Online Body Tracking by a PTZ Camera in IP Surveillance System

Parisa Darvish Zadeh Varcheie, Guillaume-Alexandre Bilodeau

Department of Computer Engineering and Software Engineering, École Polytechnique de Montréal
P.O. Box 6079, Station Centre-ville Montréal, (Québec), Canada, H3C 3A7
{parisa.darvish-zadeh-varcheie, guillaume-alexandre.bilodeau}@polymtl.ca

***Abstract*—In this paper, an online human body tracking method by an IP PTZ camera based on fuzzy-feature scoringe is proposed. Because the surveillance system uses a built-in web server, the camera control entails camera response time and network delays. Thus, the frame rate is irregular and in general low (3-7 fps). Our method has been designed specifically to perform in such conditions. At every frame, candidate targets are detected by extracting moving target using optical flow, a sampling, and appearance. The target is determined among samples using a fuzzy classifier. Results show that our system has a good target detection precision ($> 88\%$), and the target is almost always localized within 1/4th of the image diagonal from the image center.**

I. Introduction

One problem in video surveillance is how to identify and recognize events. Moving object detection and tracking is one of the key technologies for intelligent video monitoring systems. Furthermore, people detection and tracking are important capabilities for applications that require natural human-machine interactions. We propose to track objects using an IP PTZ camera (a network-based camera that pans, tilts and zooms). An IP PTZ camera responds to command via its integrated web server after some delays. Tracking with such camera implies: 1) irregular response time to control command, 2) low irregular frame rate because of network delays (the time between two frames is not necessarily constant), 3) changing field of view (FOV) and object scaling resulting from panning, tilting and zooming.

The proposed method detects in every frame candidate targets by extracting moving target using optical flow, a sampling method, and appearance. The target is detected among samples using a fuzzy classifier. Then, a movement command is sent to the camera using position and speed of the target. Our upper body tracking system has a good target detection precision ($> 88\%$), and low track fragmentation. Although one application is to track human upper body, our proposed methodology is general in the sense that it models the camera system as a servo control loop, and it accounts for response time and networks delays proper to such camera. It can be extended to other features or targets. In addition, results show that our method can cope with occlusion and large motion of the target.

The paper is structured as follows. Section II gives an overview of the state-of-the-art. Section III describes the servo system architecture. Section IV presents our proposed upper body tracking algorithm, and section V gives some results. Section VI concludes the paper.

II. Related works

Much works on face and upper body tracking have been reported. Comaniciu *et al.* [1] applied the mean-shift algorithm on an elliptical region which is modeled by histogram for face tracking. To adapt to fast scale changes, they also take advantage of the gradient perpendicular to the border of the hypothesized face region. Background subtraction is also used. This method is not designed to cope with large motion, as the target in the current frame needs to be near the previous position. The algorithm in Ido *et al.* [2] works by maximizing the PDF of the target's bitmap, which is formulated by the color and location of pixel at each frame. This information allows color and motion to be treated separately. Severe occlusions are not handled, and this algorithm is not very fast. Roha *et al.* [3] proposed a contour-based object tracking method using optical flow. It has been tested by selecting tracked object boundary edges in a video stream with a changing background and a moving camera. The face region needs to be large and it is computationally expensive. In the work of Elder *et al.* [4] two cameras are used, one is a stationary, preattentive, low-resolution wide FOV camera, and the other is a mobile, attentive, high-resolution narrow FOV camera. They used skin detection, motion detection, and foreground extraction for face tracking. The advantage of this work is a wide FOV, but it relies on a communication feedback between two cameras.

In several papers, contour-based people tracking is proposed [5], [6]. It is assumed that the background motion between two consecutive images could be approximated by an affine transformation. Their methods are limited to small motion and a time consuming for IP camera. In addition they can track only continuously moving edges and cannot track temporarily stopping objects [7]. In [8], a camera based position tracking system (PCTS) for person tracking is used. The centroid of human is calculated from the centroid of the detected foreground. It is only based on motion detection. There is no other feature comparison. Their method fails in different tracking conditions with more than one moving object in a scene. Indeed there is no object segmentation.

