EXPLOITING DISPARITY INFORMATION IN VISUAL OBJECT TRACKING

Olga Zoidi, Nikos Nikolaidis, Ioannis Pitas

Department of Informatics
Aristotle University of Thessaloniki
Box 451, Thessaloniki 54124, GREECE
{ozoidi, nikolaid, pitas}@aiia.csd.auth.gr

ABSTRACT

A novel method is proposed for visual object tracking in stereo videos. The algorithm employs Local Steering Kernel features and 2-dimensional color-disparity histograms for object texture description. The proposed framework requires no information about the intrinsic and extrinsic parameters of the stereo camera system. Therefore, it can be applied on 3D video content captured by commercial stereo cameras, as well as 3D movies and 3D TV programs. Experiments showed that the proposed method is effective in tracking objects under partial occlusion and changes in the object view angle.

Index Terms — stereo object tracking, local steering kernels, color-disparity histograms

1. INTRODUCTION

Visual object tracking finds applications in various fields, such as visual odometry, robotic vision, human-centered interfaces, surveillance systems and semantic annotation of content. Tracking performance is affected by numerous factors, such as varying illumination conditions, object partial, self, or total occlusions, presence of cluttered background, complicated object movements with varying speed and direction, object deformations, and presence of noise. The majority of the state-of-the-art methods consider the case of object tracking in monocular videos captured by a single camera [1] [2] [3]. However, recent advances in technology led to an increasing use of multiview systems in place of the monocular ones, such as surveillance systems [4]. The advantage of these systems is that they exploit the additional information obtained by the stereo geometry, namely the disparity (or depth) information [5] [6].

The majority of stereo video tracking systems use fixed position stereo cameras set in constrained environments with known camera calibration parameters [7] [8] [5]. However, these systems cannot be applied in the vast majority of available stereo data, coming from 3D cinema, 3D television or home-made 3D videos, which are captured in unconstrained environments, and usually carry no information about the intrinsic and/or extrinsic parameters of the stereo system. Therefore, the development of tracking algorithms which exploit stereo information without the knowledge of camera calibration information is required.

In this paper, we present a novel appearance-based algorithm which performs object tracking in the left or right video channel captured from an uncalibrated stereo camera, by exploiting the corresponding channel disparity information. The proposed framework combines a representation of the object texture based on Local Steering Kernel (LSK) descriptors [9], color information and disparity information. LSKs are descriptors of the object salient features and they are extracted by taking into account both the distance and the luminance value between neighboring image pixels. LSKs were employed effectively in object detection [9] and monocular object tracking [10] systems. The proposed method is an extension of [10] in stereo systems.

2. TRACKING FRAMEWORK OVERVIEW

The proposed method performs object tracking in a stereo video by employing 2-dimensional color-disparity histograms for reducing the candidate object Regions of Interest (ROIs) and the local steering kernel descriptors first introduced in [9] for texture description. First, the cosine similarity between the 2-dimensional color-disparity histogram of the object ROI in the previous video frame and equally sized regions in a search region $\mathbf{T} \in \Re^{M_x \times M_y}$ of the current video frame is computed and 80% of the patches with the smallest color-histogram cosine similarity are discarded as belonging to the background. Then, the algorithm computes the LSK similarity of the remaining candidate objects ROIs with the object instance in the first frame (initial object instance $\mathbf{I} \in \Re^{N_x \times N_y}$) and the object instance in a previous frame (stored object instance $\mathbf{Q} \in \Re^{N_x \times N_y}$), where a significant change in the object appearance was last observed. The size of the candidate objects is equal to the size of the stored object instance and the initial object instance.

