University of Nebraska - Lincoln

DigitalCommons@University of Nebraska - Lincoln

Faculty Publications from the Department of
Electrical and Computer Engineering

Electrical & Computer Engineering, Department of

2013

Binocular Video Object Tracking with Fast Disparity Estimation

Yun Ye

University of Nebraska-Lincoln

Song Ci

University of Nebraska-Lincoln

Yanwei Liu

Institute of Acoustics, Chinese Academy of Sciences, China

Haohong Wang

TCL Research America, TCL Corporation

Aggelos K. Katsaggelos

Northwestern University

Follow this and additional works at: <http://digitalcommons.unl.edu/electricalengineeringfacpub>

Part of the [Computer Engineering Commons](#), and the [Electrical and Computer Engineering Commons](#)

Ye, Yun; Ci, Song; Liu, Yanwei; Wang, Haohong; and Katsaggelos, Aggelos K., "Binocular Video Object Tracking with Fast Disparity Estimation" (2013). *Faculty Publications from the Department of Electrical and Computer Engineering*. 322.
<http://digitalcommons.unl.edu/electricalengineeringfacpub/322>

This Article is brought to you for free and open access by the Electrical & Computer Engineering, Department of at DigitalCommons@University of Nebraska - Lincoln. It has been accepted for inclusion in Faculty Publications from the Department of Electrical and Computer Engineering by an authorized administrator of DigitalCommons@University of Nebraska - Lincoln.

Binocular Video Object Tracking with Fast Disparity Estimation

Yun Ye¹, Song Ci^{1,2}, Yanwei Liu², Haohong Wang³, Aggelos K. Katsaggelos⁴

¹Department of Computer and Electronics Engineering, University of Nebraska-Lincoln, USA

²Institute of Acoustics, Chinese Academy of Sciences, China

³TCL Research America, TCL Corporation, USA

⁴Department of Electrical Engineering and Computer Science, Northwestern University, USA

Abstract

This paper presents a binocular PTU (pan-tilt unit) camera video object tracking scheme using the MeanShift algorithm and the runtime disparity estimation. The proposed method is to accommodate the requirement of 3D content generation and accurate tracking in more advanced video surveillance applications. The disparity estimation process for each stereoscopic pair is formulated as an energy minimization problem. The iterative solution procedure is implemented in a coarse-to-fine manner. The estimated disparity is used to scale the tracking window by the MeanShift algorithm, i.e. the size of the tracking area is adjustable according to its inner disparity, and thus the moving object can be better located by the camera. The program maintains the semi-real-time performance and acceptable accuracy as evaluated on a set of standard test data. In our experiment, two PointGrey cameras are controlled through a PTU device. The disparity estimation process on the recorded tracking video (640x480) achieves 6fps on an ordinary PC (2.66GHz CPU, 4GB RAM).

1. Introduction

Traditional monocular tracking methods mainly explore the temporal correlation in one video to detect moving areas [1-4]. With the development of the 3D signal processing technologies, multiview video object tracking is gaining increasing interest. In the tracking algorithm presented in [5], both inter-frame and inter-view correlations are utilized to predict the object's position and speed, using optical flow and disparity estimation. The outdoor tracking algorithm presented in [6] performed ground view alignment using homography; each moving object is detected via background subtraction. These works use fixed cameras and may have limitation in view scope. The post-capture tracking method described in [7] handles the changing camera viewpoint by constructing a panoramic image used for background registration and object detection.

