3D Reconstruction of Dynamic Scenes with Multiple Handheld Cameras

Hanqing Jiang[1,2], Haomin Liu[1], Ping Tan[2], Guofeng Zhang[1⋆], and Hujun Bao[1]

[1]State Key Lab of CAD&CG, Zhejiang University
[2]National University of Singapore

Abstract. Accurate dense 3D reconstruction of dynamic scenes from natural images is still very challenging. Most previous methods rely on a large number of fixed cameras to obtain good results. Some of these methods further require separation of static and dynamic points, which are usually restricted to scenes with known background. We propose a novel dense depth estimation method which can automatically recover accurate and consistent depth maps from the synchronized video sequences taken by a few handheld cameras. Unlike fixed camera arrays, our data capturing setup is much more flexible and easier to use. Our algorithm simultaneously solves bilayer segmentation and depth estimation in a unified energy minimization framework, which combines different spatio-temporal constraints for effective depth optimization and segmentation of static and dynamic points. A variety of examples demonstrate the effectiveness of the proposed framework.

Keywords: multi-view stereo, depth recovery, dynamic scene, spatio-temporal optimization

1 Introduction

Many 3D reconstruction algorithms such as [11, 20] have been proposed for a static scene. However, the real world is full of dynamic objects. The reconstruction of a dynamic scene is often required in many applications, but is also much more challenging. Most works on dynamic scene reconstruction such as [23, 8, 9] require an array of fixed cameras with relatively narrow baselines to minimize occlusion problems. Further, some of these, such as [23, 8, 9] require a large number of cameras (usually more than eight) to ensure good results. These requirements on data capturing limit their applications and are typically applied to data carefully captured in laboratories. There are recent non-rigid structure-from-motion algorithms such as [14], which reconstruct dynamic scenes with a single moving camera. Though data capturing is simplified, their accuracy and robustness are rather limited at this moment.

We study dense 3D reconstruction of dynamic scenes from multiple handheld cameras. We allow cameras to move freely and independently so that data

⋆ Email: zhangguofeng@cad.zju.edu.cn

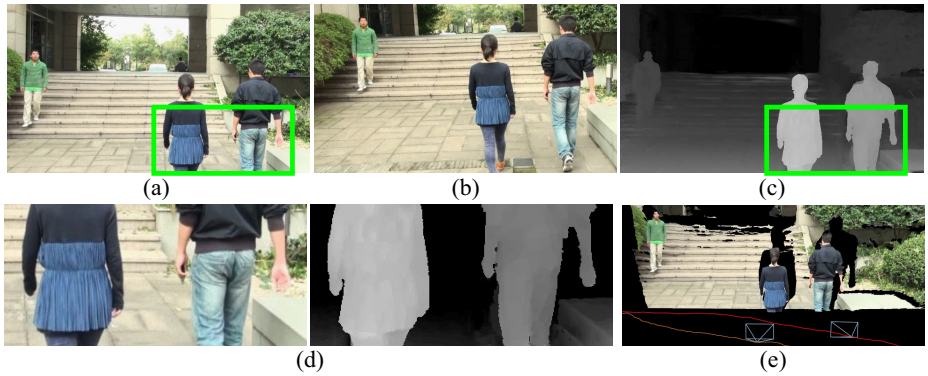

Fig. 1. Dynamic 3D reconstruction results of 'Three-Pedestrian' example. (a-b) Two synchronous frames from the two source videos. (c) The recovered depth map of (a). (d) The magnified regions of (a) and (c). (e) The texture-mapped 3D scene visualized from a novel viewpoint.

capturing is much more flexible. Further, we require as few as $2 \sim 3$ cameras to facilitate many applications. An example is shown in Fig. 1, which contains three pedestrians. Our method faithfully recovers the depths in both static and dynamic regions. Compared with previous methods with a fixed camera array, our approach achieves both flexible data capturing and accurate 3D reconstruction.

Most previous works on dynamic scene reconstruction [12, 8, 15] treated moving objects in the same way as static ones. Some methods such as [5, 6, 19] segmented pixels into static and dynamic ones while estimating their depths. These methods require exact bilayer segmentation, which itself is a difficult problem. In comparison, we propose a unified global optimization framework to solve dense depth recovery and static/dynamic segmentation simultaneously. Unlike previous methods such as [5, 6, 19], our segmentation is optimized for better depth estimation. A pixel is classified to "static" or "dynamic" depending on which kind of spatio-temporal constraint models it better. Hence, our method does not require an exact static/dynamic segmentation and is more robust to segmentation errors in challenging scenes like Fig. 6.

2 Related Work

There are many methods such as [22, 12, 23] to reconstruct a dynamic scene from multiple fixed cameras. They typically applied stereo algorithms to synchronized video frames of different cameras and smoothed the estimated disparities of corresponding pixels in the temporal domain. Here, we only briefly review some of the recent works. In [8], temporal consistency between consecutive frames in multiple cameras was enforced by an enhanced belief propagation algorithm. Aguiar et al. [1] captured temporal coherent human performances with a circular camera array. Lei et al. [9] recovered spacetime-consistent depth maps by a

region-tree based stereo approach. Recently, Yang et al. [15] extended the powerful bundle optimization framework [20] to handle dynamic scenes with a novel temporal coherence constraint. This constraint is similar to our spatio-temporal constraint for dynamic points. However, [15] only used one preceding time instance for depth optimization, and is rather sensitive to optical flow errors and occlusions. Hence, their method required more cameras with relative small baselines to ensure robust results. In comparison, our spatio-temporal optimization can effectively utilize more temporal neighboring frames and robustly handle occlusions. As a result, we require only a few cameras and allow cameras to move freely with larger baselines to significantly simplify data capturing.