The proposed algorithm starts by initialization of the position of the object at the first video frame. The object initialization can be achieved in two ways: automatically, by using an object detection algorithm, or manually, by inserting the object ROI in the initial frame. Then, the algorithm executes the following four iterative steps. In the first step, the first order Kalman filter is applied for predicting the new object position, the new search region is determined and the candidate object ROIs are initialized. In the second step, the color-histogram similarity between the object in

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement number 287674 (3DTVS). This publication reflects only the author's views. The European Union is not liable for any use that may be made of the information contained therein. The authors would like to thank the Fraunhofer Heinrich Hertz Institute for providing the stereo videos and the disparity maps used in the experimental section. The video in the experimental section belongs to the 3D4YOU project.

the previous frame and the candidate object ROIs is computed, retaining the 20% of the candidate object ROIs with the highest color-histogram similarity as candidate object ROIs. Then, the local steering kernel descriptors of the initial object instance, the stored object instance, and the remaining candidate object ROIs are extracted. Finally, the similarities of the candidate object ROIs to the initial object instance and the stored object instance are calculated and exploited in order to determine the new position of the object. The above procedure can be applied in one or both (left or right) channels of a stereoscopic video.

2.1. Position Prediction

First order Kalman filters are employed in order to estimate the new position of the object. Given the object state $\mathbf{x}_t = [p_x, p_y, dx, dy]^T$ at frame t, which consists of the object ROI center coordinates and velocity, the new state of the object ROI at frame $t+1$ is estimated from the motion state estimation model:

$$\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{n}_t \tag{1}$$

according to:

$$\hat{\mathbf{x}}_{t+1} = \mathbf{A}\hat{\mathbf{x}}_t, \tag{2}$$

$$\hat{\mathbf{P}}_{t+1} = \mathbf{A}\hat{\mathbf{P}}_t\mathbf{A}^T + \mathbf{Q}_s. \tag{3}$$

The measurement model is given by:

$$\mathbf{z}_{t+1} = \mathbf{H}\mathbf{x}_{t+1} + \mathbf{v}_{t+1}, \tag{4}$$

and it is adjusted according to:

$$\mathbf{K}_{t+1} = \hat{\mathbf{P}}_t\mathbf{H}^T(\mathbf{H}\hat{\mathbf{P}}_t\mathbf{H}^T + \mathbf{Q}_m)^{-1} \tag{5}$$

$$\hat{\mathbf{x}}_{t+1} = \hat{\mathbf{x}}_t + \mathbf{K}_{t+1}(\mathbf{z}_{t+1} - \mathbf{H}\hat{\mathbf{x}}_{t+1}) \tag{6}$$

$$\mathbf{P}_{t+1} = (\mathbf{I} - \mathbf{K}_{t+1}\mathbf{H})\mathbf{P}_{t+1}, \tag{7}$$

where $\mathbf{A}$ denotes the system transition matrix, $\mathbf{n}_t$ denotes the process noise with covariance matrix $\mathbf{Q}_s$, $\hat{\mathbf{P}}_t$ is the error covariance matrix, $\mathbf{z}_t = [p_x, p_y]^T$ is the system measurement, $\mathbf{H}$ is the measurement matrix, $\mathbf{v}_t$ is the measurement noise with covariance matrix $\mathbf{Q}_m$, and $\mathbf{K}_t$ is the Kalman gain. The matrices $\mathbf{A}$, $\mathbf{Q}_s$ and $\mathbf{Q}_m$ are kept constant throughout tracking.

Finally, the search region $\mathbf{T} \in \Re^{M_x \times M_y}$ in frame $t+1$ is defined around the predicted object ROI position that is included within the state $\hat{\mathbf{x}}_{t+1}$, whose size is equal to $M_x \times M_y = sN_x \times sN_y$, where s is a factor which determines the search region size. The value of s depends on the maximum velocity of the object and it should be large enough to keep track of the object in the selected search region. The object is then searched exhaustively in the determined search region, i.e., $(M_x - N_x + 1)(M_y - N_y + 1)$ candidate object ROIs of size $N_x \times N_y$ are extracted.

2.2. Color-disparity similarity

Disparity information is combined with color information in 2-dimensional histograms, for discriminating the object of interest from the surrounding background. We assume that, between two consecutive frames, the change of the object color and disparity histograms is rather small. Therefore, we can reduce the number of the candidate object ROIs at frame $t+1$ by discarding the ones with the lowest 2-D color-disparity histogram similarity to the detected object ROI at frame t.

