

Dynamic Depth Recovery from Unsynchronized Video Streams

Chunxiao Zhou and Hai Tao

Department of Computer Engineering
University of California, Santa Cruz, CA 95064
{chunxiao, tao}@soe.ucsc.edu

Abstract

In this paper, we propose an algorithm for estimating dense depth information of dynamic scenes from multiple video streams captured using unsynchronized stationary cameras. We solve this problem by first imposing two assumptions about the scene motion and the temporal offset between cameras. The motion of a scene is described using a local constant velocity model and the camera temporal offset is assumed to be constant within a short of period of time. Based on these models, geometric relations between the images of moving scene points, the scene depth, the scene motions, and the camera temporal offset are investigated and an estimation method is developed to compute the camera temporal offset. The three main steps of the proposed algorithm are 1) the estimation of the temporal offset between cameras, 2) the synthesis of synchronized image pairs based on the estimated camera temporal offset and optical flow fields computed in each view, and 3) the stereo computation based on the synthesized synchronous image pairs. The proposed algorithm has been tested on both synthetic data and real image sequences. Promising quantitative and qualitative experimental results are demonstrated in the paper.

1 Introduction

When a dynamic scene is imaged in multiple synchronized video streams from different viewpoints, depth information can be computed using stereo algorithms at any time instant. For such configurations, dynamic stereo algorithms have been recently developed to take advantage of the temporal coherency of the scene motion to achieve more robust depth estimation [4],[10],[11],[13],[14]. However, the requirement of synchronization among cameras greatly limits the use of these algorithms for many applications where synchronization is difficult to achieve, for technical or economic reasons. Being able to compute depth from unsynchronized video streams will significantly reduce the complexity of the video capture system and benefit many applications ranging from home video editing to 3D reconstruction in visual sensor networks. The problem we attempt to solve in this paper is: when a dynamic scene is captured by multiple

stationary *unsynchronized* cameras, how to estimate the depth information of the dynamic scene.

When cameras are synchronized, if a 3D scene point is observed in two views at a certain time instant, the scene point should be at the intersection of the two rays formed by the images points and camera centers. This is known as the triangulation process in the stereo literature (see [6] for a nice review of early stereo algorithms). The remaining problem is how to find the correct correspondences across views, a problem that most stereo algorithms are designed to solve [2],[3],[7],[8].

When the cameras are not synchronized, the problem becomes more complicated. The triangulation process is no longer valid because the same scene point observed in the two views may be imaged at different time instants. When the scene point is moving, the two image rays in general will not intersect (See Figure 1 for an illustration). An implication of this phenomenon is that in theory, if a scene point can move at arbitrarily speed and at any direction, then the stereo computation from unsynchronized video streams is an ill-posed problem. As illustrated in Figure 1, there are infinitely many solutions formed by picking an arbitrary point on each image ray.

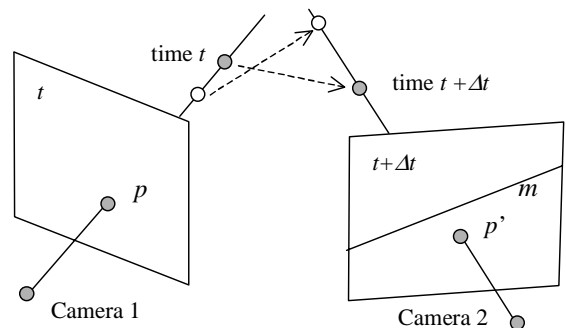


Figure 1. Images of a moving 3D scene point in two unsynchronized views.

In this paper, we propose to solve this problem by first imposing two reasonable assumptions regarding the scene motions and temporal offset between different cameras. We then develop a simple algorithm that first estimates the temporal offset and convert the unsynchronized video sequences into synchronized ones. Once this

conversion is completed, traditional synchronized stereo algorithms are employed to compute the depth information of the dynamic scenes. Our paper is related to and much inspired by the pioneering work of Caspi and Irani [5], in which two algorithms were developed to achieve spatio-temporal alignment of video sequences using a simple parametric model to describe the transformation between two views and an additional parameter to describe the temporal offset between cameras. However, in the unsynchronized stereo problem, the transformation between different views is determined not only by the camera parameters, but also by the depth values of individual pixels. A simple, yet promising method is proposed in this paper to solve this problem. In another related paper, Avidan and Shashua [1] proposed a method called *trajectory triangulation* for computing the 3D positions of moving scene points from multiple images of the points. Linear and conic motion trajectory models were considered in their work. The proposed algorithm is also closely related to recent work on dynamic stereo computation where the recovery of the depth information of dynamic scenes is concerned [4],[10],[11],[13],[14]. However, to the best of our knowledge, all existing algorithms are based on the synchronized camera capture system and we are not aware of any previous work on recovering depth information of dynamic scenes using unsynchronized cameras.