Funahasahi *et al.* [9] developed a system for human head and facial parts tracking by a hierarchical tracking method using a stationary camera and a PTZ camera. At first, irises

978-1-4244-5561-4/09/$26.00 ©2009

are recognized from motion images. Then, detected irises are used as feature for face detection. The face needs to be large enough to detect the irises. In the method of Bernardin *et al.* [10], the upper body histogram information, KLT feature tracker, and active camera calibration are combined to track the person. It is used for 3D localization. In the algorithm of Li *et al.* [11] each observer should be learned from different ranges of samples, with various subsets of features. Their method needs a learning step that is based on model complexity and increases computation time. The method has limitations in distinguishing between different targets, and has model overupdating problems. Kang *et al.* [12] used a geometric transform-based mosaicing method for person tracking by a PTZ camera. For each consecutive frame, it finds the good features for the correspondence and then tries to shift the moved image and update the changed background. They are using a high cost background modeling using a calibration scheme, which is not suitable for tracking by internet-based PTZ cameras.

III. System architecture

The servo controlling and tracking system is modeled by a closed-loop control which has a negative feedback as shown in Fig. 1. Servo system consists of three main blocks which are image capture, visual processing, and camera control. Tracking is affected by two delays which are the delay τ_1 from image capture and the delay τ_2 in the feedback loop from executing camera motion commands. The delay for visual processing is much smaller than the two other delays, thus it is neglected. The input of the system is the current pan and tilt angles of the camera. The output will be the determined pan and tilt angles by the fuzzy classifier. The delays imply that the current position of the target cannot be used for centering the camera. To compensate for motion of the target during the delays, a position predictor block is added. The algorithms which are used for the servo system control are explained in Section IV.

IV. Methodology

Our method is based on comparing elliptic samples with the target model and evaluating their likelihood to estimate the location of the target. We have made the following assumptions: 1) persons walk at a normal pace or fast, but do not run, and 2) the FOV is wide (approximately 48°) and 3) scenes are not crowded (max 2-3 persons).

A. Visual processing

1) Target modeling: A target is represented by an elliptical image region. It is modeled by two features: 1) quantized HSV color histogram with 162 bins (i.e. $18\times3\times3$), and 2) the mean of R, G and B color components of RGB color space of all the pixels inside of the elliptical region. Initialization is done manually by selecting the top part of the body (head and torso) of the person. We fit an ellipse inside the bounding box of the selected region (Fig. 2 (a) and (e)) because it approximates adequately the shape of the head and torso. Then the initial target M is modeled by the two discussed features.

2) Target candidate detection: For tracking, we sample with ellipses the image around regions of interest and model them. There are two regions of interest: 1) areas with motion, 2) the center of the image.

- *Motion-based samples*: The first type of samples is detected by estimating the motion of the target from two consecutive images I_1 and I_2, using pyramidal Lucas Kanade optical flow [13]. In optical flow [14], strong corners in the image which have big eigenvalues are detected for comparison. To solve pixel correspondence problem for a given pixel in I_1, we look for nearby pixels of the same color in I_2. We use 4 pyramid levels with 10 iterations. The threshold is 0.3.
  As found experimentally, the detected motion vector results are noisy. In addition, the camera motion vectors have effect on object motion vectors. Thus, to remove this effect, camera motion vectors are extracted. To calculate the camera motion vectors, a radial histogram of motion vectors is calculated. In a radial histogram, each bin is based on the quantized length (r) and angle (θ) of the motion vector. Our radial histogram, $h(r,\theta)$ has 36180 bins. r has 200 values and is varied based on the image size between 1 and image diameter. θ has 180 values and is varied between 0° and 360°. The r and θ of the bin that has the maximum number of vectors is assigned to be the camera motion vector length and angle. The detected motion vectors which have this length and angle are removed and then the camera motion vector is subtracted from the rest of the motion vectors.
  Motion-based samples are extracted around object motion vectors. Sample size is large or small. The largest samples are used for zooming or for object approaching the camera. Their area is 1/3 of the image area. The small samples are used for targets far from the camera and close to the center in different positions. The sizes of these elliptic samples are obtained experimentally according to the minimum face size, which is in our algorithm 5x5 pixels from frontal and lateral views. We consider F small samples located uniformly and one large sample for each motion vector group (typically $F = 16$).
- *Fixed samples*: According to our goal, the object should be always near the image center. To have robust tracking even when there is no motion from the target, we consider G additional fixed samples in the image (typically $G = 10$).