In the video tracking and streaming system introduced in [8], active cameras are adopted for runtime operation. A master camera is manually controlled and other slave cameras will automatically follow. For object tracking with moving cameras, a more practical strategy is to use automatic PTU/PTZ cameras, where the camera projection center is generally unchanged and the retinal plane is capable of angular movement. In this kind of system, the camera control algorithm for the tracking process needs to estimate the angular speed/acceleration of the moving object, and the background alignment in different video frames is required for motion detection. In the PTU camera tracking algorithm proposed by Petrov et al. [4], a linear feedback controller is applied based on the Theory of Lyapunov Stability. The control parameters are updated by object position estimated using the MeanShift [1] algorithm. MeanShift is efficient in locating object position according to the object's color distribution [7], [8]. However, a key problem with this method is the scaling of the tracking area, since the size of the object appears differently as its depth changes. A tentative scheme is suggested in [1] to adjust the tracking region according to the similarity measure. It might have problem if similar color is present around the boundary of the tracking area. The object segmentation based approach is more robust, and time consuming [2]. Yang et al. [3] developed an updating rule for scaling factor by comparing second-order moments between the template and the target, but only small tracking region is tested.

To enable tracking in an unmanned environment, we use a PTU device (Directed Perception D47) to perform camera control. The master camera is able to rotate and its projection center stays unchanged [4]. A slave camera is placed on the flank, and moves along with the master camera, as shown in Figure 1(a). The MeanShift algorithm [1] is adopted in our project for real-time tracking. With binocular video output, we consider the object depth/disparity as a natural and reliable resource for adjusting the tracking window, since disparity contains object position information and it is necessary for multi-

(a) PTU cameras

(b) Tracking procedure

Figure 1: Binocular video object tracking.

view video streaming [8]. The video object tracking procedure is illustrated in Figure 1(b). The estimated disparity values from the stereoscopic image pair are used to adjust the size of the tracking region. The object center is detected using the MeanShift algorithm, and it is provided for PTU control [4] on the master camera.

The implementation of the disparity estimation process is essential for the real-time object tracking. Stereo matching/disparity estimation has been extensively studied as a fundamental vision task. Popular solutions include the local winner-take-all [9-11] and the global MRF optimization [12-14]. Local methods compare matching cost computed within a neighborhood. They are known for their fast implementation, but have difficulty in dealing with ambiguous or similar textures. Global methods are capable of imposing smoothness constraint, such as graph cut [12] and belief propagation [13]. Their occlusion detection ability is impeded in the presence of curved surfaces and the computation is usually very expensive. Smith et al. [14] proposed to perform graph cut optimization on a sparse graph obtained using maximum spanning tree. Local filtering is applied at the finer grid for further refinement. Although this method better detects non-planar surfaces, the process of spanning tree generation and full image filtering are still costly to implement. For our tracking procedure, disparity estimation has to meet the runtime requirement in order to timely adjust the tracking region. While some real-time approach [9] relies on GPU implementation, Geiger et al.

[10] introduced an efficient matching method based on Delaunay triangulation [15]. This method exhibits superior results in less textured areas, and semi-real-time performance is reported. However, the initial supporting points acquired using the local method fall short of resolving spurious matches caused by similar textures, which are commonly encountered in an indoor surveillance environment with reduplicate building structures or wall decorations. This will in turn result in incorrect estimation on the finer grid. Moreover, the disparity values on the finer grid are also estimated using the local method, thus the estimation changes of the neighbors could not be further utilized to estimate the disparity of the current pixel.

To overcome these shortages, we propose to implement global optimization on the initial supporting points. A best disparity for each initial supporting point is selected from multiple candidates through Iterative Conditional Mode (ICM) [16]. The disparity estimation process on the finer grid is formulated as an energy minimization problem. Both the data consistency term and the smoothness term constrain the iterative solution procedure. The program still maintains semi-real-time performance and acceptable quality.

The rest of the paper is organized as follows. Section 2 introduces the binocular MeanShift tracking using the disparity information. Section 3 describes the fast disparity estimation procedure. Section 4 provides the experimental results. And conclusions are drawn in Section 5.

2. Binocular PTU camera tracking

The unique geometric property of a PTU camera model is that the camera projection center remains unchanged while the pan and tilt angles are controllable, as illustrated in Figure 2(a) [4]. The focus F denotes the projection center. The image plane is viewed down along its y axis, and is projected on the X - Y world coordinate plane. α is the angle between the object center and the X axis, θ is the angle between the image center and the X axis, f is the focal length, and x_c is the distance between the projected object center and the image center along the x axis. Only pan control is displayed in the figure. The algorithm applies to tilt control similarly.

