Consistent Depth Maps Recovery from a Trinocular Video Sequence

Wenzhuo Yang[1] Guofeng Zhang[1*] Hujun Bao[1*] Jiwon Kim[2] Ho Young Lee[2]
[1]State Key Lab of CAD&CG, Zhejiang University [2]Samsung Advanced Institute of Technology

Abstract

In this paper, we propose a novel dense depth recovery method for a trinocular video sequence. Specifically, we contribute a novel trinocular stereo matching model, which can effectively utilize the advantages of trinocular stereo images, and incorporate the visibility term with segmentation prior for robust depth estimate. In order to make the recovered depth maps more accurate and temporally consistent, we propose to first classify the pixels to static and dynamic ones, and then perform spatio-temporal depth optimization for them in different ways. Especially, we propose two motion models for handling dynamic pixels. The traditional bundle optimization model and our spatio-temporal optimization model are softly combined in a probabilistic way, so that the depths of both static and dynamic pixels can be effectively refined. Our automatic depth recovery approach is evaluated using a variety of challenging trinocular video sequences.

1. Introduction

3D reconstruction from images/videos has long been a standard problem in computer vision due to its wide applications. Especially, with the recent success of 3D movies (*e.g.* Avatar), there has been tremendous interest in 3D video generation. Although stereoscopic videos can be captured by a stereo camera, they only allow narrow multi-view display of 3D content since the stereoscopic views are generated from one viewing point. The users need to position themselves in the right locations for optimal visual experience. For achieving wide multi-view display of 3D content, accurate dense depth information is required.

For static scenes, we can use a single camera to capture multiple images or a video sequence to recover depth information by multi-view stereo techniques that have been widely studied in the past decades. Recent advances have moved the focus on the 3D reconstruction of moving objects, which typically need to use synchronized multiple cameras [19, 9]. However, most of existing methods assume the cameras are fixed and do not move, which seriously restricts the applications in practice.

In this paper, we propose a novel method to recover accurate and consistent depth maps from a trinocular stereo sequence. In our device configuration, the relative camera poses are constant, and the cameras can freely move together. Compared to binocular stereo camera, trinocular stereo camera has significant advantages in handling occlusions, because for stereo images, there are always occluded pixels whose depths can not be inferred by matching. In contrast, in our case, since the baselines between cameras are semi-wide, each pixel in the center image generally has at least one correspondence in the left or right image. Compared to the multi-camera array [19], using a trinocular stereo camera is more flexible and the cost is much lower.

2. Related Work

Two-view stereo matching has been extensively studied during the past decades [12, 14, 4]. However, two-view stereo matching has inherent difficulty in handling occlusions due to information missing. Although some methods have been proposed to incorporate smoothness constraint or segmentation prior to alleviate this problem, it is still very challenging to recover high-quality depth maps from complex natural images with strong occlusions. In contrast, multi-view stereo matching has more advantages in handling occlusions and image noise, so that the recovered depth maps could be more accurate. In recent years, multi-view stereo has achieved significant advances. Furukawa *et al.* [7] proposed a seed growing method to reconstruct accurate surface model from a set of 3D points. Vu *et al.* [15] reconstructed high quality large-scale scenes in a mesh optimization framework. Although multi-view stereo matching has been extensively studied, most multi-view stereo methods are limited to static scene reconstruction and the problem of how to accurately and consistently recover dense depth information of dynamic scenes has not been sufficiently addressed.

The typical solution for depth recovery of dynamic scenes is to use multiple synchronized cameras. Depths of dynamic objects are recovered by stereo matching on multiple synchronous camera frames and temporal smooth-

[*] Corresponding Authors: {zhangguofeng,bao}@cad.zju.edu.cn

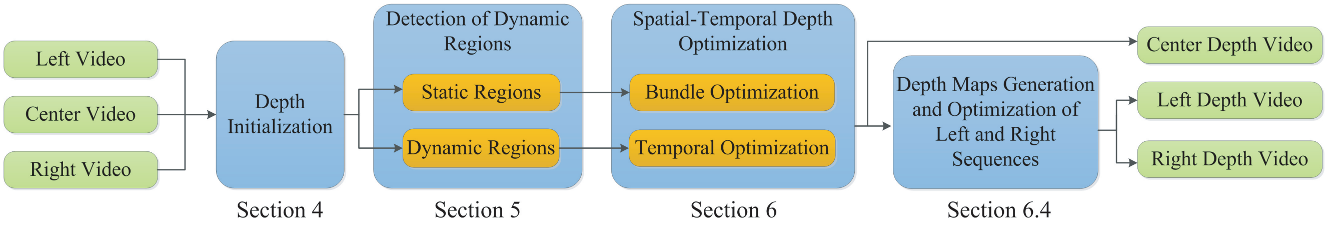


Figure 1. Framework overview.

ness constraint based on optical flow is added for depth refinement. In [9], Larsen *et al*. incorporate motion estimation and modify the BP algorithm to generate temporally consistent reconstruction results from multiple camera views. Some other methods [8, 10] extract moving objects from known or stationary background and incorporate segmentation to depth refinement, which require more than three cameras for robust depth estimate. Recently, Zhang *et al*. [17] proposed a robust bilayer segmentation method which can accurately detect and extract the moving object from a video sequence taken by a moving camera. However, this method requires a manual preprocessing for foreground color distribution learning.

Several algorithms [18, 2, 16] for temporally consistent depth maps estimation have been developed. Zhang *et al*. [18] proposed a powerful bundle optimization framework to recover spatio-temporally consistent depth maps for static scenes with a single moving camera. Recently, Yang *et al*. [16] extended this method to handle dynamic scene. However, their method only used very few neighboring frames for depth optimization, and is rather sensitive to optical flow errors and occlusions. As a result, their method required a camera array with relative small baselines for robust depth estimate. To overcome disadvantages of previous methods, we propose a novel method for high-quality dense depth recovery from a trinocular video sequence, which can generate accurate and consistent depth maps for both static and dynamic regions automatically.