The paper is organized as follows. Section 2 explains the main ideas of the proposed approach. Section 3 describes the implementation details. In Section 4, quantitative and qualitative experimental results on synthetic and real data are demonstrated and analyzed. Discussions and conclusions can be found in Section 5.

2 The approach

2.1 The basic geometric constraint

As discussed in the previous section, if the scene motion is arbitrary, there are infinitely many solutions to the unsynchronized stereo problem. However, due to the physical laws that govern the dynamics of scene objects, motion of these objects are generally smooth during a short period of time. If we assume the motion of a scene point can be locally approximated using a linear motion, a simple geometric constraint can be developed for estimating 3D positions of scene points using its four images in the two views. As shown in Figure 2, suppose a scene point undergoing linear motion is imaged at time instants t_1, t_2, t_3, t_4 in the two views as a, b, c' , and d' . In the general case, the linear trajectory can be found by intersecting the plane ac_1c and the plane $b'c_2d'$, where c_1 and c_2 are the two camera centers. The 3D positions A, B, C , and D of the scene point can be

found by intersecting the linear trajectory with c_1a, c_1c, c_2b' , and c_2d' . For the special case that the two planes are coplanar, any lines in the epipolar plane is a solution. Therefore additional constraints are needed to find a unique solution.

The geometric constraint resulted from the linear motion assumption implies a very simple depth computation algorithm. However, this requires point correspondences in four images. When a pixel lies in an area without prominent features, this is a difficult task because any estimation error in the single view correspondence, i.e. optical flow fields, or the cross view correspondences will propagate into the final solution. In the following section, we propose a simplified algorithm in which the 4-correspondence problem is avoided.

For a pair of unsynchronized video cameras, if their frame rates are the same and they are roughly aligned, then a very simple geometric constraint can be derived for the four images of a moving scene point if we impose two simple assumptions. The first assumption is that the scene points moves at a constant velocity in a short period of time. The second assumption is that the camera temporal offset Δt is a constant over a short period of time. It should be noticed that, the assumption of constant Δt is only local and gradual change of Δt is allowed over a longer period of time. The assumption of linear motion with constant velocity is the simplest scene motion model, it is also possible to use more complex model such as a constant acceleration model.

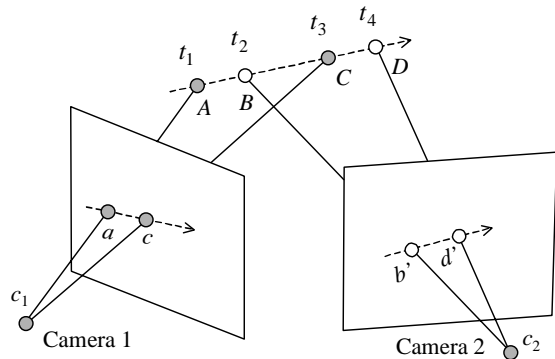


Figure 2. Determine the linear trajectory of the scene point by intersecting two planes ac_1c and $b'c_2d'$.

2.2 Conversion to synchronized videos

Instead of solving the 4-correspondence problem for every scene point, we propose to first estimate the camera temporal offset Δt using prominent feature points and then synthesize synchronized videos. A standard stereo algorithm is then applied to compute the depth maps using the synthesized synchronized sequences.

2.2.1 Computing the camera temporal offset

Camera temporal offset Δt is estimated using a sparse set of 4-correspondences. As illustrated in Figure 3, when the linear motion model is assumed, the cross ratio of four 3D points A , B , C , and D is an invariance under perspective projection. With the constant velocity model and the constant camera offset, the cross ratio is

$$\frac{AB/BD}{AC/CD} = \frac{\Delta t/1}{1/\Delta t} = \Delta t^2 \quad (1)$$

Since the cross ratio is invariant under perspective projection, therefore,

$$\frac{ab/bd}{ac/cd} = \Delta t^2 \quad (2)$$

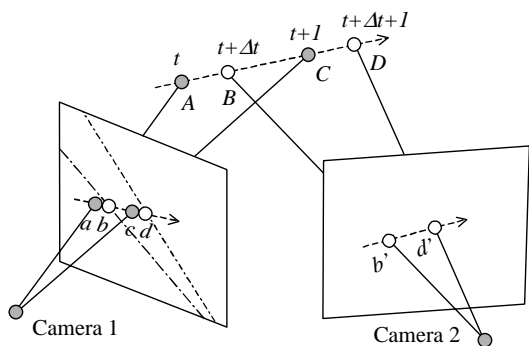


Figure 3. The relation between the camera temporal offset Δt and the cross ratio.