The linear feedback controller aims to minimize x_c and the difference between the estimated object speed and the measured object speed. According to the Theory of Lyapunov Stability, the camera angular speed $w_{\theta k}$, the camera angle θ , and the estimated distance x_c are updated at every time instance:

$$w_{\theta k} = \hat{w}_{\alpha k} + p k x_{ck} \quad (1)$$

$$\theta_{k+1} = \theta_k + w_{\theta k} \Delta t_k \quad (2)$$

$$\hat{x}_{ck+1} = f \tan(\hat{\alpha}_{k+1} - \theta_{k+1}) \quad (3)$$

Figure 2: Binocular MeanShift tracking.

where $\hat{w}_\alpha$ is the estimated object angular speed, and $\hat{\alpha}$ is the estimated object angle. p is a parameter used to control the convergence speed in the dynamic system. And Δt_k is the duration of the k -th time interval for the control parameters update.

Once the control parameters are updated, the disparity information for the master camera side is estimated using the method described in Section 3. The camera projection matrix used for the stereo rectification is obtained beforehand through chessboard calibration. After the control parameter update, the MeanShift algorithm is applied to locate the object center in the new frame. The size of the rectangular tracking window is scaled according to

$$l1/l2 = z2/z1 = d1/d2 \quad (4)$$

where $l1$, $l2$ denote the edge length of the tracking window at two consecutive updates, z is the depth of the object, and d is the average estimated disparity for the object region, as shown in Figure 2(b).

3. Fast disparity estimation

According to Bayes' rule, the process of disparity estimation can be formulated as a MAP-MRF problem [13]. For example, in binocular stereo matching, the estimation on the left image from a stereoscopic pair I_1 (left), I_2 (right), is usually considered as a process of minimizing the following energy function,

$$\begin{aligned} E(d(x, y)) &= \iint_{\Omega} F(d) dx dy \\ &= \iint_{\Omega} (\lambda_1 \cdot f_{data}(d) + \lambda_2 \cdot f_{smooth}(d)) dx dy \end{aligned} \quad (5)$$

where $d(x, y)$ is the estimated disparity at pixel (x, y) . λ_1 and λ_2 are the scaling factors. f_{data} and f_{smooth} are distance measures and represent the penalty terms on photo consistency and smoothness. For horizontally rectified images,

$$f_{data}(d(x, y)) = \frac{1}{2} \sum_{(x_k, y_k) \in \mathfrak{N}_1((x, y))} (I_1(x_k, y_k) - I_2(x_k - d, y_k))^2 \quad (6)$$

$$f_{smooth}(d) = \frac{1}{2} |\nabla d|^2, \quad (7)$$

where ∇ is the gradient, and $\mathfrak{N}$ denotes the neighboring pixels. Using the calculus of variations [17], the minimum of Equation (5) can be obtained by solving its Euler-Lagrange equation

$$\frac{\partial F}{\partial d} = \lambda_1 \cdot \sum_{(x_k, y_k) \in \mathfrak{N}_1((x, y))} I_{2x}(x_k - d, y_k) (I_1(x_k, y_k) - I_2(x_k - d, y_k)) - \lambda_2 \cdot \Delta d = 0, \quad (8)$$

where I_{2x} is the derivative of the feature response in I_2 , and Δ is the Laplacian operator. The solution procedure is implemented in an iterative manner. The initial value of d is essential to the convergence speed. Kosov et al. [18] adopted a multi-grid strategy. The disparity values are estimated at a lower resolution, and are refined at a higher resolution with a feature-adaptive full approximation scheme. The estimation at the coarser grid provides a good initial guess for the iterative refinement at the finer grid, but the computation is still high for the full coarse grid estimation. Geiger et al. [10] perform Delaunay triangulation interpolation on a set of detected feature points to achieve fast implementation. The idea is that given the disparity values at a set of sparse supporting points, triangulation on these points can segment the image into small triangular regions, and the disparity of a point inside each region can be approximated through interpolation by disparity values of its three vertices. This method is very efficient for obtaining initial disparities. A disadvantage is that the erroneous detected vertices using the local method result in false interpolation. Thus we apply the global ICM on the detected supporting points. The iterative solution procedure is performed on the normal grid with the initial values obtained from the triangulation results.