3. Framework Overview

Figure 1 gives the flow chart of our system. The camera parameters of the input trinocular stereo sequence can be estimated by SfM technique [11]. Since our paper focuses on the dense depth recovery problem, we assume the camera parameters of the input trinocular stereo sequence are already known. Our objective is to estimate the depth maps for each frame. We first initialize the depth maps for each frame in the center sequence. In order to effectively utilize the advantage of trinocular images to handle occlusions, we propose a new trinocular stereo matching model which can reliably estimate the depth maps of trinocular images in the same time instance. Considering the number of views may significantly influence the robustness of stereo matching, we need to involve temporally neighboring frames to improve the depth accuracy as much as possible. Based on the initialized depth maps, we contribute a novel moving object extraction method, which can classify the pixels to static and dynamic ones so that their spatio-temporal refinement can be performed in more effective ways. Finally, traditional bundle optimization model and our spatio-temporal optimization model are combined to refine the estimated depth maps. After depth generation of the center sequence, the ones of the left/right sequence are also computed. In our paper, disparity is defined as “inverse depth”, the same as in [18]. For simplicity, the terms “depth” and “disparity” are used interchangeably in the following sections.

4. Disparity Initialization

In this step, each depth map in the center sequence is estimated independently. Due to the semi-wide baseline configuration (the baseline is generally $8 \sim 20$ cm), the occlusion in trinocular stereo images can be better handled since each pixel in the center view generally has at least one correspondence in left or right frame. Therefore, we propose to take this advantage to define trinocular stereo matching model.

4.1. Trinocular Stereo Matching Model

Given a trinocular frame, we denote the left, center and right views as I_L, I_M and I_R, respectively, and the occlusion of the center view as $O = \{O_L, O_R\}$. $O_L(\mathbf{x})$ indicates whether a pixel $\mathbf{x}$ in the center view is occluded in the left view. If $\mathbf{x}$ is occluded, $O_L(\mathbf{x}) = 1$, otherwise $O_L(\mathbf{x}) = 0$. $O_R(\mathbf{x})$ is defined in a similar way. We denote $I_i(\mathbf{x})$ as the color of pixel $\mathbf{x}$ in the view I_i.

Our goal is to estimate the disparity map D of the center view. We denote the disparity range as $[d_{\min}, d_{\max}]$ and equally quantize the disparity into $m+1$ levels, where the k^{th} level $d_k = (m-k)/m \cdot d_{\min} + k/m \cdot d_{\max}$, $k = 0, ..., m$. Similar to the symmetric stereo matching model [14], we minimize the following energy function to solve disparity map D and occlusion O:

$$E(D, O; I) = E_d(D, O; I) + E_s(D, O; I) + E_v(D, O; I) \tag{1}$$

where $E_d(D, O; I)$ is the data term, $E_s(D, O; I)$ is the smoothness term and $E_v(D, O; I)$ is the visibility term.

4.1.1 Data term

The data term measures how well the hypothesized disparity D and occlusion O fit the given sequence I. For the center view, the data term $E_d(D, O; I)$ is defined as:

$$\begin{aligned} E_d(D,O;I) = \sum_{\mathbf{x}} \frac{1}{Z_n(\mathbf{x})} \Big(& O_L(\mathbf{x}) O_R(\mathbf{x}) \eta + \\ & (1 - O_L(\mathbf{x}))(1 + O_R(\mathbf{x})) \rho(\mathbf{x}, D(\mathbf{x}); I_L, I_M) + \\ & (1 - O_R(\mathbf{x}))(1 + O_L(\mathbf{x})) \rho(\mathbf{x}, D(\mathbf{x}); I_R, I_M)) \Big), \end{aligned} \tag{2}$$

where $\rho(\mathbf{x}, d; I_1, I_2)$ is the robust matching cost function of pixel $\mathbf{x}$ and disparity d given observation I_1 and I_2, which is the same as $\rho_d(\mathbf{x})$ in [14]. $Z_n(\mathbf{x})$ is the normalization factor, making the largest cost of each pixel $\mathbf{x}$ be 1. The cost η is to prevent the whole image from being labeled as occlusion. According to occlusion status, $E_d(D, O; I)$ is set to the binocular stereo data cost of the center and left images, or the data cost of the center and right images, or their average.

4.1.2 Smoothness term

The smoothness term $E_s(D, O; I)$ encodes piecewise smoothness on disparity D, which is defined as:

$$E_s(D,O;I) = \sum_{\mathbf{x}} \sum_{\mathbf{y} \in N(\mathbf{x})} \lambda(\mathbf{x}, \mathbf{y}) \rho_s(D(\mathbf{x}), D(\mathbf{y})), \tag{3}$$

where $N(\mathbf{x})$ is the set of adjacent pixels for pixel $\mathbf{x}$, and $\lambda(\mathbf{x}, \mathbf{y})$ encourages the disparity discontinuity to be coincident with abrupt intensity/color change, which is the same as Equation (4) in [18]. $\rho(\cdot)$ is a robust function defined as $\rho_s(D(\mathbf{x}), D(\mathbf{y})) = \min(|D(\mathbf{x}) - D(\mathbf{y})|, T)$. Here, T controls the upper limit of the cost.