The candidate object ROIs are split into their three RGB color channels and, for each channel, the 2-D color-disparity histograms $\mathbf{H}_R, \mathbf{H}_G, \mathbf{H}_B \in \Re^{16\times16}$ are computed. The 2-D color-disparity histograms are constructed by selecting 16 bins for the color and another 16 bins for the disparity information. The color bins widths are selected uniformly, while the disparity bins are selected as follows:

- The minimum and maximum disparity value of the first frame are extracted.
- The width of the first bin is set from 0 to the minimum disparity value.
- The width of the last (sixteenth) bin is set from the maximum disparity value in the first frame to the maximum disparity value in the entire video.
- The boundaries of the remaining bins (second to fifteenth) are set uniformly in the range from the minimal to the maximum disparity value.

This selection of the disparity bins provide discriminant disparity histograms in videos with small depth variations. In such videos, the disparity histogram is discriminant even when the object approaches the camera, as its disparity histogram varies significantly from the background disparity histogram. Moreover, since the algorithm employs a rough estimation of the disparity histogram in 16 bins, existence of high quality disparity map is not required. Therefore, low accuracy depth maps provided by fast disparity estimation algorithms can be employed, after the application of smoothing filters.

In order to measure the resemblance between the 2-D color-disparity histograms of the object ROI in the previous frame and the candidate object ROIs in the search region, $\mathbf{H}_R, \mathbf{H}_G, \mathbf{H}_B$ are column-stacked into 1-D color-disparity histograms $\mathbf{h}_R, \mathbf{h}_G, \mathbf{h}_B \in \Re^{256}$ and compared to the corresponding object histograms $\hat{\mathbf{h}}_R, \hat{\mathbf{h}}_G, \hat{\mathbf{h}}_B \in \Re^{256}$ of the previous frame via the cosine similarity:

$$c_k(\mathbf{h}_k, \hat{\mathbf{h}}_k) = cos(\theta) = \frac{<\mathbf{h}_k, \hat{\mathbf{h}}_k>}{\|\mathbf{h}_k\|\|\hat{\mathbf{h}}_k\|} \in [-1, 1], \quad k = R, G, B, \tag{8}$$

where $< \cdot >$ denotes the inner product, $\| \cdot \|$ is the Euclidean norm and θ denotes the angle between the two vectors. The total histogram similarity is computed as:

$$S = \sum_{k=R,G,B} \frac{c_k^2}{1 - c_k^2} \in [0, +\infty). \tag{9}$$

The total histogram similarity (9) is computed for every candidate object ROI and the 80% candidate objects with the lowest histogram similarity are discarded, meaning that only 20% of the selected candidate object ROIs will be further examined for being the new object ROI.

2.3. Local Steering Kernel feature extraction

The salient features of the initial object instance, the stored object instance, and the search region are extracted according to the following procedure. Initially, the image is transformed to the RGB color space. The local steering kernel descriptors (LSK) [9] are extracted in a locally defined $P \times P$ window. LSKs take into account both the illumination (pixel value) difference and the distance between neighboring pixels:

$$K(\mathbf{p}_l - \mathbf{p}) = \frac{\sqrt{\det(\mathbf{C}_l)}}{2\pi} \cdot \exp\left\{-\frac{(\mathbf{p}_l - \mathbf{p})^T\mathbf{C}_l(\mathbf{p}_l - \mathbf{p})}{2}\right\}, \tag{10}$$

$l = 1, \ldots, P^2$, where $\mathbf{p}$ denotes the coordinates of the image pixel, $\mathbf{p}_l$ denotes the coordinates of the neighboring pixels, and $\mathbf{C}_l$ is a covariance matrix, estimated from the matrix $\mathbf{J}_l$ of the gradient vectors of the image in a $P \times P$ window around $\mathbf{p}_l$:

$$\mathbf{J}_l = \begin{bmatrix} z_x(\mathbf{p}_1) & z_y(\mathbf{p}_1) \\ \vdots & \vdots \\ z_x(\mathbf{p}_{P^2}) & z_y(\mathbf{p}_{P^2}) \end{bmatrix}, \tag{11}$$

where $\mathbf{z}(\mathbf{p}) = [z_x(\mathbf{p}), z_y(\mathbf{p})]^T$ is the image gradient vector along x and y axes at the position $\mathbf{p}$. $\mathbf{J}_l$ is estimated by applying SVD, according to [11]:

$$\mathbf{J}_l = \mathbf{U}_l \cdot \begin{bmatrix} s_1 & 0 \\ 0 & s_2 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{v}_1^T \\ \mathbf{v}_2^T \end{bmatrix}_l. \tag{12}$$