3.1. Coarse initialization

To prepare candidate options for the ICM implementation, multiple disparities $d_1, d_2, \dots, d_n$, with the highest photo consistency are selected for each textured supporting point on a sparse grid.

$$\begin{aligned} (d_1, d_2, \dots, d_n) &= \arg \min(f_{data}(d)) \\ \text{s.t. } f_{data}(d_1) &\leq f_{data}(d_2) \leq \dots \leq f_{data}(d_n), \\ O(d_1) &= \text{false} \end{aligned} \quad (9)$$

Occlusion is detected with the thresholding method.

$$O(d) = \begin{cases} \text{false}, & \text{if } f_{data} \leq T_1 \\ \text{true}, & \text{otherwise} \end{cases} \quad (10)$$

Here T_1 is the thresholding parameter. The best option d is chosen from these n ($n = 2$ in our experiment) candidates according to

$$d = \arg \min_{(d_1, d_2, \dots, d_n)} F(d). \quad (11)$$

If a supporting point could not yield any valid match as described in (9), it will be eliminated from the set. After several iterations of ICM operation according to Equation (11), most spurious matches due to the presence of similar textures could be corrected, and a sufficient number of supporting points are obtained. The result can be observed from Figure 3 for the test data *Aloe* (1282x1110) [19]. The mismatch rate is reduced from 2.92% to 2.46% (two-pixel error threshold on non-occluded areas). And the extra computation time for two ICM iterations is negligible.

Figure 3: Disparity estimation for Aloe. The results using Geiger et al.'s method [11] with (left) and without (right) ICM operation for selecting the supporting points.

3.2. Finer evolution

After obtaining the initial disparity values by applying Delaunay triangulation interpolation on the supporting points, the result is further refined using Equation (8). The discretization form of the equation is

$$\lambda_1 \cdot \sum_{(x_k, y_k) \in \mathfrak{N}_1((x, y))} (I_2(x_k - d, y_k) - I_2(x_k - 1 - d, y_k)) (I_1(x_k, y_k) - I_2(x_k - d, y_k)) - \lambda_2 \cdot \sum_{d_j \in \mathfrak{N}_2(d)} (d_j - d) = 0 \quad (12)$$

By applying the Gauss-Seidel method [20], above linear equation can be solved iteratively. At the $(t+1)$ -th iteration, the disparity of pixel i is updated as

$$d_i^{t+1} = \frac{\lambda_1}{|\mathfrak{N}_2|} \sum_{(x_k, y_k) \in \mathfrak{N}_1((x, y))} (I_2(x_k - 1 - d_i^t, y_k) - I_2(x_k - d_i^t, y_k)) (I_1(x_k, y_k) - I_2(x_k - d_i^t, y_k)) + \frac{\lambda_2}{|\mathfrak{N}_2|} \left(\sum_{d_j \in \mathfrak{N}_2^-(d_i)} d_j^{t+1} + \sum_{d_j \in \mathfrak{N}_2^+(d_i)} d_j^t \right) \quad (13)$$

where $\mathfrak{N}_2^-(d_i)$, $\mathfrak{N}_2^+(d_i)$ are the neighboring pixels processed before and after pixel i . In practical

implementation, two situations are considered. In textured areas (detected with Sobel operator), a local search is applied according to the value computed from the second part $\lambda_2 \cdot (*)$. The disparity with the minimum data consistency penalty is used as the update. In non-textured areas, the value of the first part $\lambda_1 \cdot (*)$ from the data consistency penalty is small enough and is hence ignorable. λ_2 is set to 1 to impose the influence from the neighboring pixels. The evolution process stops when the maximum number of iterations is reached, or the change of the disparity values falls below a threshold. The iterative process is bound to converge since $|\mathfrak{N}_2| > 1$.