4.1.3 Visibility term

The visibility term $E_v(D, O; I)$ is the same with Equation (6) in [14]. Taking $E_v(D, O_L; I_L, I_M)$ as an example, $E_v(D, O_L; I_L, I_M)$ is defined as:

$$\begin{aligned} E_v(D, O_L; I_L, I_M) = & \sum_{\mathbf{x}} \beta_\omega |O_L(\mathbf{x}) - W_L(\mathbf{x}; D)| + \\ & \sum_{\mathbf{x}} \sum_{\mathbf{y} \in N(\mathbf{x})} \beta_o |O_L(\mathbf{x}) - O_L(\mathbf{y})|, \end{aligned} \tag{4}$$

where W_L is a binary map defined on the center view. For computational efficiency, the asymmetric occlusion model [4] is applied, so that W_L can be computed as follows:

$$W_L(\mathbf{x}; D) = \begin{cases} 1, & \text{if} \begin{array}{l} \exists \mathbf{y} \in I_M, P(\mathbf{x}, D(\mathbf{x})) = P(\mathbf{y}, D(\mathbf{y})) \\ \wedge D(\mathbf{x}) < D(\mathbf{y}) \wedge f(\mathbf{x}) \neq f(\mathbf{y}) \end{array} \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where $\mathbf{x}$ is a pixel in the center view, and $P(\mathbf{x}, d)$ is its corresponding pixel in the left view given the disparity value d. $f(\mathbf{x})$ is the surface that $\mathbf{x}$ belongs to, which can be determined by the mean-shift segments (i.e. if $\mathbf{x}$ and $\mathbf{y}$ are in the same segment, $f(\mathbf{x}) = f(\mathbf{y})$). Therefore, if the disparity map and the segments of a frame are known, W_L can be easily computed by using Equation (5). The visibility term $E_v(D, O_R; I_R, I_M)$ and the binary map $W_R(\mathbf{x}; D)$ are defined in a similar way.

4.2. Iterative optimization

We apply an iterative optimization algorithm to minimize (1). Similar to [14], the optimization process iterates between two steps : 1) estimate occlusion given disparity, and 2) estimate disparity given occlusion. Generally, two iterations are sufficient in our experiments. Please refer to our supplementary material [1] for more details.

The second iteration step contains the following three steps. First, BP algorithm [6] is applied to minimize (1) to generate the disparity map directly. Then the estimated disparity map is refined by fitting disparity segments to a set of 3D planes, using the same plane fitting technique introduced in [18]. However, the refined disparity map may contain errors if the segmentation information is imperfect. Similar to [4], in order to reduce this problem, we propose to fuse different disparity maps estimated under a variety of segmentation results generated by different segmentation parameters. Specifically, we choose a set of different mean-shift parameters to generate k disparity maps $\{D_1, D_2, \cdots, D_k\}$ which form the disparity candidate set $\hat{D}$. In addition, we also compute the average value (i.e. $D_{k+1}(\mathbf{x}) = \sum_{i=1}^{k} D_i(\mathbf{x})/k$), and add it into $\hat{D}$. Finally, with these disparity candidates, the disparity map is refined by minimizing (1) (occlusion variables are fixed). In our system, the candidate number is generally set to 4 or 5.

We use the Middlebury stereo data [13] to quantitatively evaluate our method. Figures 2(b)-(d) show the estimated disparity maps of "Teddy" example in different procedures, and Figure 2(e) shows the estimated occlusion map. In the first procedure, the estimated depth map contains lots of errors as shown in Figure 2(b). These errors are dramatically reduced after the following procedures. The error percentages after each procedure are listed in Table 1. Please refer to our supplementary material for inspecting the intermediate results of "Cones" example.

5. Detection of Dynamic Regions

Because the 3D position of a dynamic point may vary among different frames but a static 3D point remains constant, their spatio-temporal refinement should be performed

[1] The supplementary material and video can be found from the website: http://www.cad.zju.edu.cn/home/gfzhang/

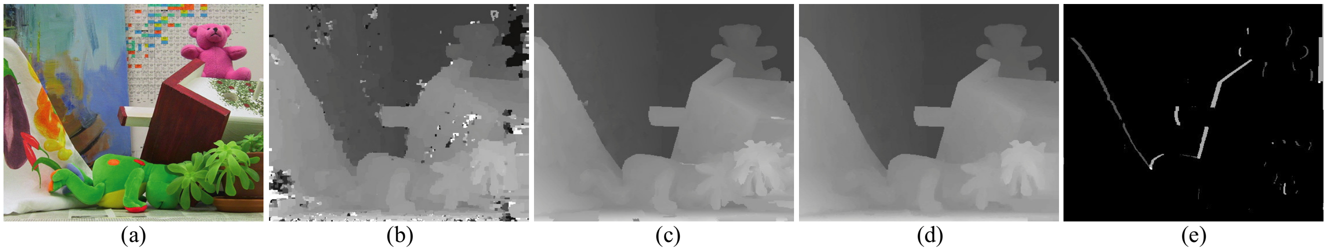


Figure 2. Intermediate results of trinocular stereo matching. (a) One image of "Teddy" data. (b) Procedure 1: the estimated disparity map without plane fitting. (c) Procedure 2: the estimated disparity map by plane fitting with one segmentation result. (d) Procedure 3: the final refined disparity map after fusion refinement. (e) The estimated occlusion map. White pixels are occluded in the right image and gray pixels are occluded in the left image.

Table 1. The percentages of error pixels (disparity errors $>$ 1 pixel).

	Teddy			Cones		
Procedure	1	2	3	1	2	3
Error	17.08	6.84	5.21	9.99	6.54	5.84

in different ways, so that the separation of dynamic and static regions is needed. In this section, we will discuss our bilayer segmentation method.

5.1. Optical Flow Estimation

We use the method proposed in [3] to estimate the optical flow maps between consecutive frames. The optical flow from frame k to $k+n$ can be computed as $U_{k\to k+n} = \sum_{i=k}^{k+n-1} U_{i\to i+1}$, where $U_{i\to i+1}$ is the optical flow from frame i to $i+1$, and the reverse optical flow from frame $k+n$ to k is defined in a similar way.