This analysis implies an algorithm for estimating Δt , which is described as follows.

Step 1: Compute 4-correspondences based on image features

Step 2: For each 4-correspondence, find the cross ratio in one of the views, e.g. left view. This can be achieved by first deriving the epipolar line using

$$\begin{aligned} b &= (a \times c) \times (F^T b') \\ d &= (a \times c) \times (F^T d') \end{aligned} \quad (3)$$

where F is the fundamental matrix between the two views. Then the intersections of these epipolar lines and the line formed by the two points in the first view are computed. The cross ratio is then computed as (2)

Step 3. If $\frac{ab/bd}{ac/cd} < 0$ then discard it. Otherwise Δt is computed as

$$\Delta t = \begin{cases} \sqrt{\Delta t^2} & \langle ab, ac \rangle \geq 0 \\ -\sqrt{\Delta t^2} & \langle ab, ac \rangle < 0 \end{cases} \quad (4)$$

where $\langle \cdot, \cdot \rangle$ is the dot product operator.

It should be mentioned that this method does not work if the four 3D points A , B , C , D and the camera centers are coplanar because then the intersections b and d cannot be found (see Figure 4). In that case, there are infinitely many solutions, both for Δt and the trajectory. This can be shown geometrically or by counting the number of unknowns and number of constraints. A special case is of particular interest and will be discussed briefly here. Suppose the two cameras are arranged in a standard stereo setup with horizontal epipolar lines and a baseline of T . Figure 5 shows a scene points moving in a direction parallel to the camera baseline, with depth d , temporal camera offset Δt , and motion v . It can be shown that any positive value α , a solution in the form of $(\alpha d, \alpha v, (1-\alpha)T + \alpha \Delta t v)$ generates exactly the same 4-correspondence. We call this phenomenon the depth-motion-time ambiguity in unsynchronized stereo.

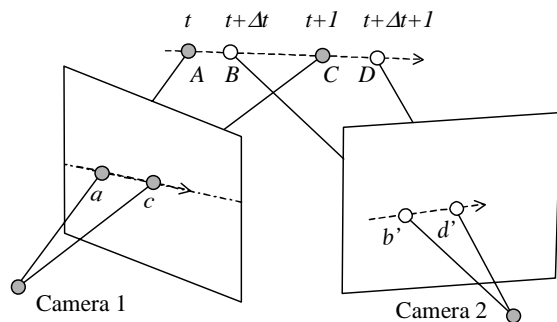


Figure 4. When the 3D scene motion and the camera centers are coplanar, Δt cannot be computed using Equation (2).

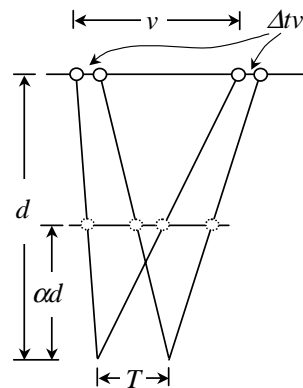


Figure 5. The depth-motion-time ambiguity in unsynchronized stereo.

2.2.2 Synthesis of synchronized image pairs

Once the temporal offset Δt between cameras is estimated, in the second view, images are synthesized to form synchronized video sequences with respect to the image sequence in the reference view. This is accom-

plished by first estimating the optical flow fields between consecutive frames in the second view and then warped the images to form new images at time instants in-between the frames. Figure 6 illustrates this process. Let $\mathbf{I}(t+\Delta t)$ and $\mathbf{I}(t+\Delta t+1)$ denote the image at time instants $t+\Delta t$ and $t+\Delta t+1$. First the optical flow field from $\mathbf{I}(t+\Delta t)$ to $\mathbf{I}(t+\Delta t+1)$ and the optical flow from $\mathbf{I}(t+\Delta t+1)$ to $\mathbf{I}(t+\Delta t)$ are computed. These optical flow fields are denoted as $f_1 = f(\mathbf{I}(t+\Delta t) \rightarrow \mathbf{I}(t+\Delta t+1))$ and $f_2 = f(\mathbf{I}(t+\Delta t+1) \rightarrow \mathbf{I}(t+\Delta t))$. Images $\mathbf{I}(t+\Delta t)$ and $\mathbf{I}(t+\Delta t+1)$ are warped to the new time instant t using $(1-\Delta t) \cdot f_1$ and $\Delta t \cdot f_2$, respectively. The resultant two warped images are the predicted $t+1$ image using $\mathbf{I}(t+\Delta t)$ and $\mathbf{I}(t+\Delta t+1)$. These two images are combined using blending factors Δt and $1-\Delta t$. In summary, the image at time t is synthesized as

$$\mathbf{I}(t+1) = \Delta t \cdot W(\mathbf{I}(t+\Delta t), (1-\Delta t) \cdot f_1) + (1-\Delta t) \cdot W(\mathbf{I}(t+\Delta t+1), \Delta t \cdot f_2) \quad (5)$$

where $W(\mathbf{I}, f)$ is a forward warping function that warps image \mathbf{I} using the optical flow f .