Bilayer segmentation is an important cue for dynamic scene reconstruction. Zhang et al. [19] proposed a robust bilayer segmentation by dense depth and motion estimation. Although the segmentation and depth/motion estimation are incorporated into an energy function, they are separately solved which may result in a solution that is only locally optimal. In addition, their method requires a manual preprocessing for foreground color distribution learning. Goldlücke and Magnor [5] performed bilayer segmentation and 3D reconstruction for static scenes with fixed cameras, where background images have been recorded from each camera beforehand. Guillemaut et al. [6] used multi-layer segmentation to achieve 3D reconstruction and free-viewpoint video of sports videos captured by multiple broadcast cameras, where the planar background color and depth is also known. Both methods used joint labeling to simultaneously estimate depths and segmentation, which is similar to our work. Nevertheless, both of these approaches require strong prior knowledge on the background scene to facilitate segmentation. Further, their depth estimation and segmentation terms were weakly coupled in the optimization formulation, making these methods easy to get stuck in a local minimum. In comparison, our depth recovery and bilayer segmentation are tightly integrated, and do not require prior background knowledge. Recently, Zhang et al. [18] used a similar expanded labeling formulation for simultaneous stereo and segmentation, but was limited to rigid objects.

There are other works with similar data capturing setup, but targeting different applications. Ballan et al.[2] addressed the video based rendering problem with multiple freely moving cameras, where dynamic objects are only segmented but not reconstructed. Hasler et al.[7] used multiple freely moving cameras for markerless motion capture. Most recently, Yang et al. [16] proposed a dense depth recovery method for a trinocular sequence. In their capture setup, the relative camera poses are constant and the baselines between neighboring cameras should be not large (8∼20cm). To the best of our knowledge, our work is the first to recover dense depth maps for dynamic scenes by multiple handheld cameras.

3 Notation and System Summary

Given a set of synchronized video sequences $\hat{I} = \{\hat{I}_m | m = 1, ..., M\}$ taken by M moving cameras, in which each $\hat{I}_m$ has N frames denoted as $\hat{I}_m = \{I_m^t | t = 1, ..., N\}$, our goal is to recover a set of corresponding depth map sequences

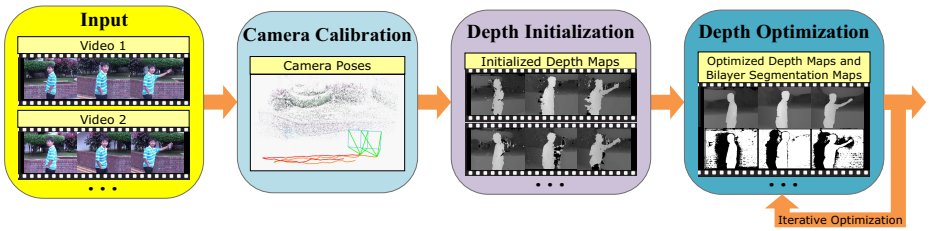

Fig. 2. This figure shows the overview of our system.

$\hat{Z} = \{Z_m^t | m = 1, ..., M; t = 1, ..., N\}$. We use $I_m^t(\mathbf{x})$, $Z_m^t(\mathbf{x})$, $D_m^t(\mathbf{x}) = 1/Z_m^t(\mathbf{x})$ to represent the color, depth and disparity at pixel $\mathbf{x}$ in frame t of sequence m, respectively. For clarity, we sometimes use $\mathbf{x}_m^t$ to denote pixel $\mathbf{x}$ in frame (m, t).

The camera parameters for frame t of video sequence m is denoted by $\mathbf{C}_m^t = \{\mathbf{K}_m^t, \mathbf{R}_m^t, \mathbf{T}_m^t\}$, where $\mathbf{K}_m^t$ is the intrinsic matrix, $\mathbf{R}_m^t$ is the rotation matrix, and $\mathbf{T}_m^t$ is the translation vector. The camera parameters for all frames in all sequences are estimated beforehand. Specifically, we use the method in [17] to track feature points in the videos and then use the SFM method proposed in [21] to recover the camera poses. To ensure system robustness, camera intrinsics are pre-calibrated and fixed during data capturing.

The system overview is shown in Fig. 2. We initialize the disparity maps $\{D_m^t | m = 1, ..., M\}$ for each time t using the synchronized images $\{I_m^t | m = 1, ..., M\}$ across M cameras (see Section 4). Then we jointly perform spatio-temporal disparity optimization and bilayer segmentation (see Section 5) so that a set of high-quality depth maps are finally obtained.

4 Depth Initialization

We first estimate the depth maps at time t using the images from different cameras at that time. Denoting the disparity range as $[d_{\min}, d_{\max}]$, we equally quantize the disparity into k levels, such that the ith level $d_i = (k-i)/(k-1) \cdot d_{\min} + (i-1)/(k-1) \cdot d_{\max}$, where $i = 1, \ldots, k$. Specifically, we minimize the following objective energy:

$$E_D(D_m^t; \hat{I}(t)) = E_d(D_m^t; \hat{I}(t)) + E_s(D_m^t), \tag{1}$$

where $\hat{I}(t) = \{I_m^t | m = 1, ..., M\}$ represents all frames captured at time t. E_d is the data term, and E_s is the smoothness term.

The smoothness term E_s measures the smoothness of neighboring disparities, and is simply defined as:

$$E_s(D_m^t) = \lambda \sum_{\mathbf{x}} \sum_{\mathbf{y} \in N(\mathbf{x})} \min\{|D_m^t(\mathbf{x}) - D_m^t(\mathbf{y})|, \eta\}, \tag{2}$$

where $N(\mathbf{x})$ denotes the set of neighbors of pixel $\mathbf{x}$, λ is the smoothness weight, and η is a truncation value for robust estimation. In our experiments, $\lambda = 0.8/(d_{\max} - d_{\min})$ and $\eta = 0.03(d_{\max} - d_{\min})$.