The ellipses are then modeled as explained previously.

3) Sample likelihood using a fuzzy classifier: To localize the target, features of each sample S_i are compared with the initial model M, and a score ($ScoreS_i$), or sample likelihood is given to each S_i using a fuzzy rule. The score is obtained by multiplying four fuzzy membership functions which will be explained in the following.

$$ScoreS_i = \mu_{EC}.\mu_{EP}.\mu_{EH}.\mu_{H}. \quad (1)$$

The target is the sample with the highest score. We are using four membership functions, each with fuzzy outputs between

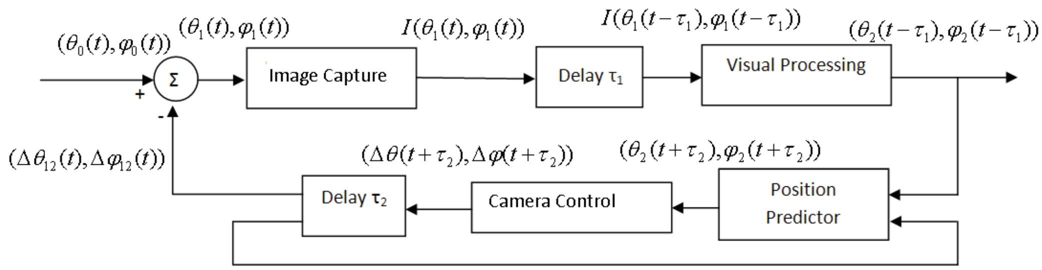


Fig. 1. The system architecture and servo control model. (θ_0,ϕ_0):initial pan-tilt angles,($\Delta\theta_{12}$,$\Delta\phi_{12}$) means (θ_1-θ_2,ϕ_1-ϕ_2)

0 and 1:

1) The membership function μ_{EC} is used for Euclidean distance between mean RGB of S_i (R_{si}, G_{si}, B_{si}) and mean RGB of M (R_m,G_m, B_m). It is defined as

$$\mu_{EC} = 1 - \frac{\sqrt{(R_{si}-R_m)^2+(G_{si}-G_m)^2+(S_{si}-S_m)^2}}{255\sqrt{3}}. \tag{2}$$

2) The membership function μ_{EP} is used for Euclidean distance of B_i centroid from the image center. Indeed, normally the person should be near the image center. It is defined as

$$\mu_{EP} = exp(-\frac{(\sqrt{(x_{si}-x_{im})^2+(y_{si}-y_{im})^2})^2}{2\sigma^2}). \tag{3}$$

where (x_{si}, y_{si}) and (x_{im}, y_{im}) are respectively the coordinate vector of the centroid of S_i and of the image center. σ^2 is equal to a quarter of the image area around the image center.

3) The membership function (μ_{EH}) is applied for Euclidean distance between quantized HSV color histogram [15] of S_i and the histogram of M. It is computed as

$$\mu_{EH} = 1 - \sqrt{\frac{\sum_n (H_{si}[n]-H_m[n])^2}{2}}. \tag{4}$$

where H_{si} and $H_m$denote the normalized histograms of S_i and M respectively and n is the histogram bin number.

4) Finally, the membership function of μ_H is used for calculating the similarity [16] of quantized HSV color histogram vector of B_i with histogram of M. It is the normalized correlation coefficient of two histogram vectors and it is defined as

$$\mu_H = \frac{1}{2} + \frac{\sum_n((H_{si}[n]-\bar{H}_{si})(H_m[n]-\bar{H}_m))}{2\times\sqrt{\sum_n (H_{si}[n]-\bar{H}_{si})^2}\sqrt{\sum_n (H_m[n]-\bar{H}_m)^2}}. \tag{5}$$

where $\bar{H}_{si}$ and $\bar{H}_m$ denote the average of normalized histograms of S_i and M.

