Evaluation and Fair Comparison of Human Tracking Methods with PTZ Cameras

Alparslan Yildiz[1], Noriko Takemura[1], Yoshio Iwai[2(✉)], and Kosuke Sato[1]

[1] Osaka University, 1–3 Machikaneyamacho,
Toyonaka-shi Osaka 560-8531, Japan
[2] Tottori University, 4-101 Koyamacho-minami,
Tottori-shi Tottori 680-0945, Japan
iwai@ike.tottori-u.ac.jp

Abstract. Evaluation and comparison of methods, repeatability of experiments, and availability of data are the dynamics driving science forward. In computer vision, a database with ground-truth information enables fair comparison and facilitates rapid improvement of methods in a particular topic. Being a high-level discipline, Human-Computer Interaction (HCI) systems rises on numerous computer vision building blocks, including eye-gaze localization, human localization, action recognition, behavior analysis etc. using mostly active systems employing lasers, projectors, infrared scanners, pan-tilt-zoom cameras and other various active sensors.

In this research, we focus on fair comparison of human tracking methods with active (PTZ) cameras. Although there are databases on human tracking, no specific database is available for active (pan-tilt-zoom) camera human tracking. This is because active camera experiments are not repeatable, as camera views depend on previous decisions made by the system. Here, we address the above problem of systematical evaluation of active camera tracking methods and present a survey of their performances.

1 Introduction

Considering the large number of human tracking methods in the literature, there is a clear need for comparative evaluation of similar methods. A database of natural human movements with ground truth information makes it possible to compare methods and allows researchers to improve the performance of their algorithms more rapidly [4,5,9]. For fair comparison, all methods should be evaluated on the same data and the experiments should be repeatable. In human tracking methods, this requirement is highly dependent on the imaging hardware. Based on hardware specialization, we can divide the human tracking methods into two sub categories: *static camera tracking* and *PTZ camera tracking*. Static cameras, in our context, have a fixed position and orientation throughout the experiments. PTZ cameras, on the other hand, adaptively alter their orientations to capture more targets while their positions remain fixed.

R. Shumaker and S. Lackey (Eds.): VAMR 2015, LNCS 9179, pp. 153–161, 2015.
DOI: 10.1007/978-3-319-21067-4_17

In the case of static cameras, it is relatively easy to evaluate competing methods using a benchmark database such as those in [4,5,9] These databases usually consist of recorded camera view images and manually or automatically marked locations of humans on all or a subset of frames. During evaluation, the same camera images are fed into the evaluated methods and the outputs are compared with the ground truth data. The Human-Eva database [9] contains videos of articulated human motion and provides a comparative basis for accurate human pose estimation and motion tracking. The POM Pedestrian dataset [5] includes multiple camera recordings of pedestrians. Human locations in some of the videos in this database are manually marked to provide a basis for comparison of human tracking methods. Similarly, the PETS-2006 database [4] contains more natural movements of people in real environments.

The evaluation of competing methods for PTZ camera tracking problems, on the other hand, is not a trivial task. PTZ camera tracking methods consist of both camera reconfiguration and human tracking. The main problem with the former is repeatability. With PTZ cameras, the camera view image at any time-step depends on the actions of the PTZ cameras on all previous time-steps. Simply recording camera images is not sufficient for evaluation purposes. To the best of our knowledge, there is no publicly available database for evaluating PTZ camera reconfiguration methods. In this research, we present a method for generating repeatable PTZ camera reconfiguration experiments using real data. Our method takes static camera images and generates geometrically consistent virtual views of the PTZ cameras for any pan/tilt/zoom (PTZ) configuration. This is achieved with minimum calibration of the cameras; only the rectifying homography of the ground plane is necessary. The rectifying homography is also required in evaluations with static cameras, so it is usually available. Without requiring any additional user input, we can produce consistent PTZ camera views from static images, and evaluate competing PTZ camera reconfiguration methods on the same data. Synthetic PTZ cameras produced this way will have the camera center same as the original static cameras. This is actually desirable, because static cameras used to create human tracking databases would be located in a way that they would capture large areas and general views of the scene.