Since optical flow estimation is not very reliable, accuracy measurement with outlier detection is quite necessary. For pixel $\mathbf{x}$ in frame k, its correspondence $\mathbf{y}$ in frame $k+n$ can be found by $\mathbf{y} = \mathbf{x} + U_{k\to k+n}(\mathbf{x})$. The corresponding point $\mathbf{x}'$ of $\mathbf{y}$, which is in frame k, can be found by using $\mathbf{x}' = \mathbf{y} + U_{k+n\to k}(\mathbf{y})$. If the optical flow is accurate, $\mathbf{x}$ and $\mathbf{x}'$ should be the same point. However, due to estimation errors, $\mathbf{x}$ and $\mathbf{x}'$ may be two different points. Therefore, the optical flow reliability at pixel $\mathbf{x}$ can be measured by:

$$P_{opt}(\mathbf{x}) = \exp(-\frac{||\mathbf{x} - \mathbf{x}'||_2}{\sigma}), \tag{6}$$

where σ is usually set to 2 in our experiments.

5.2. Moving Object Detection

Similar to [17], before moving object detection, we need to estimate the static background information first. For each reference frame k, the color images and depth maps of its neighboring frames can be warped to frame k with the estimated depth maps. So each pixel generally receive multiple projections. We use a median filter to reliably estimate the background color and depth. Then, we define the following three kinds of measures:

1. **Color difference measure.** It measures the color difference between the reference image and the estimated background image:

$$\mathcal{D}_c(\mathbf{x}) = \frac{1}{Z_c}\frac{||I(\mathbf{x}) - I_B(\mathbf{x})||_2}{||I(\mathbf{x}) - I_B(\mathbf{x})||_2 + \epsilon_c}, \tag{7}$$

where ϵ_c controls the shape of this function and $\mathcal{D}_c(\mathbf{x})$ is normalized to $[0,1]$ by factor Z_c.

2. **Disparity difference measure.** It measures the difference between the initialized disparity map and the estimated background disparity map:

$$\mathcal{D}_d(\mathbf{x}) = \frac{1}{Z_d}\min\{|D(\mathbf{x}) - D_B(\mathbf{x})|, \epsilon_d\}, \tag{8}$$

where D_B is the estimated background disparity map, and ϵ_d controls the upper limit of $|D(\mathbf{x}) - D_B(\mathbf{x})|$. $\mathcal{D}_d(\mathbf{x})$ is normalized to $[0,1]$ by factor Z_d.

3. **Motion-depth difference.** It measures the consistency of the estimated depth and the motion (or optical flow) information, defined as $\mathcal{D}_o(\mathbf{x}) = \mathcal{D}'_o(\mathbf{x})/(\mathcal{D}'_o(\mathbf{x})+\epsilon_o)$, where ϵ_o is a parameter and $\mathcal{D}'_o(\mathbf{x})$ in frame k is defined as:

$$\mathcal{D}'_o(\mathbf{x}) = \frac{1}{W}\sum_{i=-n}^{n} P_{opt}^{k\to k+i}(\mathbf{x}) \cdot ||\mathbf{x}_o^{k\to k+i} - \mathbf{x}_d^{k\to k+i}||_2^2, \tag{9}$$

where $\mathbf{x}_o^{k\to k+i}$ and $\mathbf{x}_d^{k\to k+i}$ are $\mathbf{x}$'s corresponding pixels in frame $k+i$ according to the estimated optical flow and depth, respectively. If pixel $\mathbf{x}$ is static, $\mathbf{x}_o^{k\to k+i}$ and $\mathbf{x}_d^{k\to k+i}$ should be very close. Otherwise, $||\mathbf{x}_o^{k\to k+i} - \mathbf{x}_d^{k\to k+i}||_2^2$ should be large. W is a normalization factor, defined as $W = \sum_{i=-n}^{n} P_{opt}^{k\to k+i}(\mathbf{x})$.

5.3. Disparity Error Detection

Directly combining the computed measures may have problems, since they are computed with different ways and have respective characteristic. Especially, the estimated motion/depth information may contain significant errors, which will harm the segmentation results. Figure 3 shows

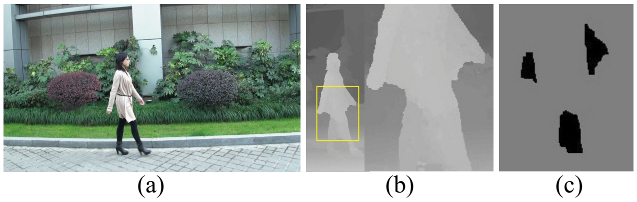


Figure 3. Disparity error detection. (a) A center image. (b) The initial estimated disparity map, which contains noticeable errors marked by a yellow rectangle. (c) The detected disparity errors.

an example, where the region marked by the yellow rectangle contain obvious disparity errors, so that the computed disparity difference measure will be unreliable. Figures 4(a)-(b) show a segmentation result directly using the computed disparity difference measure. Zhang *et al.* [17] used a voting-like scheme to alleviate this problem. Here, we propose a disparity error detection method to more effectively utilize the computed depth cue.

We found that the estimated disparity of a pixel $\mathbf{x}$ is very likely to be inaccurate, if its corresponding disparity difference measure $\mathcal{D}_d(\mathbf{x})$ is large but color difference measure $\mathcal{D}_c(\mathbf{x})$ is quite small, except the case that the dynamic region have similar colors with the static background. The intuition is that if the estimated disparity is accurate, the pixel with large $\mathcal{D}_d(\mathbf{x})$ should be in dynamic regions, so that $\mathcal{D}_c(\mathbf{x})$ also should be large. In order to address the ambiguity caused by color similarity, we further use motion-depth difference measure, and define the following rule: if a pixel $\mathbf{x}$, whose $\mathcal{D}_d(\mathbf{x})$ is large but $\mathcal{D}_c(\mathbf{x})$ and $\mathcal{D}_o(\mathbf{x})$ are both small, we think the estimated disparity value is incorrect.