B. Target position prediction and camera control

As discussed in Section III, a position predictor based on the two last motion vectors has been designed to compensate for motion of the target during the delays. This motion predictor considers the angle between two consecutive motion vectors. If the angle difference is smaller than 25°, it is assumed that the target is moving in the same direction. Thus, the system will put the camera center on the predicted position which is :

$$x_P = x_E + \bar{\tau}_2 \times \frac{\Delta x_1 + \Delta x_2}{\tau_1^1 + \tau_1^2}. \tag{6}$$

where Δx_1 and Δx_2 are the two target displacement vectors (i.e. target motion vector). τ_1^1, τ_1^2 are delay τ_1 between the two last captured images. x_P is the predicted target coordinate and x_E is the extracted target coordinate from the fuzzy classifier. $\bar{\tau}_2$ is the average delay time τ_2 obtained from previous camera movements.

To follow the target, the PTZ motors are commanded based on x_P. Camera is controlled by computing the pan and tilt angles from a workstation and sending HTTP POST request using the CGI scripts of the camera [17].

V. Experiments and analysis

A. Data acquisition and ground-truth

We used two Sony IP PTZ cameras (SNC-RZ50N and SNC-RZ25N) for our tests. When the cameras are not moving, a frame rate of 30 fps is achievable. For validation, we tested the complete system in online upper body tracking experiments. No dataset is available for testing the complete tracking system, because of its dynamic nature. The tracking algorithm has been tested over events such as entering or leaving the FOV of the camera and occlusion with other people in the scene. We recorded all the experiments to extract their ground-truth manually for performance evaluation. The general scenario of the experiments is the following. An actor from the frontal view is selected for initial modeling. The actor starts to walk around in the room. Two or three actors can walk at the same time in different directions, crossing and occluding with the target. The target actor makes some pauses while walking to verify the performance for stationary target.

The target actor also moves parallel, toward or away from the camera. Fig. 2 shows the initial model selection and some frames obtained during tracking.

To evaluate our method, four metrics are used as explained in Table I. We have done twenty experiments with the two IP cameras. The experiments are reported in Table II. Experiments are classified into four classes based on the camera model, initial model position from camera, and image resolution.

B. Results

We first present the results. Discussion follows in the next section. Fig.3 shows the d_{gp} and d_{gc} values for E_8 and E_{14}. These distances allow us to verify if the tracker has lost the target, and if the target is near the ground-truth. If d_{gc} is more than 1, target is lost (outside of FOV). For distances smaller than 0.6, the object is in the FOV, but for the range of $(0.6 < d_{gc} < 1)$, it depends if the centroid coordinates of the target are inside the image.

For E_8, the target is always in the FOV. For E_{14}, the target is lost several times between frame 59-80. For both experiments, d_{gp} is small except when the target is lost. Table II shows the results of the five metrics with the system frame rate for all experiments. The algorithm is implemented on an Intel Xeon(R) 5150 in C++ using OpenCV. For d_{gc} and d_{gp}, we show the mean and variance of all experiments. For class 1 and 3, because of the lower system frame rate, the method has lost the target several times, but eventually recovers. For experiments of class 1, the system frame rate is a little higher. Thus, there is fewer target lost (e.g. TF is smaller), less distance error ($\mu_{d_{gc}}$ and $\mu_{d_{gp}}$) and the results of precision are better (P is larger). Because of d_{EP} and camera control, the error on $\mu_{d_{gc}}$ has effect on $\mu_{d_{gp}}$ and vice versa. For class 2 and 4, the system frame rate is increased because of smaller image size. TF for class 2 and 4 are smaller than class 1 and 3. A faster system frame rate improves the results of TF, $\mu_{d_{gc}}$, $\mu_{d_{gp}}$ and P. Δx_{min} and Δx_{max} in Table II are the minimum and maximum length of target motion vector in number of pixels.

In comparison of two Sony IP PTZ cameras, results from SNC-RZ50N (class 1 and class 2) are better than the results of SNC-RZ25N (class 3 and class 4).It is because of camera model, the SNC-RZ50N moves faster than SNC-RZ25N [19], [20].

