Fuzzy Feature-Based Upper Body Tracking with IP PTZ Camera Control

Parisa Darvish Zadeh Varcheie and 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, we propose a fuzzy-feature based method for online upper body tracking using an IP PTZ camera. Because the camera uses a built-in web server, camera control entails camera response time and network delays, and thus, the frame rate is irregular and in general low (2-7 fps). It detects at every frame, candidate targets by extracting motion, a sampling method, and appearance. The target is detected among samples with a fuzzy classifier. Results show that our system has a good target detection precision ($> 85\%$), low track fragmentation, and the target is almost always localized within 1/6th of the image diagonal from the image center.

1 Introduction

People detection and tracking are important capabilities for applications that desire to achieve a natural human-machine interaction such as people identification. Here, we are interested in human upper body tracking by an IP PTZ camera (a network-based camera that pans, tilts and zooms). Upper body tracking determines the location of the upper body in each image. An IP PTZ camera communicates and responds to command via its integrated web server after some delays. Tracking with such camera involves some difficulties which are: 1) irregular response time to control command, 2) low usable frame rate (while the camera executes the motion command, the frames received are useless), 3) irregular frame rate because of network delays (the time between two frames is not necessarily constant), 4) changing field of view (FOV) resulting from panning, tilting and zooming and 5) various scales of objects.

Much works on face and upper body tracking have been reported. Comaniciu *et al.* [1] applied the mean-shift algorithm to an elliptical region which is modeled by histogram for face tracking. They also take advantage of the gradient perpendicular to the border of the hypothesized face region and background subtraction. This method is not designed to cope with large motion. 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. 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

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 809–816, 2009.

is computationally expensive. In the work of Elder *et al.* [4] a stationary, preattentive, low-resolution wide FOV camera, and a mobile, attentive, high-resolution narrow FOV camera are used. 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. Funahasahi *et al.* [5] developed a hierarchical tracking method using a stationary camera and a PTZ camera. The face needs to be large enough to detect the irises. Then, detected irises are used as feature for face detection. In the method of Bernardin *et al.* [6] the upper body histogram information, KLT feature tracker, and active camera calibration are combined to track the person for 3D localization application. In the algorithm of Li *et al.* [7] each observer should be learned from different ranges of samples, with various subsets of features. Learning step 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.* [8] uses 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. They are using a high cost background modeling using a calibration scheme, which is not suitable for tracking by internet-based PTZ cameras.

In our work, we want to cope with the problem of large motion detection, low usable frame rate, and tracking with various scale changes. In addition, the tracking algorithm should handle the camera response time. The proposed method consists of target modeling to represent the tracked object, target candidates detection (sampling), target localization using a fuzzy classifier, target position prediction and camera control to center the PTZ camera on the target. Results show that our system has a good target detection precision ($> 85\%$), low track fragmentation, and the target is almost always localized within 1/6th of the image diagonal from the image center.

2 System Architecture and Methodology

The servo controlling and tracking system is modeled by a closed-loop control which has a negative feedback as shown in Fig. 1. It consists of three main blocks : image capture, upper body detection and camera control. Tracking is affected by two delays which are the delay from image capture and the delay in the feedback loop from executing camera motion commands. The delay from upper body detection is considered negligible compared to the two other delays. The input of the system is the current pan

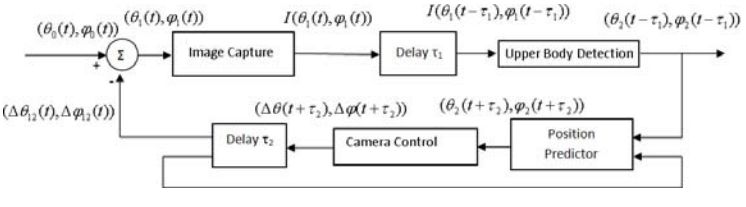

Fig. 1. The system architecture and servo control model. (θ_0,ϕ_0):initial pan-tilt angles, ($\Delta\theta_{12}$,$\Delta\phi_{12}$) means (θ_1-θ_2,ϕ_1-ϕ_2).