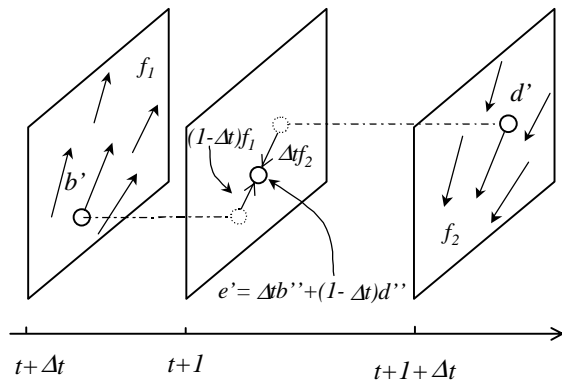


Figure 6. Synthesize the image at $t+1$ from images at $t+\Delta t$ and $t+\Delta t+1$.

It should be mentioned that an approximation is used in this image synthesis procedure. Ratio between segments on a 3D linear trajectory, which is $1-\Delta t : \Delta t$ in our case, is not a projective invariance in general. The ratio between projected 2D segments is not exactly $1-\Delta t : \Delta t$. However, when the inter-frame scene motion is small with respect to the object-to-camera distance, it is a very good approximation.

2.2.3 Synchronous stereo

Once the image at time $t+1$ is synthesized in the second view, traditional stereo algorithms can be used to compute the depth map.

Careful readers may have noticed that the depth map for each time instant is computed independently. The 3D motions of scene points are not computed in the proposed algorithm. In other words, there is no correspondence between depth maps computed at different time instants. This is exactly why the difficult 4-correspondence problem is avoided. In our approach, the only requirement is that the synthesized images should be similar to the real ones. This can be achieved even if the optical flow is wrong. This phenomenon is mostly noticeable in textureless regions.

3 Implementation

The proposed unsynchronized stereo algorithm has been implemented. Details of the implementation and various problems encountered are discussed in the following subsections.

3.1 Camera calibration and pose estimation

In the proposed algorithm, we assume the intrinsic and extrinsic camera parameters are known. In our implementation, we use Zhang's algorithm [16], which uses planar calibration object, to compute the intrinsic camera parameters. Since the algorithm also computes the relative 3D pose of each camera with respect to the calibration object, it is possible to use the same algorithm to compute the relative poses between cameras. However, when the cameras are unsynchronized, this method is not applicable. Instead, we compute the camera pose using static scene without any moving objects [15]. When the scene lacks natural 3D features, additional 3D objects are inserted to provide enough image features for computing camera poses.

In a real system, it is possible to simplify the above setup procedure. To obtain camera intrinsic parameters, self-calibration methods can be employed to avoid the use of calibration objects. The requirement of a complete static scene in the calibration and pose estimation stages can be dropped by using background estimation techniques to detect and remove moving foreground objects in each frame. It is also possible to estimate camera poses using feature points on the moving, which itself is an open research problem.

3.2 Coarse temporal offset between cameras

To compute the accurate temporal offset between cameras, our algorithm requires the unsynchronized video sequences to be roughly aligned. When the two cameras are close to each other, the image transformation between the two views can be approximated by an affine transform. Under such condition, the method developed by Caspi and Irani [5] can be used for computing the initial temporal alignment. Another possible method is to align the sequence by detecting unique movement or change in the dynamic scenes, automatically or manu-

ally. Our current experimental results are obtained by manually aligning the two video sequences up to an accuracy of several frames.

3.3 Detecting features and 4-correspondences

In order to compute the camera temporal offset, we need to extract 4-correspondences on the moving object to compute the cross ratio. Zhang’s algorithm [15] for detecting corresponding feature points across two views were applied and a post-processing step that links pair-wise correspondences into 4-correspondences has been developed. There are multiple ways of choosing frames for computing pair-wise correspondences. If we denote the first frame and the second frame in camera 1 as v_1 and v_3 and denote the first and the second frames in camera 2 as v_2 and v_4 . We can first find pair-wise correspondences for v_1v_2 , v_2v_3 , v_3v_4 and then link them to form 4-correspondences. Another possibility is that we instead find pair-wise correspondences for image pairs v_1v_3 , v_2v_4 , and v_1v_2 . Experiments show that the first method is more reliable. This is probably because the pair-wise feature detection algorithms tend to find similar correspondences when all the transforms across views are similar. It is also possible to modify Zhang’s algorithm to find 4-correspondences directly, which hopefully will make the point tracks more reliable.