Subsequently, $\mathbf{C}_l$ is evaluated as:

$$\mathbf{C}_l = \gamma \sum_{q=1}^{2} a_q^2 \mathbf{v}_q \mathbf{v}_q^T, \tag{13}$$

$$a_1 = \frac{s_1 + 1}{s_2 + 1}, \quad a_2 = \frac{s_2 + 1}{s_1 + 1}, \quad \gamma = \left(\frac{s_1 s_2 + 10^{-7}}{P^2} \right)^a, \tag{14}$$

where a is a parameter that restricts γ and in our experiments takes the value 0.008.

Given the image pixel $\mathbf{p}$, equation (10) is computed P^2 times, one for each neighboring pixel $\mathbf{p}_l$, $l = 1, \ldots, P^2$. Therefore, for each image pixel we calculate an LSK feature vector $\mathbf{K}(\mathbf{p}) \in \Re^{P^2 \times 1}$. By performing L_1-normalization:

$$\mathbf{N}(\mathbf{p}) = \frac{\mathbf{K}(\mathbf{p})}{\sum_{l=1}^{P^2} |K(\mathbf{p}_l - \mathbf{p})|} \in \Re^{P^2 \times 1}, \tag{15}$$

the LSK feature vectors become invariant to brightness and contrast changes. Finally, the LSK feature vectors of the $n = N_x N_y$ pixels of the stored object instance are ordered column-wise to form the LSK feature matrix $\mathbf{N}_Q \in \Re^{P^2 \times n}$. The LSK feature matrices $\mathbf{N}_I \in \Re^{P^2 \times n}$ and $\mathbf{N}_T \in \Re^{P^2 \times n_T}$, $n_T = M_x M_y$, for the initial object instance and the search region, respectively, are formed accordingly.

2.4. Position evaluation

The LSK feature matrices $\mathbf{N}_Q$, $\mathbf{N}_I$, are extracted for the stored object instance and the initial object instance respectively. Then, the similarity of the search region patches to the stored object instance and the initial object instance is measured according to cosine similarity. At first, PCA is performed to the normalized LSK feature vectors $\mathbf{N}_I$ of the object in the first frame, producing the matrix $\mathbf{F}_I \in \Re^{d \times n}$:

$$\mathbf{F}_I = \mathbf{A}_I \mathbf{N}_I. \tag{16}$$

The projection matrix of PCA $\mathbf{A}_I \in \Re^{d \times P^2}$ is then used in order to project the LSK feature matrices of the search region candidate object ROIs and the stored object ROI in a previous frame to the produced space, as follows:

$$\mathbf{F}_Q = \mathbf{A}_I \mathbf{N}_Q \in \Re^{d \times n} \qquad \mathbf{F}_T = \mathbf{A}_I \mathbf{N}_T \in \Re^{d \times n_T}. \tag{17}$$

The search region of size $M_x \times M_y$ is then divided into overlapping regions $\mathbf{T}_{ij}$, $i = 1, \ldots, m_x = M_x - N_x + 1$, $j = 1, \ldots, m_y = M_y - N_y + 1$, of size $N_x \times N_y$. For each region $\mathbf{T}_{ij}$, which was not discarded in subsection 2.2, the corresponding LSK feature matrices $\mathbf{F}_{T_{ij}} \in \Re^{d \times n}$ are extracted, containing only the columns of $\mathbf{F}_{T_I}$, which correspond to the pixels of the patch $\mathbf{T}_{ij}$. The LSK similarity of the search region patches to the query image and the initial query image is then computed by the cosine similarity:

$$s_{Qij} = s(\mathbf{F}_Q, \mathbf{F}_{T_{ij}}), \qquad s_{Iij} = s(\mathbf{F}_I, \mathbf{F}_{T_{ij}}), \tag{18}$$