and tilt angles of the camera and the output will be the determined pan and tilt angles by the fuzzy classifier. The delays imply the current position of the target cannot be used for centering the camera. The algorithm in upper body detection has three steps: 1) target modeling to represent the tracked object, 2) target candidates detection (sampling), and 3) target localization by scoring sample features. Target position prediction and camera control utilize the upper body detection results. To compensate for motion of the target during the delays, a position predictor block is added. Camera control is used to put the image center of PTZ camera on the target. We have made the following assumptions: 1) skin detection will be done over the yellow, white, light brown and pink color skin types from number 1 to 26 on Von Luschans skin chromatic scale (almost 73% of all skin types in the world [9]), 2) persons walk at a normal pace or fast, but do not run, 3) the target person can walk in any direction, but the face should be always visible partially, 4) a wide FOV (approximately $48\,^{\circ}$) is assumed and scenes are not crowded (max 2-3 persons).

2.1 Upper Body Detection

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. $18 \times 3 \times 3$) 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)). Ellipse fits better the shape of the head and torso. Then the initial target M is modeled by the two discussed features.

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

1. *Motion-based samples*: The first type of samples is detected using motion of the target from the difference of two consecutive frames while the camera is not moving. The difference results are noisy and some morphological operations such as erosion, dilation, image closing (by a circular structuring element of 3 pixels radius) and filtering (by a 3×3 median filter) are used to reduce noise. Whenever a moving object in the scene has a color similar to the background or has an overlap with its previous frame position, some parts of the moving object are not detected as foreground regions. This results in detecting smaller regions that are fragments of a larger one. Fragments are merged iteratively based on their proximity. The small regions that are nearby, and whose contours are in intersection, are merged. A motion-based sample is an ellipse which circumscribes an area with motion.
2. *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 F additional fixed samples in the image which are generated by a uniform function and located around the center (typically $F = 16$). Samples are in large and small sizes. The largest sample is used for zooming or for object approaching the camera. Its area is 1/3 of the image area. The small samples are used for a target far from the camera and close to the center in different positions. The sizes of these

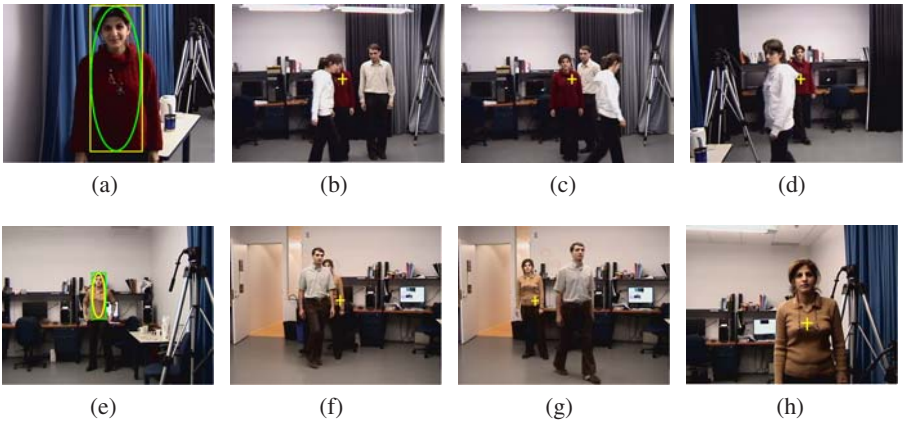

Fig. 2. Examples of tracking frames for Exp_1 (a) to (d) and Exp_8 (e) to (h). Exp_1 (a) initial model selection, (b) short-term occlusion, (c) after occlusion, (d) scale variation; Exp_8 (e) initial model selection, (f) short-term occlusion, (g) after occlusion, (h) scale variation.

elliptic samples are obtained experimentally according to the minimum face size, which is in our algorithm 5x5 pixels from frontal and lateral views.

3. *Blob filtering and modeling*: Our targets of interest are persons, thus all samples are filtered by a Bayesian skin classifier which has high true positive rate, simple implementation and minimum cost [10]. All the skin regions that contain less than 40 skin pixels (less than the half of the minimum face size) or do not contain skin regions are removed. The torso is assumed to be below the detected skin region and two times longer than the height of detected skin region. Thus, the ellipse width is the same as the skin region width, and its height is three times longer than the skin region height.