Another issue encountered in our implementation is that the feature points need to be located at sub-pixel accuracy for computing Δt . This is accomplished by locally fitting the images using a quadratic surface and finding the zero-curvature point. This step is important because the motion of scene points in each view is relatively small between two consecutive frames. Errors in feature point positions will result in inaccurate temporal camera offset estimation.

3.4 Robust estimation of Δt

Once 4-correspondences are found, they are used for computing camera offset Δt . As mentioned in the previous sections, such feature points should lie on the moving objects and ideally the motion trajectory should be as perpendicular to the epipolar plane as possible. In our implementation, the candidate 4-correspondences are used in the Δt computation if they satisfy the following two conditions:

$$|a-c| > \tau_l \quad \text{and} \quad \theta > \tau_\theta \quad (6)$$

where $|a-c|$ is the motion of the feature points in the first view (Figure 3) and θ is the angle between ac and the epipolar line passing through a . In our implementation, the two thresholds were set to be $\tau_l = 1.5$ pixels and $\tau_\theta = \arctan(0.3)$.

In order to make the estimation more robust, 4-correspondences are collected in a temporal sliding window from time $t-W$ to time $t+W$. All the features in this window are used for estimating Δt . For each 4-correspondence in the sliding window, a candidate Δt can be computed. The RANSAC algorithm is then applied to obtain a robust estimation of the Δt using all candidate Δt ’s. In our implementation, $W = 4$.

3.5 Synthesizing synchronized video streams

New images are synthesized to create synchronized video streams. As described in Section 2.2.2, in the second camera view, images $\mathbf{I}(t+\Delta t)$ and $\mathbf{I}(t+\Delta t+1)$ are warped to the time instant t using $(1-\Delta t) \cdot f_1$ and $\Delta t \cdot f_2$, respectively. Two forward warping steps are required in this process. In our implementation, we approximate this process using a backward warping procedure [12], which uses the optical flow fields $-(1-\Delta t) \cdot f_1$ and $-\Delta t \cdot f_2$. This is a good approximation to the forward warping method when the motion is not too large.

3.6 Depth computation

Once the synthetic synchronized video stream is rendered, traditional synchronized stereo algorithm can be applied to compute depth information. A stereo algorithm based on our previous work has been used in our implementation [9]. Recently published dynamic stereo algorithms can also be applied to further take advantage of the temporal coherency of dynamic scenes.

4 Experimental results

The proposed algorithm has been tested on synthetic and real image sequences. A single set of parameters was used to obtain all the experimental results shown in this paper. Quantitative as well as qualitative results will be shown in this section.

4.1 Synthetic sequence - *Painting*

A synthetic sequence has been generated using 3ds Max™ to obtain quantitative results on the performance of the proposed algorithm. The dynamic scene consists of a part of the interior of a room and a moving painting in a frame. Images as well as depth maps were synthesized at two viewpoints mimicking a standard stereo setup. To generate unsynchronized video streams from the two viewpoints, we first render the scene at high frame rate from both viewpoints. The rendered sequences are then temporally sub-sampled with the same rate but with different offset. For example, in order to achieve $\Delta t = 1/3$, we select the $3n$ frames from the first sequence, but select the $3n+1$ frames from the second sequence. We show an example with $\Delta t = 0.5$. This is the most difficult case for synthesizing synchronized

sequence because the new view is the farthest from the nearest real frame. Figure 7 shows two images in the first view (left view) and the corresponding real disparity map, which are converted from the depth map generated by 3ds Max™. The floating-point disparity values range approximately from 3 to 13 pixels in this sequence.

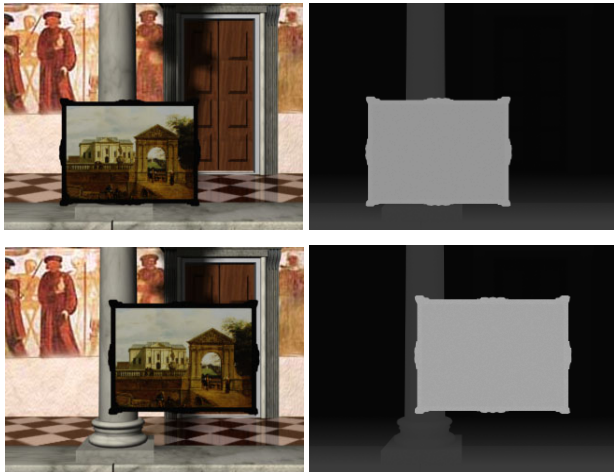


Figure 7. Frames 30 and 50 (from top to bottom) in the left view and the corresponding true disparity maps in the *Painting* sequence.