4. Experimental results

The experimental results contain two parts. First is the evaluation of the proposed disparity estimation method. Several stereoscopic image pairs from the Middlebury dataset [19] are tested on the matching accuracy and the processing time. The second part provides the object tracking results using the disparity-based window scaling MeanShift.

4.1. Disparity estimation

The supporting feature points used for triangulation interpolation are selected from a sparse grid on the tested images. Only intensity data are processed. Two types of grids with different cell size are tested in the experiment, the 8x8 cell size, and the 16x16 cell size. The calculated mismatch rate (M.R.) and the number of selected supporting points are listed in Table 1, with two-pixel error threshold on all non-boundary areas. Two iterations are performed on the interpolated initial estimation. The disparity estimation results can be observed from Figure 4. The occluded areas are interpolated using the results from the neighboring pixels.

	<i>Cones</i> (900x750)		<i>Teddy</i> (900x750)		<i>Aloe</i> (1282x1110)	
	M.R.	Points	M.R.	Points	M.R.	Points
8x8	6.2	1497	7.6	1247	9.9	2503
16x16	6.7	541	7.8	405	10.1	864

Table 1 Mismatch rate (%) and the number of supporting points.

The processing time on *Cones* for different phases of the estimation process is listed in Table 2. The mismatch rate reduction is displayed in Figure 5. Most of the disparity change occurs during the first two iterations. Note that the multi-grid algorithm by Kosov et al. takes up 300ms to 1300ms to process the same data at half the resolution (450x375), with similar computational resources, as reported in [19]. Compared to Geiger et al.'s method [10], the extra processing time concerns the iterative evolution at the finer grid. The average processing time is 52ms per

Figure 4: Disparity estimation. From left to right: the left image, ground truth, initial estimation, 1st iteration, 2nd iteration.

iteration, and the average mismatch rate reduction is 0.5%.

	Supp.	Tri.	Iter.1	Iter.2	Iter.3	Iter.4	Iter.5
8x8	489	158	57	52	52	52	52
16x16	383	124	62	53	52	52	53

Table 2 Processing time (ms) for different phases: computing supporting points, triangulation interpolation, 1st iteration, 2nd iteration, 3rd iteration, 4th iteration, and 5th iteration.

Figure 5: Mismatch rate reduction.

4.2. Object tracking

In the object tracking process, two PointGrey Firefly MV CMOS cameras are placed on the PTU device, and are connected to a desktop via a 1394 firewire USB2.0 hub. The 640x480 video is recorded at a frame rate of 15 fps. The average processing time for the disparity estimation is

178 ms per frame. An initial tracking window is obtained through user input. The tracking window is rescaled every 3 frames. The disparity estimation and the tracking results are provided in Figure 6. In the tracking process, the object walked along the corridor inside our department building. The environment contains both textured (mosaic tiles) and non-textured (wall, floor, pillar) materials. When applying traditional MeanShift tracking using fixed window size, the camera lost track of the object easily when the object approached areas with similar colors, such as the situation in Figure 6(b), when the color distribution inside the old tracking window could not fully represent the original target model.

Figure 6: Video object tracking.