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 ($ScoreS_i$) as a sample liklihood 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.

$$ScoreS_i = \mu_{EC}.\mu_{EP}.\mu_{EH}.\mu_H. \tag{1}$$

The target is the sample with the highest score. We are using four membership functions, each with fuzzy outputs between 0 and 1:

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

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

2. The membership function μ_{EP} is utilized for Euclidean distance of S_i centroid, (x_{si}, y_{si}), from the image center,(x_{im}, y_{im}). Indeed normally, the person should be near the image center. σ^2 is equal to a quarter of the image area around the image center. It is defined as

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

3. The membership function (μ_{EH}) is applied for Euclidean distance of normalized quantized HSV color histogram of S_i, H_{si}, with the histogram of M, H_m, with n histogram bins [11]. It is computed as

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

4. Finally the membership function of μ_H is used for similarity of normalized quantized HSV color histogram of S_i with histogram of M with average of normalized histograms of $\bar{H}_{si}$ and $\bar{H}_m$ respectively [12]. It is the normalized correlation coefficient of two histograms and is defined as

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

2.2 Target Position Prediction and Camera Control

As discussed in Section 2, 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 will consider the angle between two consecutive motion vectors. If the angle difference is smaller than 25 °, it is assumed 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 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 [13].

3 Results and Discussion

We used one Sony IP PTZ camera (SNC-RZ50N) for our tests. For validation, we tested the complete system in online experiments. The algorithm is implemented on an Intel

Xeon(R) 5150 in C++ using OpenCV. 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. In the general scenario of the experiments, a target actor from the frontal view is selected for initial modeling. She starts to walk around in a room. Two or three actors can walk at the same time in different directions, crossing and occluding with the target. Fig. 2 shows the initial model selection and some frames obtained during tracking. We have done ten experiments with the IP camera and experiments are classified into two classes based on the image resolution as described in table 2. To evaluate our method, four metrics are used as explained in table 1.

Table 1. 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.
$d_{gp} = \frac{T_{OUT}}{NF}$	to evaluate the error of tracking algorithm. Ideally, d_{gp} should be zero.
$TF = \frac{TP}{TP+FP}$	to indicate the lack of continuity of the tracking system for a single target track [14]

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 coordinate, (x_c,y_c): image center, (x_p,y_p): tracked object coordinate, T_{OUT}: number of frames with target out of FOV, NF: total number of frames.

Table 2 shows the results of the four metrics with the system frame rate for all experiments. For d_{gc} and d_{gp}, we show the mean and variance of all experiments. For class 1, because of the lower system frame rate, the method has lost the target several times, but eventually recovers. 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, the system frame rate is increased because of smaller image size. Thus TF for class 2 is smaller than class 1. A faster system frame rate improves the results of TF, $\mu_{d_{gc}}$, $\mu_{d_{gp}}$ and P. The last two columns in Table 2 are the minimum and maximum length of target motion vector in number of pixels. Results show that our algorithm can handle and overcome large motion (i.e. high values of $max(\Delta x)$) because of using a repetitive target detection scheme and motion prediction technique that does 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 2, which may cause the target to exit the FOV.

Table 2. 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
E_1	97	2.3	0.1834	0.0296	0.0573	0.0047	3.02	563	10	246	L	N
E_2	96	1.5	0.1180	0.0170	0.0232	0.0009	2.98	472	1	194	L	N
E_3	88	1.8	0.2462	0.0341	0.0774	0.0043	2.75	524	3	306	L	F
E_4	92	2.1	0.1427	0.0209	0.0420	0.0018	2.64	602	2	562	L	F
E_5	70	1.7	0.3194	0.0618	0.0690	0.0044	3.1	578	12	391	L	M
class 1	**88**	**1.9**	**0.2019**	**0.0327**	**0.0538**	**0.0032**	**2.88**	**2739**	**1**	**562**	L	A
E_6	94	0.7	0.1597	0.0354	0.0302	0.0019	6.18	908	1	242	S	N
E_7	100	0	0.1101	0.0156	0.0395	0.0022	6.22	964	0	184	S	N
E_8	100	0	0.0997	0.0127	0.0215	0.0012	6.87	889	2	152	S	F
E_9	93	0.4	0.1210	0.0368	0.0282	0.0026	6.69	952	0	154	S	F
E_{10}	90	0.2	0.3577	0.0341	0.0877	0.0032	6.97	994	2	255	S	M
class 2	**95**	**0.26**	**0.1696**	**0.0269**	**0.0414**	**0.0022**	**6.57**	**4707**	**0**	**255**	S	A

$\mu_{d_{gc}}$: 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.