Based on the above rule, we formulate the disparity error detection as a labeling problem. Let L be a binary map where $L(\mathbf{x})$ equals to 0 if the disparity of pixel $\mathbf{x}$ is wrong, and 1 otherwise. The energy function is defined as:

$$E^l(L) = E^l_d(L) + \beta E^l_s(L),$$

where $E^l_d(L)$ is data term, and $E^l_s(L)$ is the smoothness term defined the same as (3). β is the smoothness weight. $E^l_d(L)$ is define as:

$$E^l_d(L) = \sum_{\mathbf{x}} (T_c(1-L(\mathbf{x})) + \frac{\mathcal{D}_d(\mathbf{x})}{\max(\mathcal{D}_c(\mathbf{x}), \lambda_o \mathcal{D}_o(\mathbf{x})) + T_d} \cdot L(\mathbf{x})),$$

where T_c and T_d are thresholds and λ_o controls the weight of the motion-depth difference measure. We use BP algorithm to solve the energy function. Figure 3(c) shows the detected disparity errors.

5.4. Segmentation Model

After detecting the disparity errors, the dynamic likelihood of pixel $\mathbf{x}$ can be defined as:

$$\mathcal{L}(\mathbf{x}) = \frac{\mathcal{D}_c(\mathbf{x}) \cdot (1 - L(\mathbf{x})) + \max(\mathcal{D}_c(\mathbf{x}), \mathcal{D}_d(\mathbf{x})) \cdot L(\mathbf{x})}{1 + \exp(-(D(\mathbf{x}) - M)/V)}, \tag{10}$$

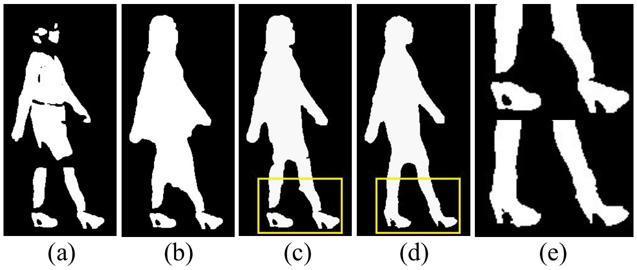


Figure 4. Detection of dynamic regions. (a) The segmentation result only using color difference measure. (b) The segmentation result using both of color and disparity difference measures. (c) The initial segmentation result by solving (11). (d) The segmentation result after temporal refinement. (e) The magnified regions of (c) and (d).

where $D(\mathbf{x})$ is the disparity of $\mathbf{x}$. Because pixels with small disparities are usually in static regions, we add a denominator $1 + \exp(-(D(\mathbf{x}) - M)/V)$ into this equation, and use parameters M and V to control the function shape. Since optical flow estimation is not reliable, the estimated motion-depth difference measure may have problems. Therefore, we only use it to detect disparity errors and do not consider it in (10). With this likelihood definition, we define the following segmentation energy function:

$$\begin{aligned} E_S(S) = &\sum_{\mathbf{x}} (\mathcal{L}(\mathbf{x})(1 - S(\mathbf{x})) + (1 - \mathcal{L}(\mathbf{x}))S(\mathbf{x})) \\ &+ \beta_S \sum_{\mathbf{x}} \sum_{\mathbf{y} \in N(\mathbf{x})} \lambda(\mathbf{x}, \mathbf{y}) \rho_s(S(\mathbf{x}), S(\mathbf{y})), \end{aligned} \tag{11}$$

where β_S is the smoothness weight. $\lambda(\cdot)$ and $\rho_s(\cdot)$ are defined the same as in Equation (3). S is the segmentation map, where $S(\mathbf{x}) = 1$ if pixel $\mathbf{x}$ is dynamic, and $S(\mathbf{x}) = 0$ otherweise. We use Graph Cuts [5] to solve (11) for each frame independently. Figure 4(c) shows a segmentation result.

We combine the local classifier model [1] and our spatio-temporal prior together to further improve the segmentation result. First, a set of overlapping local classifiers are constructed around the initial segmentation boundaries, so that for each pixel $\mathbf{x}$, its foreground probability $P^c_F(\mathbf{x})$ and background probability $P^c_B(\mathbf{x})$ can be computed. Then we apply optical flow to make segmentation more temporally consistent. For pixel $\mathbf{x}$ in frame k, the probability that l is the segmentation label of $\mathbf{x}$ can be defined as:

$$\begin{aligned} P_s(l|\mathbf{x}) = \sum_{i=1}^{2n+1} \sum_{\mathbf{y} \in W(\mathbf{x})} &(\exp(-||l - S_{k_i}(\mathbf{y})||_2^2/T_1) \cdot P_{opt}(\mathbf{y}) \cdot \\ &\exp(-||\mathbf{y} - \mathbf{x}||_2^2/T_2) \cdot \frac{T_3}{T_3 + ||I(\mathbf{x}) - I(\mathbf{y})||_2}), \end{aligned} \tag{12}$$

where $\{S_{k_1}, ..., S_{k_{2n+1}}\}$ is generated by warping the segmentation results of frames $\{k - n, ..., k + n\}$ to frame k by optical flow. $W(\mathbf{x})$ is a local window centered at $\mathbf{x}$, and P_{opt} is optical flow reliability (Equation (6)). T_1, T_2 and T_3

are constants. The normalized probability $P(l|\mathbf{x})$ is:

$$P(l|\mathbf{x}) = \frac{P_s(l|\mathbf{x})}{(P_s(l=0|\mathbf{x}) + P_s(l=1|\mathbf{x}))}.$$

We denote $P_F^t(\mathbf{x}) = P(l=1|\mathbf{x})$ and $P_B^t(\mathbf{x}) = P(l=0|\mathbf{x})$. The data cost for segmentation can be finally written as:

$$E_d^S(S) = -\sum_{\mathbf{x}}((1-S(\mathbf{x}))\log(P_F(\mathbf{x})) + S(\mathbf{x})\log(P_B(\mathbf{x}))), \tag{13}$$

where $P_F(\mathbf{x}) = P_F^c(\mathbf{x})P_F^t(\mathbf{x})$ and $P_B(\mathbf{x}) = P_B^c(\mathbf{x})P_B^t(\mathbf{x})$. The smoothness term is defined the same as in (11). We use Graph Cuts [5] to solve it. The refinement can be performed iteratively. Two iterations are enough in our experiments. Figure 4(d) shows the refined segmentation result.