C. Discussion

Results show that our algorithm can handle and overcome large motion (i.e. high values of Δx_{max}) because of using a repetitive target detection scheme and a motion prediction technique that do not rely on spatial proximity. It will lose a target only if the target changes its motion direction suddenly and walks very fast in the opposite predicted position (e.g. experiments with $TF \neq 0$). By using a repetitive detection scheme and combining it with a motion predictor, we can handle random motion between frames, as long as the target position is well predicted, and its appearance does not change significantly. The motion predictor is used to compensate the two delays τ_1 and τ_2 discussed in Section III, which may cause the target to exit the FOV. Small values of $\sigma^2_{d_{gp}}$ result from utilizing pyramidal Lucas Kanade optical flow for the motion extraction and combining it with a sampling method. This process of motion extraction with combination of fuzzy classifier which uses color features to generate sample likelihood allow us to detect and track any moving objects.

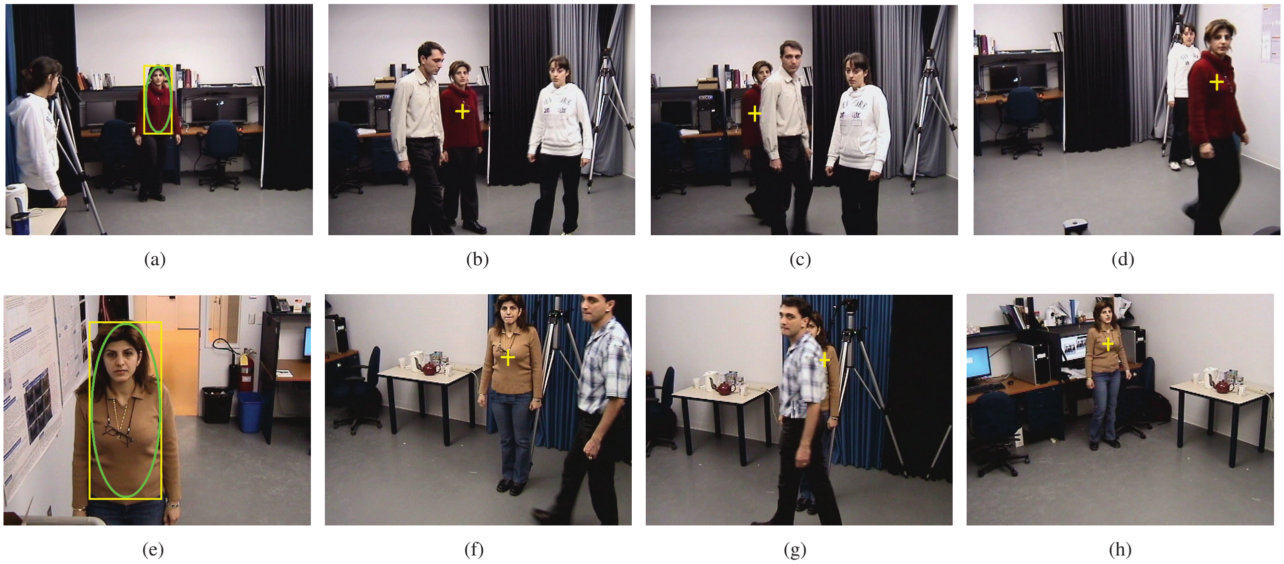
Generally, according to the mean of distances, the location of the target is near to the ground-truth. The target is usually localized within 1/4th of the image diagonal from the image center. With faster system frame rate the results of tracking have been improved significantly. In that case, we will have more frames to process. When localization fails, it is because of similarity or closeness of the color histogram of the target with other samples. The image resolution has effect on the system frame rate and thus on tracking error.

In all experiments, there are scale changes to verify tracking against scaling. Our algorithm can overcome scaling variations even in images with minimum 5×5 face size(e.g. Fig.2(e) and (d)). It is because of using normalized color histogram and average color features. These two features are independent of the size of the target.