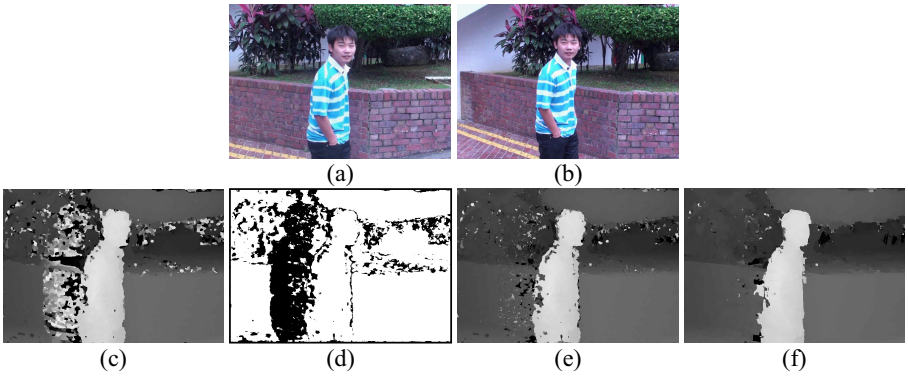

Fig. 3. Illustration of disparity initialization. (a-b) Two synchronous source frames from the 'Boy' data. (c) The estimated disparity map of (a) using the data term defined in Eq. (4), incorporating DAISY descriptors. (d) The visibility map estimated from the disparity maps of (a) and (b). (e) The refined disparity map after occlusion handling. The disparities in the occluded regions are improved. (f) The further refined disparity map after plane fitting.

For conventional stereo algorithms, the data term E_d mainly uses pixel-wise or window-based color similarity. In our examples, baselines between cameras are often large, as shown in Fig. 3(a-b). Due to the foreshortening and occlusions, a simple color similarity measure is insufficient to generate good disparities. To better solve this problem, we use DAISY [13] in the data term. DAISY is an efficient local descriptor for robust wide baseline stereo matching. We define a disparity cost function based on the similarity of DAISY descriptor as:

$$L_{\mathcal{D}}(\mathbf{x}_m^t, d_i; I_m^t, I_{m'}^t) = ||\mathcal{D}(\mathbf{x}_m^t) - \mathcal{D}(\mathbf{x}_{m'}^t)||_2, \tag{3}$$

where $\mathcal{D}(\mathbf{x})$ is the DAISY descriptor at $\mathbf{x}$. The term $\mathbf{x}_{m'}^t = l_{m\to m'}^t(\mathbf{x}_m^t, d_i)$ is the corresponding pixel of $\mathbf{x}_m^t$ by projecting $\mathbf{x}_m^t$ to $I_{m'}^t$ according to the candidate disparity value d_i and camera parameters.

We then define the data term E_d as:

$$E_d(D_m^t; \hat{I}(t)) = \sum_{\mathbf{x}_m^t} \frac{\sum\limits_{m'\neq m} L_{\mathcal{D}}(\mathbf{x}_m^t, D_m^t(\mathbf{x}_m^t); I_m^t, I_{m'}^t)}{M-1}. \tag{4}$$

Basically, we use all the synchronous views in other cameras to initialize the disparity D_m^t. Loopy belief propagation algorithm [4] is used to solve Eq. (1). Fig. 3(c) shows an estimated depth map. There are still visible problems in the occluded regions. In the next subsection, we estimate a visibility map for each pair of synchronous images in cameras m and m' to further handle occlusions.

4.1 Disparity Inference of Occluded Pixels

If a point is visible in both cameras m and m', its depth in both images should be consistent (i.e. corresponding to the same 3D point). Hence, we define the

visibility map of camera m with respect to m' as:

$$V^t_{m\to m'}(\mathbf{x}^t_m) = \begin{cases} 1 & |1/Z^t_{m\to m'}(\mathbf{x}^t_m) - D^t_{m'}(\mathbf{x}^t_{m'})| \le \delta_d \\ 0 & |1/Z^t_{m\to m'}(\mathbf{x}^t_m) - D^t_{m'}(\mathbf{x}^t_{m'})| > \delta_d \end{cases}, \tag{5}$$

where $V^t_{m\to m'}(\mathbf{x}^t_m)$ denotes whether the pixel $\mathbf{x}^t_m$ in I^t_m is visible in $I^t_{m'}$ (1 for visible and 0 for invisible). Here, δ_d is a threshold determined empirically as $0.02(d_{\max} - d_{\min})$. The term $Z^t_{m\to m'}(\mathbf{x}^t_m)$ denotes the depth of $\mathbf{x}^t_m$ in m'. It is determined by casting $\mathbf{x}^t_m$ to 3D space according to its disparity $D^t_m(\mathbf{x}^t_m)$ from the camera m, and then projecting it to the camera m'. We define $\hat{V}^t_m(\mathbf{x}^t_m)$ as the total visibility of pixel $\mathbf{x}^t_m$. It is equal to 0, if $\mathbf{x}^t_m$ is invisible in all the other cameras. Otherwise, it is equal to 1. Fig. 3(d) shows the estimated visibility map.

The disparities can be well initialized only at pixels which are visible in at least one reference camera. For pixels with total visibility $\hat{V}^t_m(\mathbf{x}) = 0$, Eq. (4) does not capture its data cost well. Hence, we estimate their disparities using the traditional photo-consistency constraint from neighboring temporal frames.

Given a candidate disparity d_i and a reference frame t' in camera m', We define a disparity likelihood based on the color similarity measure between I^t_m and $I^{t'}_{m'}$ as follows:

$$L_c(\mathbf{x}^t_m, d_i; I^t_m, I^{t'}_{m'}) = p_c(\mathbf{x}^t_m, \mathbf{x}^{t'}_{m'}), \tag{6}$$

where p_c measures the color similarity of $\mathbf{x}^t_m$ and $\mathbf{x}^{t'}_{m'}$, and is similarly defined as in [20]:

$$p_c(\mathbf{x}^t_m, \mathbf{x}^{t'}_{m'}) = \frac{\sigma_c}{\sigma_c + \left\| I^t_m(\mathbf{x}^t_m) - I^{t'}_{m'}(\mathbf{x}^{t'}_{m'}) \right\|_1}. \tag{7}$$

The parameter σ_c controls the sensitivity of color difference. The term $\mathbf{x}^{t'}_{m'}$ is the projection pixel of $\mathbf{x}^t_m$ to $I^{t'}_{m'}$ according to disparity d_i and camera parameters.