where

$$s(\mathbf{F}_1, \mathbf{F}_2) = \sum_{l=1,j=1}^{n,d} \frac{F_1(l,j) F_2(l,j)}{\sqrt{\sum_{l=1,j=1}^{n,d} |F_1(l,j)|^2 \sum_{l=1,j=1}^{n,d} |F_2(l,j)|^2}}, \tag{19}$$

and $F_1(l,j)$, $F_2(l,j)$ are the (l,j) elements of matrices $\mathbf{F}_1$ and $\mathbf{F}_2$ respectively. The LSK cosine similarity values s_{Qij}, s_{Iij} are grouped in the resemblance maps $\mathbf{R}_Q, \mathbf{R}_I \in \Re^{m_x \times m_y}$. The new object instance in frame t is then defined as the region $\mathbf{T}_{ij}$ with the highest similarity R_{max}^t to the object model (i.e., the initial and stored object instances):

$$\mathbf{p}_t = \arg\max_{i,j} \left\{ \frac{1}{2} (\mathbf{R_Q} + \mathbf{R}_I) \right\}. \tag{20}$$

If the similarity R_{max}^t of the new object instance to the object model at frame t drops under a threshold with respect to R_{max}^{t-1} ($(R_{max}^{t-1} - R_{max}^t)/R_{max}^{t-1} < threshold$), then a change in the object appearance is detected and the new object instance acts as the new stored object instance $\mathbf{Q}$. This way, the algorithm incorporates the object appearance changes in the object model. In our experiments, we set $threshold = 0$.

3. EXPERIMENTAL RESULTS

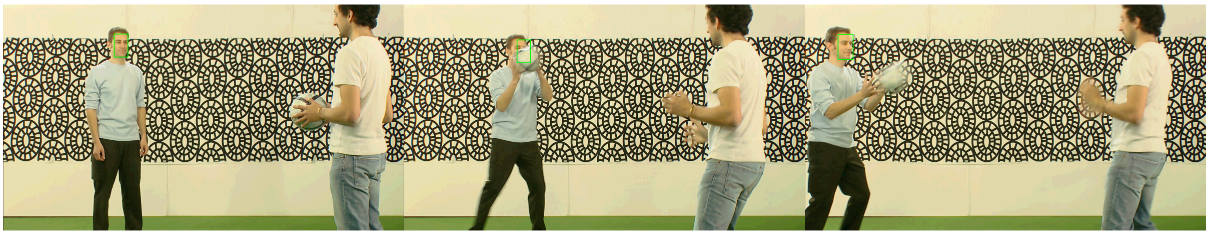
The performance of the proposed tracking scheme was tested in multiple videos captured by a stereo camera. The method employed for extracting the disparity maps is described in [12], [13]. The initialization of the tracking algorithm was accomplished with the object detector described in [9]. Snapshots of the tracking performance in the left channel of a video with resolution 1920 × 1080 pixels is shown in Figure 1. The tracker tracks successfully the man's face that performs fast movements under partial occlusion. The tracking performance of the proposed algorithm is compared to that of the tracker in [10] with respect to the Frame Detection Accuracy (FDA) measure. Given T the area of the tracked object in the video frame and G the area of the corresponding ground truth, the FDA at frame t is defined as:

$$FDA(t) = \frac{1}{N_t} \sum_{i=1}^{N_t} \frac{G_i(t) \cap T_i(t)}{G_i(t) \cup T_i(t)}, \tag{21}$$

where N_t denotes the number of objects in the frame. The results are shown in Figure 2. The Average Tracking Accuracy (ATA) in a video with N frames is defined as:

$$ATA = \frac{1}{N} \sum_{t=1}^{N} FDA(t). \tag{22}$$

The ATA of the proposed tracker and the tracker in [10] for three stereo videos are shown in Table 1. We notice that the disparity information improves the tracking accuracy in all three videos.



1^{st} frame 105^{th} frame 308^{th} frame

Figure 1. Tracking results on a fast moving object with partial occlusion.