Our method can also recover the tracking if it loses the object (e.g. experiments with $TF \neq 0$), because of the repetitive detection scheme. Of course, it is conditional to the object being in the FOV of the camera. Occlusions are handled in the same way. However, when the object is occluded, another similar object will be tracked (the most likely sample) until the occlusion ends. This may cause the real target to become out of the FOV of the camera. Fig. 2 shows an example of short-term occlusion handling. The proposed method can handle it in this case. In the reported experiments, occlusion did not cause difficulties. The duration of the experiments are short because the goal the algorithm will be zooming on target face and capturing it for identification purpose.

VI. Conclusion

In this paper, an object tracking method is proposed and applied to human upper body tracking by IP PTZ camera in online application. Human body tracking determines the location and size of each human body for each input image of a video sequence. It can be used to get images of the face of a human target in different poses. It detects in every frame, candidate targets by extracting moving objects using optical flow, and sampling around the image center. The target is detected among candidate target samples using a fuzzy classifier. Then, a movement command is sent to the camera using the target position and speed. Results show that our algorithm can handle and overcome large motion between two consecutive frames with low track fragmentation. Future work will be adding camera zooming and enhancing robustness of the motion prediction to prevent the target from being out of the camera FOV.



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 2. Examples of tracking frames for Exp_9 (a) to (d) and Exp_{12} (e) to (h). Exp_9 (a) initial model selection, (b) short-term occlusion, (c) after occlusion, (d) scale variation; Exp_{12} (e) initial model selection, (f) short-term occlusion, (g) after occlusion, (h) scale variation

TABLE I
EVALUATION METRICS.

Metric	Description
$P = \frac{TP}{TP+FP}$	to calculate the target localization accuracy
$d_{gc} = \frac{\sqrt{(x_c-x_g)^2+(y_c-y_g)^2}}{a}$	to evaluate the dynamic performance of the tracking system; It is the spatial latency of the tracking system, as ideally, the target should be at the image center.
$d_{gp} = \frac{\sqrt{(x_p-x_g)^2+(y_p-y_g)^2}}{2a}$	to evaluate the error of the tracking algorithm. Ideally, d_{gp} should be zero.
$TF = \frac{T_{OUT}}{NF}$	to indicate the lack of continuity of the tracking system for a single target track [18]

TP: number of frames with target located correctly, FP:number of frames with target not located correctly, a: radius of circle which circumscribes the image, (x_g,y_g): ground-truth target coordinates, (x_c,y_c): image center, (x_p,y_p): tracked object coordinates, T_{OUT}: number of frames with target out of FOV, NF: total number of frames.

REFERENCES

[1] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE T-PAMI*, vol. 25, no. 5, pp. 564–577, 2003.

[2] I. Leichter, M. Lindenbaum, and E. Rivlin, "Bittracker- a bitmap tracker for visual tracking under very general conditions," *IEEE T-PAMI*, vol. 30, no. 9, pp. 1572–1588, 2008.

[3] M. Roha, T. Kima, J. Park, and S. Lee, "Accurate object contour tracking based on boundary edge selection," *Pattern Recognition*, vol. 40, no. 3, pp. 931–943, 2007.

[4] J. H. Elder, S. Prince, Y. Hou, M. Sizintsev, and E. Olevsky, "Pre-attentive and attentive detection of humans in wide-field scenes," *International Journal of Computer Vision*, vol. 72, no. 1, pp. 47–66, 2007.

[5] D. Murray and A. Basu, "Motion tracking with an active camera," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, pp. 449–459, 1994.

[6] 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, 2004.

[7] S. Araki, N. Matsuoka, N. Yokoya, and H. Takemura, "Realtime tracking of multiple moving object contours in a moving camera image sequences," *IEICE Transaction on Information and Sytem*, no. 7, pp. 1581–1591, 2000.

[8] C. Yang, R. Chen, C. Lee, and S. Lin, "Ptz camera based position tracking in ip-surveillance system," *Int. Conf. on Sensing Technology*, pp. 142–146, 2008.