We accumulate the disparity likelihood with all the cameras at nearby time instances (20 nearest frames in our experiments), and compute the data term of the totally invisible pixels as:

$$E_d(D^t_m; \hat{I}) = \sum_{\mathbf{x}^t_m} 1 - \frac{\sum_{m'=1}^{M} \sum_{t'\in N(t)} L_c(\mathbf{x}^t_m, D^t_m(\mathbf{x}^t_m); I^t_m, I^{t'}_{m'})}{M|N(t)|}, \tag{8}$$

where $N(t)$ denotes the set of nearby frames.

By incorporating the above data cost, we can obtain a refined disparity map, as in Fig. 3(e). To better handle textureless regions and disparity noise, we incorporate segmentation information into the disparity initialization. We use Mean-shift [3] to segment the image and consider each segment as a 3D plane with parameters $[a, b, c]$, which are fit to the computed data costs in the segment as in [20]. Disparities of pixels in the segment can be recomputed as $d_{\mathbf{x}} = ax + by + c$. Fig. 3(f) shows the further refined disparity map after plane fitting.

5 Spatio-Temporal Optimization

In principle, static and dynamic pixels require different coherence constraints for disparity optimization. A straightforward solution is to perform explicit bilay-

er segmentation by estimating the background information using methods such as [19] and reconstruct static and dynamic pixels with different strategies. However, accurate background estimation is difficult for many cases, especially when a dynamic object moves slightly such that the background is almost occluded in all frames.

We propose to solve depth optimization and bilayer segmentation in a unified framework. For each pixel, we not only estimate its depth but also infer whether it is static or dynamic. Therefore, each pixel $\mathbf{x}_m^t$ has two variables to be inferred – one is its disparity value $D_m^t(\mathbf{x}_m^t)$ and the other is its static/dynamic label $S_m^t(\mathbf{x}_m^t) \in \{0, 1\}$ (0 for static and 1 for dynamic). Iteratively inferring these two variables might easily get stuck in local minimum. Instead, we formulate these two variables in an expanded labeling set, which is defined as:

$$\mathcal{L} = \{l_1^0, l_2^0, \ldots, l_k^0, l_1^1, l_2^1, \ldots, l_k^1\}.$$

Each label l_i^s naturally encodes a static/dynamic label $s \in \{0, 1\}$ and a candidate disparity value d_i, where $i = 1, 2, \ldots, k$. With this compact representation, we can now estimate a joint labeling map $\mathbf{L}_m^t$ ($\mathbf{L}_m^t(\mathbf{x}_m^t) \in \mathcal{L}$) for each frame I_m^t by minimizing a new energy:

$$E_{\mathbf{L}}(\mathbf{L}_m^t; \hat{I}, \hat{D}) = E_d(\mathbf{L}_m^t; \hat{I}, \hat{D}) + E_s(\mathbf{L}_m^t). \tag{9}$$

Since the spatio-temporal coherence constraints for static and dynamic pixels are different, we use different data cost functions for l_i^0 and l_i^1, which we denote as e_d^0 and e_d^1 respectively.

5.1 Different Spatio-Temporal Data Costs on Bilayer Hypotheses

The two different data cost functions are combined for temporal coherence measure on the bilayer hypotheses. Both of them measure the consistency of color and geometry among multiple cameras and multiple temporal frames.

If a pixel $\mathbf{x}_m^t$ is static, we follow the bundle optimization algorithm proposed in [20] to compute its data cost with candidate depth d_i. Given a reference frame t' in camera m', we can project the pixel $\mathbf{x}_m^t$ to $I_{m'}^{t'}$ by d_i. The projection point is denoted as $\mathbf{x}_{m'}^{t'}$. Similar to [20], we can measure the color and geometry coherence between $\mathbf{x}_m^t$ and $\mathbf{x}_{m'}^{t'}$, and compute a disparity likelihood as

$$L_d(\mathbf{x}_m^t, d_i; I_m^t, I_{m'}^{t'}, D_{m'}^{t'}) = p_c(\mathbf{x}_m^t, \mathbf{x}_{m'}^{t'}) p_v(\mathbf{x}_m^t, \mathbf{x}_{m'}^{t'}), \tag{10}$$

where p_c is the color similarity defined as Eq. (7). p_v is the geometric coherence by computing the backward projection error, which is defined as:

$$p_v(\mathbf{x}_m^t, \mathbf{x}_{m'}^{t'}) = \frac{\sigma_w}{\sigma_w + \left\| \mathbf{x}_m^t - l_{m' \to m}^{t' \to t}(\mathbf{x}_{m'}^{t'}, D_{m'}^{t'}(\mathbf{x}_{m'}^{t'})) \right\|_1}. \tag{11}$$

The term $l_{m' \to m}^{t' \to t}(\mathbf{x}_{m'}^{t'}, D_{m'}^{t'}(\mathbf{x}_{m'}^{t'}))$ is the point by projecting $\mathbf{x}_{m'}^{t'}$ to I_m^t as in [20].