6. Spatio-Temporal Depth Optimization

Due to the lack of explicit temporal coherence constraint, the initialized depth maps may be not consistent among neighboring frames. After segmentation, we can perform spatio-temporal depth optimization for static and dynamic pixels with different models. For static pixels, we can use the bundle optimization model [18] to refine the depth. For dynamic pixels, we define the following two motion models.

6.1. Local Linear Motion Model

The motion of an object can be usually decomposed as the motion of several rigid bodies, so we assume that the 3D trajectory of a point in dynamic regions can be approximated to a 3D line within a short time. For the points that violate this assumption, we will use a different model introduced in the next subsection.

For 2D point $\mathbf{x}'$ in frame k, its corresponding point set $\hat{\mathbf{x}} = \{\mathbf{x}'_{k-n}, ..., \mathbf{x}'_{k-1}, \mathbf{x}'_{k+1}, ..., \mathbf{x}'_{k+n}\}$ in neighbor frames from $k-n$ to $k+n$ can be found by optical flow. n is set to $2 \sim 4$ in our experiments. With the estimated depth maps, we project the points in set $\hat{\mathbf{x}}$ to 3D space, denoted as $\{X_1, X_2, ..., X_{2n+1}\}$.

The parametric equation of a line can be represented by $l: X = X_0 + tv$, where X_0 is a point on the line and v is a direction vector. Given the 3D point set, the line can be easily computed using PCA. Let μ be the average of these $2n+1$ points and H be their covariance matrix decomposed as $H = Q\Sigma Q^T$, where Q is an orthogonal matrix and Σ is a diagonal matrix whose diagonal entries are the eigenvalues of H. The first and second largest eigenvalues of H are denoted as λ_1 and λ_2, respectively. Then parameter X_0 and v can be set to μ and λ_1, respectively. If $\lambda_1/\lambda_2 \geq \tau$ ($\tau = 50$ in our experiments), we think the 3D trajectory of $\mathbf{x}$ can be approximately described by a 3D line.

Then the disparity likelihood can be defined as:

$$P(d|\mathbf{x}) = \exp(-\frac{\mathrm{Dis}(X(\mathbf{x},d),l)^2}{v(\mathbf{x})^2 \cdot \sigma_t}), \tag{14}$$

where $\mathrm{Dis}(\mathbf{y},l)$ is the distance from point $\mathbf{y}$ to line l. $X(\mathbf{x},d)$ is the 3D position of $\mathbf{x}$ given disparity d, and $v(\mathbf{x}) = \min_d \mathrm{Dis}(X(\mathbf{x},d),l)$. Parameter σ_t controls the function shape. Obviously, if the distance from the corresponding 3D point to the fitted 3D line is small, $P(d|\mathbf{x})$ will be large. Otherwise, $P(d|\mathbf{x})$ will be small.

6.2. Planar Motion Model

If a point violates the above assumption (i.e. $\lambda_1/\lambda_2 < \tau$), we alternatively assume that it approximately moves in a constant depth plane among neighboring frames. We use optical flow to estimate the disparity likelihood $P_p(d|\mathbf{x})$ by:

$$\begin{aligned} P(d|\mathbf{x}) =& \sum_{i=1}^{3}\sum_{\mathbf{y}\in W(\mathbf{x})}(\exp(-||d - D_{k_i}(\mathbf{y})||_2^2/T_1) \cdot P_{opt}(\mathbf{y}) \cdot \\ & \exp(-||\mathbf{y}-\mathbf{x}||_2^2/T_2) \cdot \frac{T_3}{T_3 + ||I(\mathbf{x}) - I(\mathbf{y})||_2}. \end{aligned} \tag{15}$$

Here, $\{D_{k_1}, D_{k_2}, D_{k_3}\}$ is generated by warping the disparity maps of frames $\{k-1, k, k+1\}$ to frame k by optical flow. It is similar to (12), estimating the distribution by sampling the disparities among neighboring frames. Because the disparity of a dynamic point may vary quickly, we only select two neighboring frames to estimate $P(d|\mathbf{x})$.

6.3. Depth Optimization

We combine bundle optimization and our proposed motion models together. The data term for refinement of dynamic regions is defined as:

$$E_{d_1}(D;I) = E_d(D,O;I) - \lambda\sum_{\mathbf{x}}(\log P(D(\mathbf{x})|\mathbf{x})), \tag{16}$$

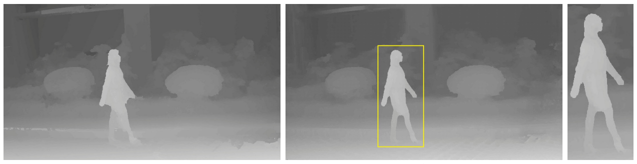
where $E_d(D,O;I)$ is the data term defined in (2) for trinocular stereo matching, and $P(D(\mathbf{x})|\mathbf{x})$ is Equation (14) or (15) according to different situations. λ controls the weight of the prior.