Generally, according to the mean of distances, the location of the target is near to the ground-truth. The target is usually localized within 1/6th of the image diagonal from the image center. With faster system frame rate the results of tracking have been improved significantly. When localization fails, it is because of similarity or closeness of the color histogram of the target with other blobs. 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 the image 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 candidate blob) until the occlusion ends. This could 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 is short because the goal the algorithm will be zooming on target face and capturing it for identification purpose.

4 Conclusion

In this paper, an upper body tracking algorithm for IP PTZ camera in online application is proposed. The proposed method consists of three main parts: image capture, upper body detection and camera control. We use a fuzzy classifier because our system has uncertainty and is nonlinear. Results show that our algorithm can handle and overcome

large motion between two consecutive frames, because it is based on combination of re-detecting the target and target position prediction at each frame. We will lose a target if the person changes its motion direction suddenly and walks very fast in the opposite predicted direction. We can recover the track if the target moves inside the FOV of the camera again. The proposed method can handle indirectly the short-term occlusion at the condition that the object stays in the FOV. We get better results with faster system frame rate.

Future work of the method will be adding camera zooming and enhancing robustness of the motion prediction to prevent the target being out of the camera FOV.

References

1. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. IEEE T-PAMI 25(5), 564–577 (2003)
2. Leichter, I., Lindenbaum, M., Rivlin, E.: Bittracker- a bitmap tracker for visual tracking under very general conditions. IEEE T-PAMI 30(9), 1572–1588 (2008)
3. Roha, M., Kima, T., Park, J., Lee, S.: Accurate object contour tracking based on boundary edge selection. Pattern Recognition 40(3), 931–943 (2007)
4. Elder, J.H., Prince, S., Hou, Y., Sizintsev, M., Olevsky, E.: Pre-attentive and attentive detection of humans in wide-field scenes. International Journal of Computer Vision 72(1), 47–66 (2007)
5. Funahashi, T., Tominaga, M., Fujiwara, T., Koshimizu, H.: Hierarchical face tracking by using ptz camera. In: IEEE Int. Conf. on Automatic Face and Gesture Recognition (FGR), pp. 427–432 (2004)
6. Bernardin, K., Camp, F., Stiefelhagen, R.: Automatic person detection and tracking using fuzzy controlled active cameras. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 1–8 (2007)
7. Li, Y., Ai, H., Yamashita, T., Lao, S., Kawade, M.: Tracking in low frame rate video: a cascade particle filter with discriminative observers of different life spans. IEEE T-PAMI 30(10), 1728–1740 (2008)
8. Kang, S., Paik, J., Koschan, A., Abidi, B., Abidi, M.: Real-time video tracking using ptz cameras. In: 6th Int. Conf. on Quality Control by Artificial Vision, pp. 103–111 (2003)
9. Wikipedia: Von luschan's chromatic scale — wikipedia, the free encyclopedia (2008), `http://en.wikipedia.org/w/index.phptitle=Von_Luschan%27s_chromatic_sca%le&oldid=249213206` (Online; accessed June 9, 2009)
10. Kakumanu, P., Makrogiannis, S., Bourbakis, N.: A survey of skin-color modeling and detection methods. Pattern Recognition 40(3), 1106–1122 (2007)
11. Cha, S.H., Srihari, S.N.: On measuring the distance between histograms. Pattern Recognition 35(6), 1355–1370 (2002)
12. Boufama, B., Ali, M.: Tracking multiple people in the context of video surveillance. In: Kamel, M.S., Campilho, A. (eds.) ICIAR 2007. LNCS, vol. 4633, pp. 581–592. Springer, Heidelberg (2007)
13. Sony corporation: Snc-rz25n/p cgi command manual, version 1.0 (2005)
14. Yin, F., Makris, D., Velastin, S.: Performance evaluation of object tracking algorithms. In: IEEE Int. Workshop on Performance Evaluation of Tracking and Surveillance(PETS) (2007)