We accumulate L_d with all the cameras at nearby time instances (20 nearest frames), and compute the data cost of $\mathbf{x}_m^t$ on the static-pixel hypothesis as:

$$e_d^0(\mathbf{x}_m^t, d_i; \hat{I}, \hat{D}) = 1 - \frac{\sum_{m'=1}^{M} \sum_{t' \in N(t)} L_d(\mathbf{x}_m^t, d_i; I_m^t, I_{m'}^{t'}, D_{m'}^{t'})}{M|N(t)|}. \tag{12}$$

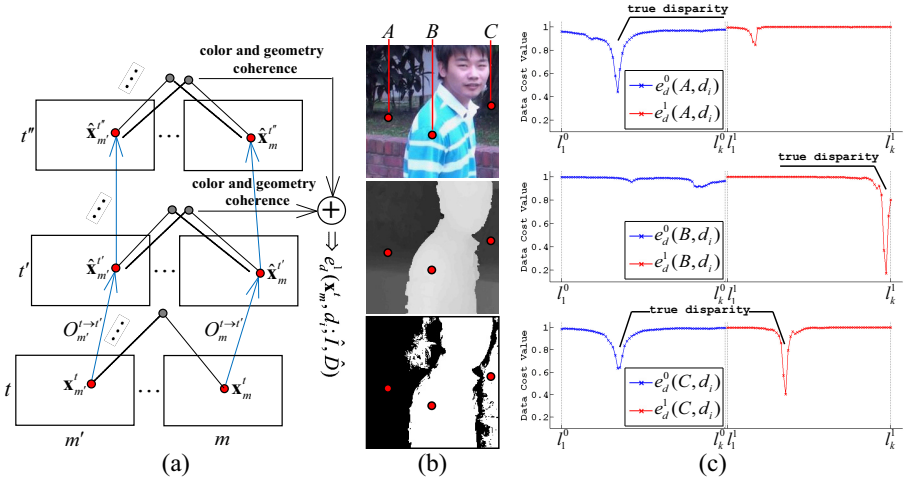

Fig. 4. (a) Illustration of the spatio-temporal coherence constraint on dynamic-pixel hypothesis. (b) The magnified source frame of Fig. 3(a), its spatio-temporally optimized depth map and the bilayer segmentation map. (c) The data cost values of the three representative pixels A, B and C in (c), computed by e_d^0 and e_d^1, which have a clear minimum to infer the true disparity robustly.

If the pixel $\mathbf{x}_m^t$ is dynamic, the data cost computation is more sophisticated. A straightforward way is to compute its depth among multiple synchronous cameras at time instance t also using bundle optimization as in [20]. However, most of our experimental examples have only $2 \sim 3$ cameras. The small number of synchronous views will significantly degrade the effectiveness of bundle optimization. To address this problem, we incorporate a spatio-temporal coherence measure similar to [15] for dynamic depth optimization. Unlike [15] which only used a preceding time instance for optimization, our data cost combines multiple nearby time instances to enhance the robustness of coherence measures.

Fig. 4(a) illustrates our spatio-temporal constraint for dynamic points. Specifically, given a candidate disparity d_i and a reference camera m', we can project the pixel $\mathbf{x}_m^t$ from camera m to m'. The projection point is denoted as $\mathbf{x}_{m'}^t$. To add their color and geometric coherence constraint from temporal frames at t', we track $\mathbf{x}_m^t$ and $\mathbf{x}_{m'}^t$ to t' by optical flows $O_m^{t\to t'}$ and $O_{m'}^{t\to t'}$ to get the corresponding pixels $\hat{\mathbf{x}}_m^{t'}$ and $\hat{\mathbf{x}}_{m'}^{t'}$. We compute the optical flow maps using the algorithm proposed in [10]. If the optical flows are accurate (i.e., $||O_m^{t\to t'}(\mathbf{x}_m^t) + O_m^{t'\to t}(\hat{\mathbf{x}}_m^{t'})|| < \tau$, and $||O_{m'}^{t\to t'}(\mathbf{x}_{m'}^t) + O_{m'}^{t'\to t}(\hat{\mathbf{x}}_{m'}^{t'})|| < \tau$, where $\tau = 3$ pixels in our experiments), we compute a disparity likelihood $L_g(\hat{\mathbf{x}}_m^{t'}, \hat{\mathbf{x}}_{m'}^{t'})$, by measuring the color and geometry coherence between $\hat{\mathbf{x}}_m^{t'}$ and $\hat{\mathbf{x}}_{m'}^{t'}$ as

$$L_g(\hat{\mathbf{x}}_m^{t'}, \hat{\mathbf{x}}_{m'}^{t'}; I_m^{t'}, I_{m'}^{t'}, D_m^{t'}, D_{m'}^{t'}) = p_c(\hat{\mathbf{x}}_m^{t'}, \hat{\mathbf{x}}_{m'}^{t'}) p_g(\hat{\mathbf{x}}_m^{t'}, \hat{\mathbf{x}}_{m'}^{t'}), \tag{13}$$

where p_g is the geometric coherence defined as:

$$p_g(\hat{\mathbf{x}}_m^{t'}, \hat{\mathbf{x}}_{m'}^{t'}) = \frac{\sigma_w}{\sigma_w + d_g(\hat{\mathbf{x}}_m^{t'}, \hat{\mathbf{x}}_{m'}^{t'}; D_m^{t'}, D_{m'}^{t'})}. \tag{14}$$

The term d_g is the symmetrical projection error. $\hat{\mathbf{x}}_m^{t'}$ is projected to frame $I_{m'}^{t'}$, and calculate its distance to $\hat{\mathbf{x}}_{m'}^{t'}$. Symmetrically, $\hat{\mathbf{x}}_{m'}^{t'}$ is projected to frame $I_m^{t'}$, and calculate its distance to $\mathbf{x}_m^{t'}$. d_g is the RMS of these two distances.