[9] T. Funahashi, M. Tominaga, T. Fujiwara, and H. Koshimizu, "Hierarchical face tracking by using ptz camera," *IEEE Int. Conf. on Automatic Face and Gesture Recognition (FGR)*, pp. 427–432, 2004.

[10] K. Bernardin, F. Camp, and R. Stiefelhagen, "Automatic person detection and tracking using fuzzy controlled active cameras," *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8, 2007.

[11] Y. Li, H. Ai, T. Yamashita, S. Lao, and M. Kawade, "Tracking in low frame rate video: a cascade particle filter with discriminative observers of different life spans," *IEEE T-PAMI*, vol. 30, no. 10, pp. 1728–1740, 2008.

[12] S. Kang, J. Paik, A. Koschan, B. Abidi, and M. Abidi, "Real-time video tracking using ptz cameras," *6 th Int. Conf. on Quality Control by Artificial Vision*, pp. 103–111, 2003.

[13] J. Y. Bouguet, "Pyramidal implementation of the lucas kanade feature tracker: Description of the algorithm," Jean-YvesBouguet, 2002, kLT implementation in OpenCV.

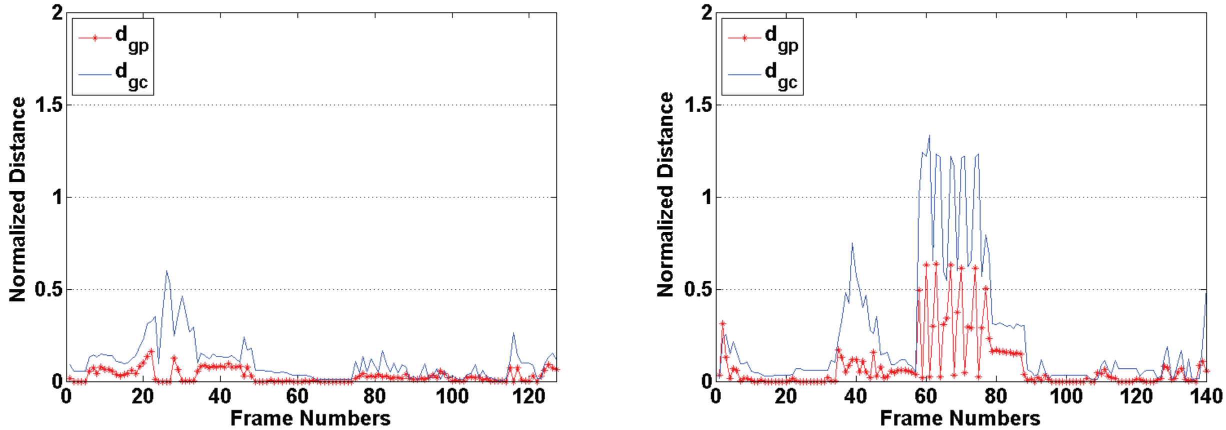
[14] J. Shi and C. Tomasi, "Good features to track," *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 593–600, 1994.

[15] S. H. Cha and S. N. Srihari, "On measuring the distance between histograms," *Pat. Recog.*, vol. 35, no. 6, pp. 1355–1370, 2002.

[16] B. Boufama and M. Ali, "Tracking multiple people in the context of video surveillance," *Int. Conf. on Image Analysis and Recognition (ICIAR)*, pp. 581–592, 2007.

[17] Sony corporation, "Snc-rz25n/p cgi command manual," 2005, version 1.0.

[18] F. Yin, D. Makris, and S. Velastin, "Performance evaluation of object



(a) E_8

(b) E_{14}

Fig. 3. d_{gc} and d_{gp} results for (a) E_8 and (b) E_{14}

TABLE II
EXPERIMENTAL RESULTS.