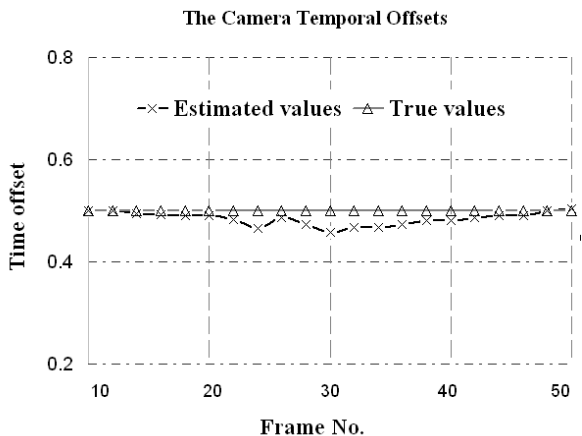


Figure 8. The estimation of Δt in the *Painting* sequence. The true value is $\Delta t = 0.5$.

Using the method described in Section 3.4, the camera temporal offset Δt is estimated for every frame based on features collected in the neighboring 9 frames. The results are shown in Figure 8. The estimation is fairly accurate, especially when the vertical motion, which is large around the beginning and the end of the sequence.

Using the estimated camera temporal offset, synchronized images are synthesized. The real images in the second view can be rendered using 3ds Max™ to serve as the ground truth for testing how well the image synthesis process performs. In Figure 9(a-b), the

synthesized frame 40 in the right view and the real frame 40 in the second view are shown. Figure 9(c) shows the absolute difference between the two images and Figure 9(d) shows one of the optical flow fields that were used for image warping. The quality of the warped image is very good except that some errors occur in the occluded regions and around the depth boundaries.

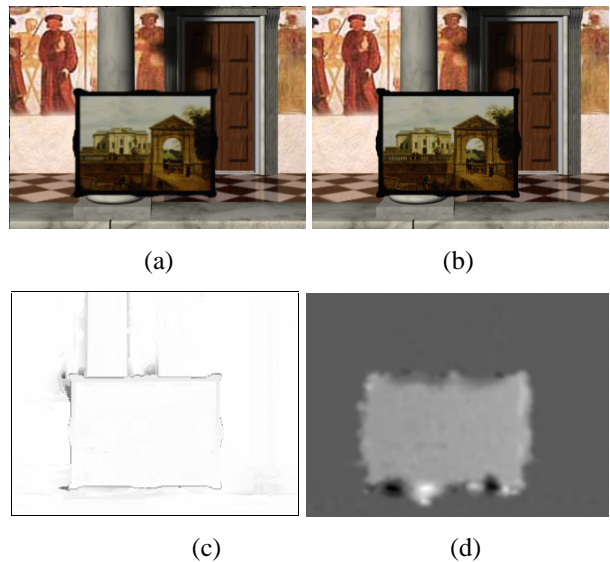


Figure 9. (a) The synthetic frame 40 in the right view, (b) the real frame 40 in the right view, (c) the absolute difference between the two images, with a average value of 0.293 and the maximum value is 13.04, (d) one of the optical flow fields (the x component) used in the image warping.

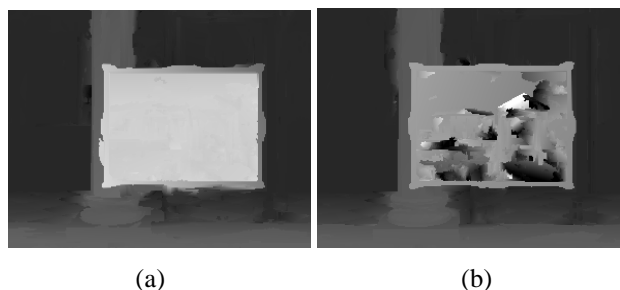


Figure 10. Disparity maps computed in frame 50 computed (a) using the proposed algorithm and (b) using the original unsynchronized images.

Disparity maps are computed using the synthetic synchronized images sequences. Some of the results are shown in Figure 10. The estimated disparity values are compared with the ground truth in each frame and the mean square errors are computed both for the whole scene and for the moving object. The disparity has also been computed using the same stereo algorithm on the original unsynchronized two sequences and the mean square errors were computed as well. The MSE values for both experiments are shown in figure 11. It can be

observed that the proposed algorithm dramatically improves the results. The MSE value for the moving part is 0.746 pixels in average using our algorithm. Since our depth algorithm uses discrete disparity values with an interval of 0.25 pixels, the MSE values also including rounding errors. In comparison, when the disparity is computed directly from the unsynchronized sequences, the average MSE is 9.98 pixels.