L_g is accumulated with all the reference cameras at nearby time instances (10 nearest frames in our experiments), and the data cost on dynamic-pixel hypothesis is defined as:

$$e_d^1(\mathbf{x}_m^t, d_i; \hat{I}, \hat{D}) = 1 - \frac{\sum_{t' \in N(t)} \sum_{m' \neq m} L_g(\hat{\mathbf{x}}_m^{t'}, \hat{\mathbf{x}}_{m'}^{t'}; I_m^{t'}, I_{m'}^{t'}, D_m^{t'}, D_{m'}^{t'})}{(M-1)|N(t)|}. \tag{15}$$

Now we can redefine the data term in Eq. (9) by combining the two kinds of data cost functions e_d^0 and e_d^1 as:

$$E_d(\mathbf{L}_m^t; \hat{I}, \hat{D}) = \sum_{\mathbf{x}_m^t} e_d(\mathbf{x}_m^t, \mathbf{L}_m^t(\mathbf{x}_m^t); \hat{I}, \hat{D}),$$

$$\text{with } e_d(\mathbf{x}_m^t, l_i^s; \hat{I}, \hat{D}) = \begin{cases} e_d^0(\mathbf{x}_m^t, d_i; \hat{I}, \hat{D}) & s = 0 \\ e_d^1(\mathbf{x}_m^t, d_i; \hat{I}, \hat{D}) & s = 1 \end{cases}. \tag{16}$$

Since L_d and L_g are both multiplication of color and geometry coherence measures in the same form, e_d^0 and e_d^1 can compete fairly to infer the true disparity of each pixel.

5.2 Iterative Optimization

With the redefined data term (16), we iteratively minimize Eq. (9) to jointly refine the disparity maps and infer the static/dynamic labeling (the smoothness term remains the same as Eq. (2), except that the optimized variables are newly defined labels instead of disparities). Since the number of new labels doubles, we typically use 70 disparity levels in our experiments due to memory limitations of BP. The scalability problem can be well addressed by using hierarchical BP as in [18]. While refining $\mathbf{L}_m^t$, we fix the disparity values in all other frames. Each pass of optimization starts from frame 1. One pass completes when the disparity and segmentation maps of all the frames in each camera are optimized. A result is shown at Fig. 5. As shown in Fig. 5(e), after one pass of optimization, the disparity maps are greatly improved in both the static and dynamic regions. Furthermore, the estimated disparity maps becomes more temporally consistent, as evidenced in our supplementary video [1].

As shown in Fig. 5(c-d), only using e_d^0 or e_d^1 for depth optimization will cause visible problems. In comparison, our combination of both e_d^0 and e_d^1 can robustly

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

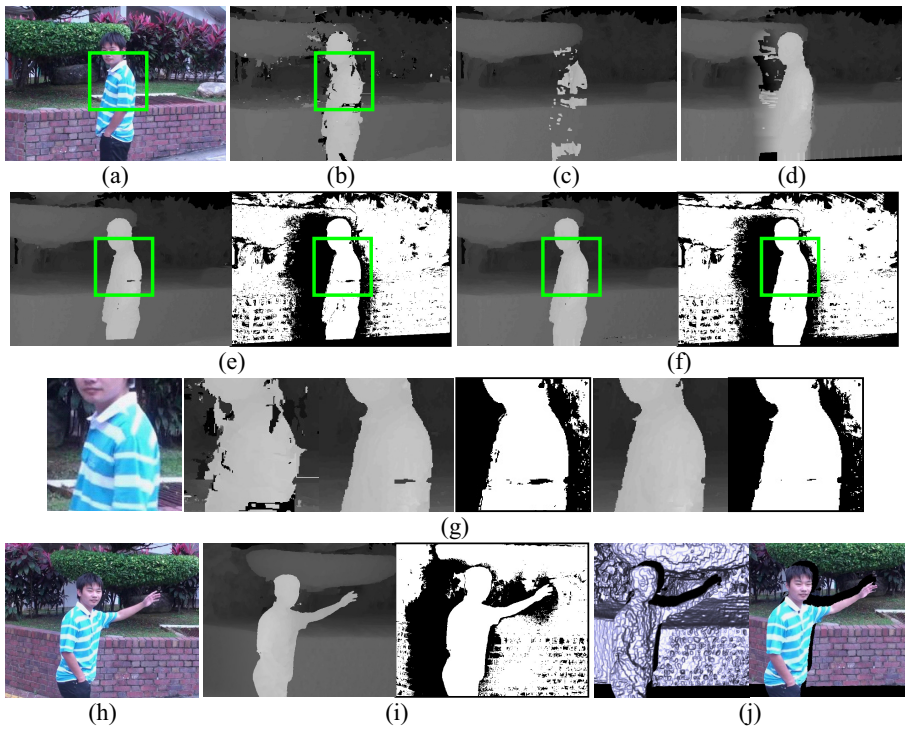

Fig. 5. The results of our spatio-temporal disparity optimization. (a) One source frame. (b) The initialized disparity map. (c) The optimized depth result only using e_d^0 as in [20]. (d) The depth map optimized only using e_d^1 as in [15]. (e) The optimized depth map and bilayer segmentation map by our method after the first iteration, using both e_d^0 and e_d^1. (f) Results after the second iteration of spatio-temporal optimization. (g) Zoomed view of the green rectangles in (a), (b), (e) and (f). (h-i) Another source frame and its spatio-temporal depth optimization result. (j) The 3D surface model reconstructed from the depth map in (i), with texture mapping.

infer the true disparity for both static and dynamic pixels (Fig. 5(e)). There are three marked points A, B and C in Fig. 4(b) and their associated data cost values in (c). Pixel A has a clear minimum cost in e_d^0, and has relatively larger cost in e_d^1. Therefore, A has strong preference to be labeled as static. In comparison, pixel B can reach a strong minimum in e_d^1, and has larger cost in e_d^0. Hence, it is labeled as dynamic. Pixel C has low costs in both e_d^0 and e_d^1. In other words, no matter if it is considered as static or dynamic, its disparity can be inferred reliably. This explains why a large number of background static points are labeled as 'dynamic' by our method. Again, our 'bilayer segmentation' only means to select the right data cost function for a pixel. Another interesting phenomena is that, the occluded pixels are generally labeled as static and the segmentation are quite accurate around discontinuous foreground boundaries. The reason is

Fig. 6. Dynamic 3D reconstruction of the 'Standing' sequences with two input videos. (a) A source frame from the first video. (b-c) The estimated background image and segmentation result by [19]. (d) The refined disparity map based on (c), which contain serious artifacts highlighted in the red rectangles. (e) The refined disparity map by our method, which are accurate in both static and dynamic regions.

that the optical flow estimation is generally difficult for the occluded regions so that the motion-based dynamic data cost e_d^1 will output large values in this case. In contrast, the static data cost e_d^0 are computed by projecting the pixel to temporally neighboring frames with depth, which are robust to occlusions and can output relative small values.