5. Conclusions

In a multi-camera surveillance environment, online 3D content generation is required for more advanced applications. The presented video object tracking scheme incorporates a fast disparity estimation process into the MeanShift based tracking algorithm, in order to adjust the size of the tracking window, thus the camera can better follow the moving object. Due to the run-time requirement of the tracking application, traditional disparity estimation methods such as graph cut and belief propagation do not suffice. While most of the existing real-time disparity estimation methods rely on GPU implementation, the proposed scheme achieves 6fps on an ordinary PC (2.66GHz CPU, 4GB RAM) on the recorded tracking video (640x480). Its accuracy and runtime performance are evaluated on a set of standard test data, and the comparison with the semi-real-time schemes by Geiger et al. and Kosov et al. is analyzed. Currently the disparity estimation is performed independently on each image pair. The temporal correlation utilizing the camera control parameters and the tracking performance with more complex scenes will be studied in the future work.

References

- [1] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," TPAMI, vol.25, no.5, pp. 564- 577, May 2003.
- [2] J.-W. Hsieh and J.-X. Lee, "Video object segmentation using kernel-based models and spatiotemporal similarity," ICIP2006, pp. 1821-1824.
- [3] C. Yang, R. Duraiswami, and L. Davis, "Efficient mean-shift tracking via a new similarity measure," CVPR2005, pp. 176- 183.
- [4] P. Petrov, O. Boumbarov, and K. Muratovski, "Face detection and tracking with an active camera," ICIS2008, pp. 14-34-14-39.
- [5] T. Dang, C. Hoffmann, and C. Stiller, "Fusing optical flow and stereo disparity for object tracking," in Proc. 5th Intl. IEEE Conf. Intelligent Transportation Systems 2002. pp. 112- 117.
- [6] G. Mohammadi, F. Dufaux, T. H. Minh, and T. Ebrahimi, "Multi-view video segmentation and tracking for video surveillance," in SPIE Mobile Multimedia Image Processing, Security and Applications, April 2009.
- [7] C. Park and K.-H. Bae, "Quasi-Feature based Panoramic Video Creation for Multiview Object Tracking System," International Journal of Advanced Robotic Systems, 2009.
- [8] J. G. Lou, H. Cai, and J. Li, "A real-time interactive multi-view video system," in Proc. 13th ACM Intl. Conf. Multimedia, Singapore, Nov.2005, pp. 161-170.
- [9] C. Richardt, D. Orr, I. Davies, A. Criminisi, and N. Dodgson, "Real-time spatiotemporal stereo matching using the dual-cross-bilateral grid," ECCV2010, pp. 510-523.
- [10] A. Geiger, M. Roser and R. Urtasun, "Efficient Large-Scale Stereo Matching", ACCV2010, pp. 25-38.
- [11] C.L. Zitnick and T. Kanade, "A cooperative algorithm for stereo matching and occlusion detection," TPAMI, vol.22, no.7, pp.675-684, Jul 2000.
- [12] O. Woodford, P. Torr, I. Reid and A. Fitzgibbon, "Global stereo reconstruction under second-order smoothness priors," CVPR2008, pp.1-8.
- [13] J. Sun, H.-Y. Shum, and N.-N. Zheng. "Stereo matching using belief propagation," TPAMI, 25(7):787-800, 2003.
- [14] B.M. Smith, Li Zhang, and Hailin Jin, "Stereo matching with nonparametric smoothness priors in feature space," CVPR2009, pp.485-492.
- [15] J.R. Shewchuk, In: Applied Computational Geometry: Towards Geometric Engineering. vol. 1148. Springer, Berlin (1996) pp. 203-222.
- [16] J. Besag, "On the statistical analysis of dirty pictures," J. R. Stat. Soc. B, vol. 48, pp. 259-502, 1986.
- [17] L. Evans, Partial Differential Equations, Providence: American Mathematical Society, 1998.
- [18] S. Kosov, T. Thormahlen, and H. P. Seidel, "Accurate Real-Time Disparity Estimation with Variational Methods," ISVC '09 Proceedings of the 5th International Symposium on Advances in Visual Computing: Part I, pp. 796-807.
- [19] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," IJCV, 47(1-3):7-42, 2002.
- [20] D. M. Young, "Iterative Solution of Large Linear systems," New York: Academic, 1971.