The data term $E_{d_2}(D;I)$ for refinement of static regions is defined as follows:

$$\begin{aligned} E_{d_2}(D;I) =& E_d(D,O;I) + \sum_{\mathbf{x}}\Bigg(1 - u(\mathbf{x}) \cdot \\ & \sum_{t'} p_c(\mathbf{x}, D(\mathbf{x}), I, I_{t'}) \cdot p_v(\mathbf{x}, D(\mathbf{x}), D_{t'})\Bigg), \end{aligned}$$

where p_c is the same to Equation (2) in [18], and p_v is similar to Equation (8) in [18]. The minor difference is that the geometric coherence term p_v is defined in disparity space instead of image space. Please refer to our supplementary material for more details. $u(\mathbf{x})$ is the normalization factor defined the same as [18]. Because it is better to softly combine $E_{d_1}(D;I)$ and $E_{d_2}(D;I)$, we define the combinatorial data term as:

$$E_d^f(D;I) = \sum_{\mathbf{x}}(E_{d_1}(D(\mathbf{x});I) \cdot P_F^f(\mathbf{x}) + E_{d_2}(D(\mathbf{x});I) \cdot P_B^f(\mathbf{x})).$$



(a) (b) (c)

Figure 5. Spatio-temporal depth optimization. (a) The initial estimated depth map. (b) The refined disparity map after spatio-temporal optimization. (c) The magnified region of (b).

Here, $P_F^f(\mathbf{x})$ is the probability that pixel $\mathbf{x}$ is in the dynamic regions, defined as:

$$P_F^f(\mathbf{x}) = \begin{cases} S(\mathbf{x}), & \text{if } Dis(\mathbf{x}) > T_b \\ P_F(\mathbf{x})/(P_F(\mathbf{x}) + P_B(\mathbf{x})), & \text{otherwise} \end{cases}$$

where $P_F(\mathbf{x})$ and $P_B(\mathbf{x})$ are already introduced in Equation (13). S is the segmentation map. $Dis(\mathbf{x})$ is the distance from $\mathbf{x}$ to the segmentation boundary and T_b is a threshold. Here, $P_B^f(\mathbf{x})$ can be computed by $P_B^f(\mathbf{x}) = 1 - P_F^f(\mathbf{x})$.

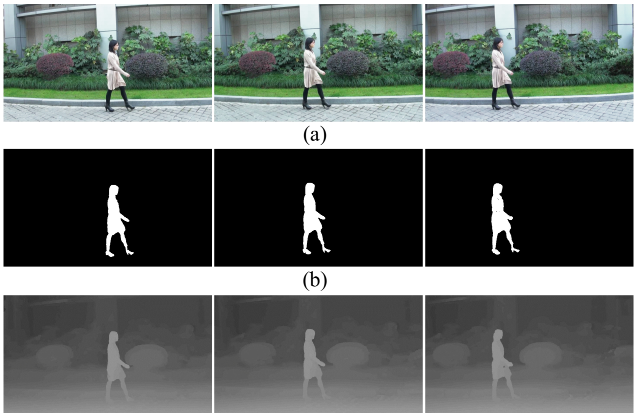
The smoothness term $E_s(D; I)$ is defined the same as (3). We use BP to minimize the energy function $E(D; I) = E_d^f(D; I) + E_s(D; I)$. While refining one disparity map, other disparity maps are fixed. Two iterations are generally sufficient. Figure 5(b) shows the refined disparity map.

6.4. Depth Recovery of Left and Right Sequences

After obtaining the segmentation and depth maps of the center sequence, we can quickly generate the segmentation and depth maps for the left/right sequences, by warping the estimated segmentation and depth maps of the center sequence to the left/right ones. Due to occlusion, the warped depth map may contain some holes. We use mean-shift segmentation information to infer the depth values of these missing pixels. If a segment contains both static and dynamic pixels (based on the warped segmentation masks), it will further split until a segment only contains either static or dynamic pixels. If a segment contains missing pixels, we will fit the 3D plane using the pixels with available depth values, and then fill the depth values for missing pixels. Finally, we employ the spatio-temporal optimization method introduced in Section 6 to refine the depth maps. With this strategy, the initialization stage can be saved, and the obtained depth maps of left/rigth sequence are naturally consistent with the center ones.

7. Experimental Results

We have conducted experiments on several challenging trinocular video sequences to evaluate the performance of our method. All the experiments are conducted on a desktop PC with Intel i7 3.4GHz CPU, 4G memory and NVIDIA GeForce GTX 560 Ti display card. Data cost computation of stereo matching has been accelerated by GPU in our implementation. It takes about 3.0 minutes per frame to recover the segmentation and depth maps of the center sequence with 960 × 540 image resolution. Table 2 lists the parameter values used in our experiments.



(a)

(b)

(c)

Figure 6. “Walking” example. (a) The trinocular stereo images of one time instance. (b) The extracted dynamic regions. (c) The computed disparity maps.

Table 2. Parameter configuration.

η	T	ϵ	β	β_w	β_o	β_S	ϵ_c	ϵ_d	ϵ_o	λ_o
0.5	2	50	0.1	1.0	1.0	0.1	80-240	8-15	5	0-0.5
T_c	T_d	M	V	T_1	T_2	T_3	τ	σ_t	λ	T_b
2.5	0.2	7	1.0	2	8	10	50	16	0.1	3

The “Walking” sequence is shown in Figure 6. The estimated dynamic regions and depth maps are shown in Figure 6(b) and (c), respectively. Figure 7 shows an example containing two moving objects. Figure 8 shows another challenging sequence “Indoor” captured by a fixed camera, which contains two men, a statue, and very large textureless regions. Our method faithfully reconstructs all these structures. The recovered disparity maps are shown in Figures 8(d)-(e). To verify the accuracy of the recovered disparities, we warp the center image to the left view and make a comparison. Figure 8(c) shows the warped image, and (f) shows the difference image, which demonstrates the accuracy of the estimated disparity map. Please refer to our supplementary material for more results.