Fig. 5(e) also shows the bilayer segmentation map S_m^t obtained from the optimized $\mathbf{L}_m^t$. As can be seen in Fig. 5(g), most dynamic pixels are correctly labeled and reconstructed except some outliers. These outliers are generally caused by similar foreground and background colors. Fortunately, we can further improve our depths and segmentation in a second spatio-temporal optimization. Specifically, we check the static/dynamic labels of corresponding pixels in temporal neighboring frames to improve static/dynamic segmentation and depth estimation. We track a pixel by optical flow to its 10 nearest neighboring frames and compute the following dynamic-pixel probability:

$$P_d(\mathbf{x}_m^t) = \frac{\sum_{t' \in N(t)} S_m^{t'}(\mathbf{x}_m^t + O_m^{t \to t'}(\mathbf{x}_m^t))}{|N(t)|}. \tag{17}$$

If the dynamic-pixel probability is large, we tend to believe that $\mathbf{x}_m^t$ is dynamic. Otherwise, it is more likely to be static. We incorporate P_d into the second pass of data cost computation and redefine e_d as:

$$e_d(\mathbf{x}_m^t, l_i^s; \hat{I}, \hat{D}) = \begin{cases} (1 + P_d(\mathbf{x}_m^t)) e_d^0(\mathbf{x}_m^t, d_i; \hat{I}, \hat{D}) & s = 0 \\ (2 - P_d(\mathbf{x}_m^t)) e_d^1(\mathbf{x}_m^t, d_i; \hat{I}, \hat{D}) & s = 1 \end{cases}. \tag{18}$$

When $P_d < 0.5$, e_d^0 has smaller weight than e_d^1, so that $\mathbf{x}_m^t$ tends to be labeled as static. Otherwise, we tend to label $\mathbf{x}_m^t$ as dynamic. After the second pass, the incorrectly estimated disparity and static/dynamic labels are successfully corrected as shown in Fig. 5(f).

6 Experimental Results

We have performed experiments on several examples with source videos taken by $2 \sim 3$ handheld cameras. We used Sony HDR-XR550 cameras to capture the

Fig. 7. Dynamic 3D reconstruction results of the 'Bear' sequences (two videos). Two selected frames from the first video and their depth results.

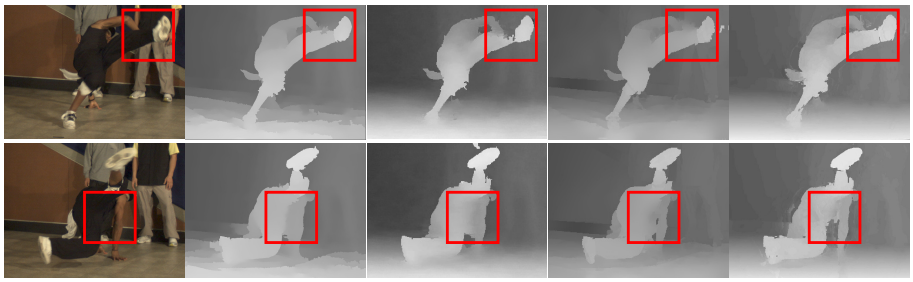

Fig. 8. Comparison of our method to [23], [9] and [15]. (a) Two selected source frames from the third source video of the Microsoft Breakdancing dataset [23]. (b) Microsoft results [23]. (c) The results of [9]. (d) The results of [15]. (e) Our results.

videos. Instead of using genlock, we blinked a flashlight when capturing began and then synchronized the videos according to the blinking timestamp. All our experiments are conducted on a desktop PC with Intel 4-Core 2.83GHz CPU. For a sequence with resolution 960×540, depth initialization takes 172 seconds per frame, and disparity optimization requires 81 seconds per frame, which is more efficient than the methods of [9, 15]. Fig. 6 shows our results on the 'Standing' sequences, which capture a boy standing and moving slightly. Only part of the body is moving in this example, which makes accurate background estimation and static/dynamic segmentation difficult with the traditional methods such as [19]. As shown in Fig. 6(b), the background is almost occluded in all the frames by the dynamic foreground, and can not be well estimated. Hence, the bilayer segmentation result by [19] is not very accurate as shown in Fig. 6(c). If we use different data cost functions to optimize static and dynamic pixels with this inaccurate segmentation result (e_d^0 for static pixels, and e_d^1 for dynamic ones), the refined depth map will contain serious artifacts as shown in Fig. 6(d). In contrast, our method does not rely on precise bilayer segmentation. By spatio-temporal optimization using two kinds of data cost functions, our method cannot only achieve high-quality dynamic depth maps for both static and dynamic pixels, but also preserve accurate depth boundaries without explicit bilayer segmentation as shown in Fig. 6(e).

Another challenging example is provided in Fig. 7, capturing a crawling bear. Our method can also handle multiple dynamic objects, as shown in Fig. 1. We also experimented on the Microsoft Research Breakdancing dataset [23]. Fig. 8 shows the comparison of our results with Microsoft results [23], [9] and [15]. The floor reconstructed in [23] and [15] and the body of the dancer reconstructed in [9] both had visible artifacts, as highlighted in the red rectangles. In comparison, our method generated better results. Please refer to our supplementary material and video for the complete frames and more results.