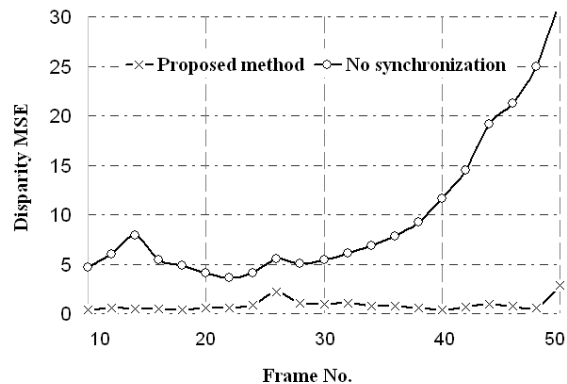


Figure 11. Average disparity values on the moving part of the scene

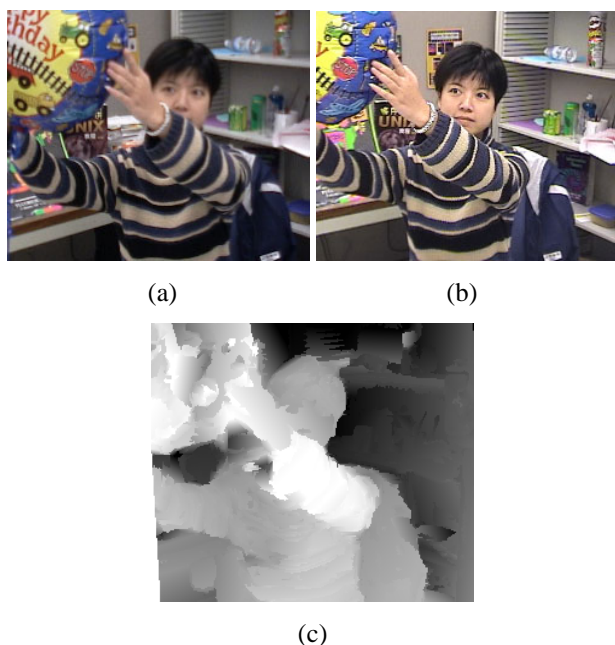


Figure 12. The *Balloon* sequence: (a) the original frame 50 in the first view, (b) the synthesized synchronized view at the same time instant, and (c) the corresponding disparity map in the left view.

4.2 Real sequences – *Balloon* and *Walking*

Two different consumer camcorders, one Canon Elura M20 and one Sony PC-110, were used to take the two unsynchronized sequences in this experiment. The cameras were mounted on tripods at roughly the same

height. We first calibrated the cameras individually using a planar pattern. The camera distortion was then estimated and compensated. The relative pose between the two cameras was estimated using Zhang’s algorithm [15] and the fundamental matrix between the two cameras was derived. The difference between this experiment and the previous one is that the ground truth values are not available for the camera temporal offset, the real synchronized views in the second camera, and the disparity maps. In Figure 12(a), a video frame in the first view is shown. Figure 12(b) shows the synthetic image in the other view. Figure 12(c) shows the corresponding disparity maps using our algorithm. Figure 13 shows the estimates of the camera temporal offset, with an average value of 0.64.

The proposed algorithms has also been tested on outdoor scenes. The camera setup is similar to the one used in the *Balloon* sequence except that the two cameras are at different heights to avoid the coplanar configuration discussed in Section 2.2.1. In Figure 14, some of the initial frames and the computed depth maps are shown. Figure 15 shows the estimates of the camera temporal offset, with an average value of -0.077.

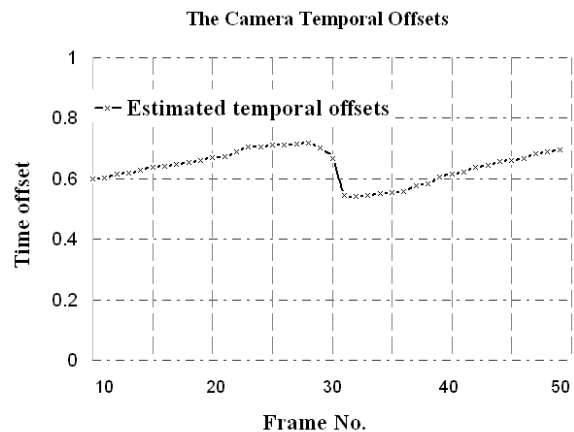


Figure 13. The estimated Δt in the *Balloon* sequence.