8. Conclusions

In this paper, we propose a novel dense depth recovery method which can automatically recover accurate and consistent depth maps from a trinocular video sequence. A novel trinocular stereo matching model is introduced, which can robustly estimate the depth map of center image given the trinocular stereo images in the same time instance. With the initialized depth maps, we first separate the static and dynamic pixels, and then refine their depths with different

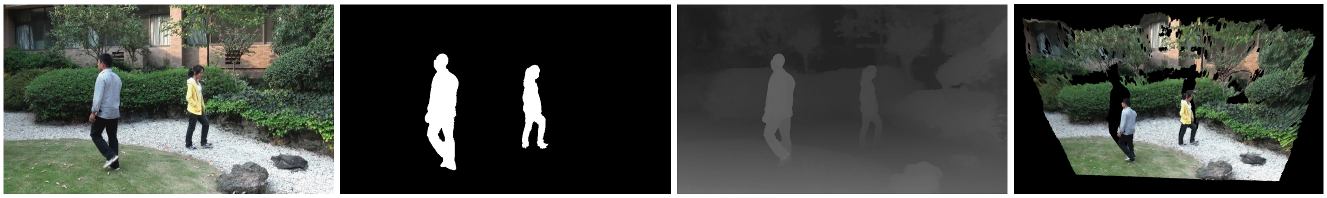
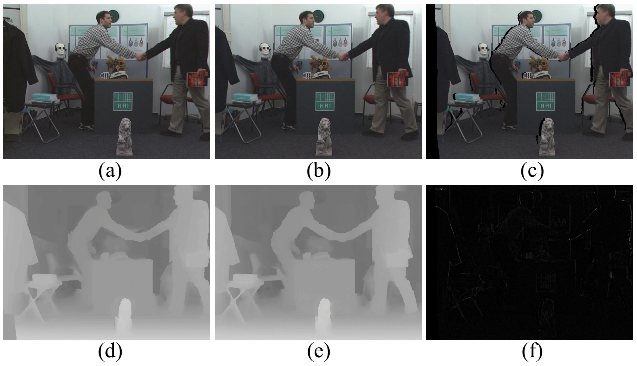


Figure 7. “Two Persons” example. From left to right: a center image, segmentation map, the estimated disparity map and a synthesized novel view.



(a) (b) (c)

(d) (e) (f)

Figure 8. “Indoor” example. (a) One selected image in the left sequence. (b) One selected image in the center sequence. (c) Warping the center image to the left image with the estimated disparity map. (d) The estimated disparity map of (a). (e) The estimated disparity map of (b). (f) The difference between (a) and (c).

spatio-temporal optimization models. The experimental results demonstrate the effectiveness of the proposed method.

If the moving objects do not have sufficient movement, or have very similar colors with the background, our moving object detection method may recognize it as background with default parameters. If this happens, we can tune the parameters to avoid recognizing the dynamic pixels as static ones. If a background region is recognized as moving region, it may not harm the depth estimation because our spatio-temporal depth refinement for dynamic pixels also can be used for static pixels.

Acknowledgements. This work is partially supported by the 973 program of China (No. 2009CB320802), NSF of China (No. 60903135), the Specialized Research Fund for the Doctoral Program of Higher Education of China (Grant No. 20110101130011), the Fundamental Research Funds for the Central Universities, and a research grant from Samsung Advanced Institute of Technology.

References

[1] X. Bai, J. Wang, D. Simons, and G. Sapiro. Video snapcut: Robust video object cutout using localized classifiers. In *Siggraph*, 2009.

[2] T. Basha, Y. Moses, and N. Kiryati. Multi-view scene flow estimation: A view centered variational approach. In *CVPR*, 2010.

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

[4] M. Bleyer, C. Rother, and P. Kohli. Surface stereo with soft segmentation. In *CVPR*, 2010.

[5] 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.

[6] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient belief propagation for early vision. *International Journal of Computer Vision*, 70(1):41–54, 2006.

[7] Y. Furukawa and J. Ponce. Accurate, dense, and robust multiview stereopsis. *TPAMI*, 32:1362–1376, 2010.

[8] B. Goldlucke and M. A. Magnor. Joint 3D-reconstruction and background separation in multiple views using graph cuts. In *CVPR*, 2003.

[9] S. Larsen, P. Mordohai, M. Pollefeys, and H. Fuchs. Temporally consistent reconstruction from multiple video streams using enhanced belief propagation. In *ICCV*, 2007.

[10] C. Lei, X. Chen, and Y. Yang. A new multiview spacetime-consistent depth recovery framework for free viewpoint video rendering. In *ICCV*, 2009.

[11] H. RI and Z. A. *Multiple View Geometry in Computer Vision, 2nd edition*. Cambridge University Press, 2004.

[12] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47(1/2/3):7–42, 2002.

[13] D. Scharstein and R. Szeliski. High-accuracy stereo depth maps using structured light. In *CVPR (1)*, pages 195–202, 2003.

[14] J. Sun, Y. Li, S. B. Kang, and H.-Y. Shum. Symmetric stereo matching for occlusion handling. In *CVPR*, 2005.

[15] H. H. Vu, R. Keriven, P. Labatut, and J.-P. Pons. Towards high-resolution large-scale multi-view stereo. In *CVPR*, pages 1430–1437, 2009.

[16] M. Yang, X. Cao, and Q. Dai. Multiview video depth estimation with spatial-temporal consistency. In *BMVC*, 2010.

[17] G. Zhang, J. Jia, W. Hua, and H. Bao. Robust bilayer segmentation and motion/depth estimation with a handheld camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (99):1–1, 2011.

[18] G. Zhang, J. Jia, T.-T. Wong, and H. Bao. Consistent depth maps recovery from a video sequence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(6):974–988, 2009.

[19] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski. High-quality video view interpolation using a layered representation. In *Siggraph*, pages 600–608, 2004.