7 Conclusion and Discussion

In this paper, we propose a novel dense depth recovery method for dynamic scenes captured by multiple handheld cameras. We first use the synchronous images in the same time instance from different cameras to compute the initial depth maps. Then, by combining two different kinds of spatio-temporal constraints, depth optimization and bilayer segmentation are performed simultaneously to achieve a high-quality dense depth estimation.

The accuracy of the optical flow will influence the quality of depth recovery, especially when the motions of moving objects are very large. Severely erroneous optical flow estimation will involve incorrect color and geometry coherence constraints in temporal frames, which harms the depth refinement. Fortunately, this problem can be alleviated by backward verification of optical flow error to discard inaccurate optical flow estimation. How to more effectively solve this problem remains to be our future work.

Acknowledgements: We would like to thank Dr. Xun Cao for helping compute the results of Microsoft Breakdancing dataset by running [15]. This work is supported by the 973 program of China (No. 2009CB320802), National Science and Technology Support Plan Project (No. 2012BAH35B02), NSF of China (No. 60903135), the Specialized Research Fund for the Doctoral Program of Higher Education of China (No. 20110101130011), the Fundamental Research Funds for the Central Universities, and the Singapore MOE grant R-263-000-620-112.

References

1. de Aguiar, E., Stoll, C., Theobalt, C., Ahmed, N., Seidel, H.P., Thrun, S.: Performance capture from sparse multi-view video. ACM Transactions on Graphics 27, 98:1–98:10 (August 2008)
2. Ballan, L., Brostow, G.J., Puwein, J., Pollefeys, M.: Unstructured video-based rendering: interactive exploration of casually captured videos. ACM Transactions on Graphics 29, 87:1–87:11 (July 2010)
3. Comaniciu, D., Meer, P., Member, S.: Mean shift: A robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence 24, 603–619 (2002)
4. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient belief propagation for early vision. International Journal of Computer Vision 70(1), 41–54 (2006)

5. Goldlücke, B., Magnor, M.A.: Joint 3D-reconstruction and background separation in multiple views using graph cuts. In: CVPR. vol. 1 (2003)
6. Guillemaut, J.Y., Kilner, J., Hilton, A.: Robust graph-cut scene segmentation and reconstruction for free-viewpoint video of complex dynamic scenes. In: ICCV. pp. 809–816 (2009)
7. Hasler, N., Rosenhahn, B., Thormählen, T., Wand, M., Gall, J., Seidel, H.P.: Markerless motion capture with unsynchronized moving cameras. In: CVPR. pp. 224–231 (2009)
8. Larsen, E.S., Mordohai, P., Pollefeys, M., Fuchs, H.: Temporally consistent reconstruction from multiple video streams using enhanced belief propagation. In: ICCV. pp. 1–8 (2007)
9. Lei, C., Chen, X.D., Yang, Y.H.: A new multiview spacetime-consistent depth recovery framework for free viewpoint video rendering. In: ICCV. pp. 1570–1577 (2009)
10. Liu, C.: Beyond pixels: exploring new representations and applications for motion analysis. Ph.D. thesis, Massachusetts Institute of Technology (May 2009)
11. Seitz, S.M., Curless, B., Diebel, J., Scharstein, D., Szeliski, R.: A comparison and evaluation of multi-view stereo reconstruction algorithms. In: CVPR. vol. 1, pp. 519–528 (2006)
12. Tao, H., Sawhney, H.S., Kumar, R.: Dynamic depth recovery from multiple synchronized video streams. In: CVPR. pp. 118–124 (2001)
13. Tola, E., Lepetit, V., Fua, P.: Daisy: An efficient dense descriptor applied to wide-baseline stereo. IEEE Transactions on Pattern Analysis and Machine Intelligence 32(5), 815–830 (2010)
14. Torresani, L., Hertzmann, A., Bregler, C.: Nonrigid structure-from-motion: Estimating shape and motion with hierarchical priors. IEEE Transactions on Pattern Analysis and Machine Intelligence 30(5), 878–892 (2008)
15. Yang, M., Cao, X., Dai, Q.: Multiview video depth estimation with spatial-temporal consistency. In: BMVC (2010)
16. Yang, W., Zhang, G., Bao, H., Kim, J., Lee, H.Y.: Consistent depth maps recovery from a trinocular video sequence. In: CVPR. pp. 1466–1473 (2012)
17. Zhang, G., Dong, Z., Jia, J., Wong, T.T., Bao, H.: Efficient non-consecutive feature tracking for structure-from-motion. In: ECCV (5). pp. 422–435 (2010)
18. Zhang, G., Jia, J., Bao, H.: Simultaneous multi-body stereo and segmentation. In: ICCV. pp. 826–833 (2011)
19. Zhang, G., Jia, J., Hua, W., Bao, H.: Robust bilayer segmentation and motion/depth estimation with a handheld camera. IEEE Transactions on Pattern Analysis and Machine Intelligence (99), 603–617 (2011)
20. Zhang, G., Jia, J., Wong, T.T., Bao, H.: Consistent depth maps recovery from a video sequence. IEEE Transactions on Pattern Analysis and Machine Intelligence 31(6), 974–988 (2009)
21. Zhang, G., Qin, X., Hua, W., Wong, T.T., Heng, P.A., Bao, H.: Robust metric reconstruction from challenging video sequences. In: CVPR (2007)
22. Zhang, Y., Kambhamettu, C.: Integrated 3D scene flow and structure recovery from multiview image sequences. In: CVPR. vol. 2, pp. 674–681 (2000)
23. Zitnick, C.L., Kang, S.B., Uyttendaele, M., Winder, S., Szeliski, R.: High-quality video view interpolation using a layered representation. ACM Transactions on Graphics 23, 600–608 (August 2004)