5 Discussions and conclusions

We propose in this paper a simple method to compute depth information from unsynchronized video streams. The algorithm is based on two reasonable assumptions regarding the scene motion and the camera temporal offset. The proposed algorithm avoided the difficult 4-correspondence problem by computing depth from synthesized synchronized views. A feature-based method has been developed for robust estimation of camera temporal offset. Promising experimental results have been obtained with a relatively straightforward implementation. During the development of this algorithm, we noticed several new exciting research problems in unsynchronized stereo computation. Some of them

include the camera calibration and pose estimation with moving objects in the scenes, the robust estimation of camera temporal offset when very few features can be detected on the moving objects, and the stereo computation using unsynchronized moving cameras.



Figure 14. The *Walking* sequence. Left column: the original frames. Right column: the estimated depth maps.

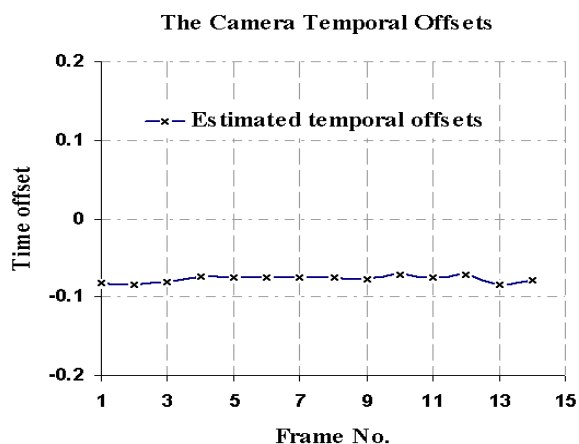


Figure 15. The estimated Δt in the *Walking* sequence.

6 Acknowledgments

We would like to thank Richard Szeliski for comments on the asynchronous stereo problem and Brendan Frey for the very helpful initial discussion.

References

- [1] S. Avidan and A. Shashua, "Trajectory triangulation: 3D reconstruction of moving points from a monocular image sequence," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 22(4), pp. 348-357, 2000.
- [2] P. N. Belhumeur, "A Bayesian-approach to binocular stereopsis," *Int. Journal of Computer Vision*, vol. 19, no. 3, pp. 237-260, August 1996.
- [3] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," in *Proc. Int. Conf. on Computer Vision (ICCV'99)*, Sept. 1999.
- [4] R. L. Carceroni and K. N. Kutulakos, "Scene capture by surfel sampling: from multi-view streams to non-rigid 3D motion, shape and reflectance," in *Proc. Int. Conf. on Computer Vision (ICCV'01)*, July 2001.
- [5] Y. Caspi and M. Irani, "Spatio-temporal alignment of sequences," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 11, November 2002.
- [6] U. R. Dhond and J. K. Aggarwal, "Structure from stereo: a review," *IEEE Transactions on System, Man, and Cybernetics*, vol. 19, no. 6, pp. 1489-1510, 1989.
- [7] K. N. Kutulakos and S. M. Seitz, "A theory of shape by space carving," in *Proc. Int. Conf. on Computer Vision (ICCV'99)*, pp. 307-314, 1999.
- [8] D. Scharstein and R. Szeliski, "Stereo matching with nonlinear diffusion," *International Journal of Computer Vision*, 28(2):155-174, July 1998.
- [9] Hai Tao and Harpreet S. Sawhney, "Global matching criterion and color segmentation based stereo," in *Proc. Workshop on the Application of Computer Vision (WACV2000)*, pp. 246-253, Dec. 2000.
- [10] Hai Tao, Harpreet S. Sawhney, Rakesh Kumar, "Dynamic depth recovery from multiple synchronized video sequences," in *Proc. IEEE conf. on Computer vision and Pattern Recognition 2001 (CVPR01)*, 2001.
- [11] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade, "Three-dimensional scene flow," in *Proc. Int. Conf. on Computer Vision*, pp. II-722 - 729, Sept. 1999.
- [12] G. Wolberg, *Digital Image Warping*, Wiley-IEEE Press, July 1990.
- [13] G. Young and R. Chellappa, "3-D motion estimation using a sequence of noisy stereo images: models, estimation and uniqueness results," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, pp.735-759, 1990.
- [14] Y. Zhang and C. Kambhampettu, "Integrated 3D scene flow and structure recovery from multiview image sequences," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'00)*, pp. II-674-681, South Carolina, June 2000.
- [15] Z. Zhang, R. Deriche, O. Faugeras, Q.-T. Luong, "A Robust Technique for Matching Two Uncalibrated Images Through the Recovery of the Unknown Epipolar Geometry," *Artificial Intelligence Journal*, Vol.78, pages 87-119, October 1995.
- [16] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 1330-1334, 2000.