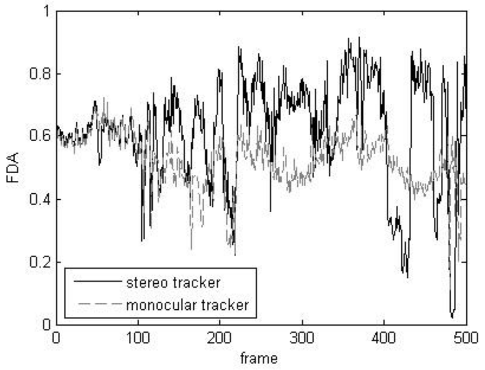


Figure 2. FDA accuracy of the proposed stereo tracker and the monocular tracker in [10]

.

Table 1. ATA of the stereo and monocular trackers.

ATA	stereo tracker	monocular tracker
video 1	0.6169	0.5236
video 2	0.6016	0.5498
video 3	0.6910	0.5313

4. CONCLUSION

In this paper, a novel method for visual object tracking in stereo videos was proposed, which employs Local Steering Kernel descriptors and 2-dimensional color-disparity histograms for the representation of object appearance. The proposed framework performs object tracking, without any information about the intrinsic and extrinsic parameters of the stereo system, rendering it suitable for application in any 3D content captured from commercial stereo cameras. Experimental results proved the effectiveness of the proposed stereo LSK tracker in tracking objects under partial occlusion, as well as changes in the object view angle. Future work is directed towards the fusion of information obtained from both luminance channels for concurrent object tracking in the left and right video channels.

5. REFERENCES

[1] C. Yang, R. Duraiswami, and L. Davis, “Efficient mean-shift tracking via a new similarity measure,” in *IEEE Conference on Computer Vision and Pattern Recognition.*, June 2005, vol. 1, pp. 176 – 183.

[2] A. Yilmaz, X. Li, and M. Shah, “Contour-based object tracking with occlusion handling in video acquired using mobile cameras,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 11, pp. 1531 –1536, Nov. 2004.

[3] Li-Qun Xu and Pere Puig, “A hybrid blob- and appearance-based framework for multi-object tracking through complex occlusions,” in *2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance.*, Oct. 2005, pp. 73 – 80.

[4] Ling Cai, Lei He, Yiren Xu, Yuming Zhao, and Xin Yang, “Multi-object detection and tracking by stereo vision,” *Pattern Recognition*, vol. 43, no. 12, pp. 4028 – 4041, 2010.

[5] Feng Tang, M. Harville, Hai Tao, and I.N. Robinson, “Fusion of local appearance with stereo depth for object tracking,” in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, June 2008, pp. 1 –8.

[6] Rafael Muñoz Salinas, Miguel García-Silvente, and Rafael Medina Carnicer, “Adaptive multi-modal stereo people tracking without background modelling,” *J. Vis. Comun. Image Represent.*, vol. 19, no. 2, pp. 75–91, Feb. 2008.

[7] A. Gaschler, D. Burschka, and G. Hager, “Epipolar-based stereo tracking without explicit 3d reconstruction,” in *20th International Conference on Pattern Recognition*, Aug. 2010, pp. 1755 –1758.

[8] Michael Harville, “Stereo person tracking with adaptive plan-view templates of height and occupancy statistics,” *Image and Vision Computing*, vol. 22, no. 2, pp. 127 – 142, 2004.

[9] H.J. Seo and P. Milanfar, “Training-free, generic object detection using locally adaptive regression kernels,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1688–1704, Sept. 2010.

[10] Olga Zoidi, Anastasios Tefas, and Ioannis Pitas, “Visual object tracking based on the object’s salient features with application in automatic nutrition assistance,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 25-30 March 2012.

[11] Hae Jong Seo and Peyman Milanfar, “Static and space-time visual saliency detection by self-resemblance.,” *Journal of Vision*, vol. 9, no. 12, pp. 1–27, 2009.

[12] N. Atzpadin, P. Kauff, and O. Schreer, “Stereo analysis by hybrid recursive matching for real-time immersive video conferencing,” *IEEE Transactions on Circuits and Systems for Video Technology,*, vol. 14, no. 3, pp. 321 – 334, March 2004.

[13] Zilly F. Kauff P. Riechert, C., “”real time depth estimation using line recursive matching,” in *European Conference on Visual Media Production*, Nov 2011.