	P (%)	TF (%)	$\mu_{d_{gc}}$	$\sigma^2_{d_{gc}}$	$\mu_{d_{gp}}$	$\sigma^2_{d_{gp}}$	$FR(fps)$	NF	Δx_{min}	Δx_{max}	IS	IMP	CM
E_1	96	2.5	0.2849	0.0474	0.0785	0.0069	3.72	314	12	247	L	N	50N
E_2	97	1.8	0.3488	0.0359	0.0714	0.0064	3.71	308	5	189	L	N	50N
E_3	93	1.6	0.2182	0.0499	0.0411	0.0050	3.57	299	6	298	L	F	50N
E_4	92	2.4	0.3132	0.0559	0.0458	0.0028	3.76	283	1	543	L	F	50N
E_5	89	1.9	0.2903	0.0388	0.1154	0.0168	3.69	266	13	394	L	M	50N
class 1	**97**	**2.04**	**0.2911**	**0.0456**	**0.0704**	**0.0076**	**3.68**	**1470**	**1**	**543**	L	A	**50N**
E_6	95	0.7	0.1978	0.0285	0.0313	0.0012	7.31	703	8	231	S	N	50N
E_7	100	0	0.2581	0.0204	0.0485	0.0019	7.30	672	0	144	S	N	50N
E_8	96	0.1	0.1898	0.0227	0.0352	0.0015	7.22	686	2	112	S	F	50N
E_9	93	0.5	0.2667	0.0228	0.0496	0.0014	7.24	674	5	180	S	F	50N
E_{10}	90	0.4	0.2563	0.0261	0.0611	0.0080	7.14	641	2	235	S	M	50N
class 2	**94.8**	**0.34**	**0.2337**	**0.0241**	**0.0451**	**0.0028**	**7.24**	**3376**	**0**	**235**	S	A	**50N**
E_{11}	87	2.6	0.2565	0.0414	0.0727	0.0099	3.34	263	14	359	L	N	25N
E_{12}	89	2.7	0.3213	0.0472	0.0885	0.0127	3.30	260	10	207	L	N	25N
E_{13}	88	2.3	0.3253	0.0852	0.1261	0.0256	3.28	258	4	259	L	F	25N
E_{14}	90	1.9	0.3056	0.0384	0.0428	0.0069	3.33	252	6	275	L	F	25N
E_{15}	86	2.8	0.4692	0.1024	0.1254	0.0360	3.30	256	21	313	L	M	25N
class 3	**87.9**	**2.46**	**0.3356**	**0.0629**	**0.0911**	**0.0182**	**3.30**	**1289**	**4**	**359**	L	A	**25N**
E_{16}	92	1.3	0.2595	0.0348	0.0548	0.0019	7	623	7	272	S	N	25N
E_{17}	95	0.76	0.2536	0.0286	0.0348	0.0020	6.87	618	2	106	S	N	25N
E_{18}	100	0	0.2064	0.0306	0.1244	0.0409	6.76	619	0	109	S	F	25N
E_{19}	92	0.8	0.2246	0.0195	0.0954	0.0203	6.76	594	4	122	S	F	25N
E_{20}	93	0.6	0.3033	0.0259	0.0574	0.0026	6.92	611	6	111	S	M	25N
class 4	**94.4**	**0.69**	**0.2495**	**0.0279**	**0.0734**	**0.0135**	**6.86**	**3065**	**0**	**272**	S	A	**25N**

$\mu_{d_{gc}}$: mean of d_{gc}, $\mu_{d_{gp}}$: mean of d_{gp}, $\sigma^2_{d_{gc}}$: variance of d_{gc}, $\sigma^2_{d_{gp}}$: variance of d_{gp}, FR: System frame rate and Δx_{min} and Δx_{max}: minimum and maximum motion vector length, IS: Image Size, IMP: Initial model position from camera, N: Near, F: Far, M: Middle, L: 640 x 480, S: 320 x 240, A: All possible initial model positions from camera, CM: Camera Model.

tracking algorithms," *IEEE Int. Workshop on Performance Evaluation of Tracking and Surveillance(PETS)*, 2007.

[19] Sony corporation, "Network camera- snc-rz50n, snc-rz50p," 2005, http://www.sony.ca/ip/brochures/ip_cameras/SNCRZ50N%20Brochure.pdf, [Online; accessed 21-September-2009].

[20] ——, "Network camera- snc-rz25n," 2005, http://www.sony.ca/ip/brochures/ip_cameras/SNCRZ25N.pdf, [Online; accessed 21-September-2009].