ICCV*, pages 1305–1312, 2003. 1

[11] M. Pollefeys, L. J. V. Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, and R. Koch. Visual modeling with a hand-held camera. *International Journal of Computer Vision*, 59(3):207–232, 2004. 4

[12] C. Rother, V. Kolmogorov, and A. Blake. ”grabcut”: interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23(3):309–314, 2004. 2, 6

[13] Y. Sheikh and M. Shah. Bayesian object detection in dynamic scenes. In *CVPR (1)*, pages 74–79, 2005. 1

[14] J. Sun, Y. Li, and S. B. Kang. Symmetric stereo matching for occlusion handling. In *CVPR (2)*, pages 399–406, 2005. 3

[15] J. Sun, W. Zhang, X. Tang, and H.-Y. Shum. Background cut. In *ECCV (2)*, pages 628–641, 2006. 1, 6

[16] Y. Weiss. Belief propagation and revision in networks with loops. Technical report, Cambridge, MA, USA, 1997. 3

[17] Y. Weiss and E. H. Adelson. A unified mixture framework for motion segmentation: Incorporating spatial coherence and estimating the number of models. In *CVPR*, pages 321–326, 1996. 1

[18] J. Wills, S. Agarwal, and S. Belongie. What went where. In *CVPR (1)*, pages 37–44, 2003. 1

[19] C. R. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(7):780–785, 1997. 1

[20] P. Yin, A. Criminisi, J. Winn, and I. Essa. Tree-based classifiers for bilayer video segmentation. In *CVPR*, 2007. 1

[21] G. Zhang, X. Qin, W. Hua, T.-T. Wong, P.-A. Heng, and H. Bao. Robust metric reconstruction from challenging video sequences. In *CVPR*, 2007. 2