We convert a human tracking evaluation database that is captured with static cameras into an evaluation database for PTZ camera reconfiguration. The input to our system is the static camera images and the rectifying homography for the ground plane of each camera, which is readily available for the original static camera databases. The evaluation database is generated online with negligible computational cost. Our method, in a sense, simulate PTZ cameras given the recordings of wide angle static cameras. Camera positions and scene setup is naturally limited to the static camera positions. We do not see this as a limitation since our main objective here is to make repeatable experiments for PTZ camera reconfiguration methods. For all competing methods, our method provides the same camera images and diversity in camera views and camera positions can be achieved by using different static camera tracking databases.

Our method generates virtual PTZ cameras on desired pan/tilt speed or ranges. We do not have any limiting requirements for the static camera database as well.

However, one desirable property of static cameras of the input database would be that the camera views should not have very small field-of-views. In order to generate *meaningful* outputs, static camera views should be able to view at least a few people at the same time, so that we can generate virtual views using small portions of that view to track individuals with virtual PTZ cameras. Human tracking databases such as PETS-2006 [4] and POM [5] already satisfy this simple requirement. Other properties of the static camera databases such as image resolution, frame-rate, color quality, lighting quality, exact value of the field-of-view etc. are not major interests of this research, since all evaluated algorithms on these databases will use the same virtual views generated by our method. These properties may affect individual algorithms that are evaluated, however this is outside the scope of this research. We mainly aim to provide repeatable and fair experiments for PTZ camera reconfiguration methods.

For experimental evaluation in this research, we use well-known human tracking databases such as PETS-2006 and POM to generate the PTZ camera views. In this way, the natural movements of humans are reflected in the PTZ camera experiments, and the ground truth human locations on static camera images are translated to PTZ camera frames for evaluation.

2 PTZ Camera Reconfiguration Methods

We generated PTZ camera evaluation databases from POM and PETS-2006 videos, and compared several PTZ camera reconfiguration methods with out method in terms of performance. These methods, together with a brief explanation of their underlying algorithms, are listed below. The bold headings are our chosen abbreviations for future references to these methods in this paper.

Bidding. In a recent study, Li *et al.* [8] presented a tracking system for multiple PTZ cameras. They formulated PTZ camera reconfiguration as an assignment problem, where each target in the scene is assigned to a camera. The assignment problem is solved by an *auction* approach. For each target, each camera provides a *bid* on how well the camera can track the target. Once the bids have been collected, each target is assigned to a camera in such a way as to maximize the total bid.

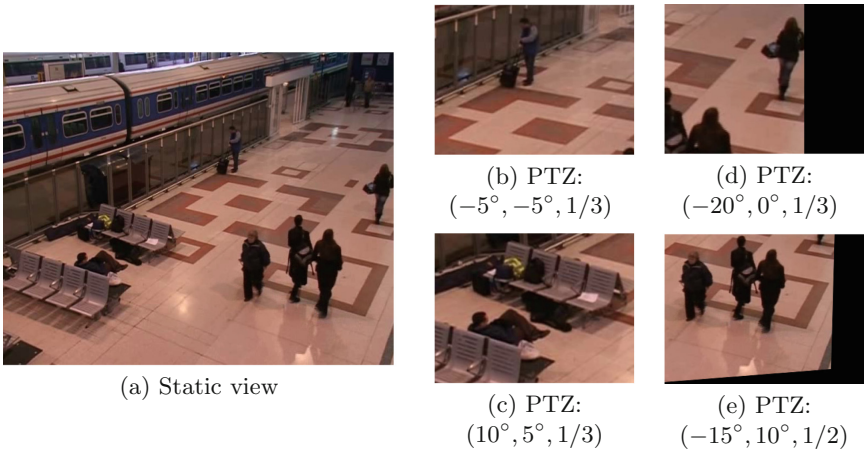
Earliest. Costello *et al.* [3] formulated PTZ camera reconfiguration as a scheduling problem and utilized well-known scheduling policies to schedule the PTZ cameras. They reported the *earliest deadline* policy as the most successful one. This policy tries to maximize target coverage by scheduling the PTZ cameras to track those targets that are expected to leave the scene the soonest. In this way, more tracking time can be allocated to future targets.

MI. Sommerlade *et al.* [10] presented a probabilistic surveillance method for multiple PTZ cameras. In their method, the pan/tilt configuration of each camera is optimized to maximize mutual information. Multiple PTZ cameras are considered and optimized simultaneously.

Motion. Konda et al. [7] formulated coverage of targets as an assignment problem. Each camera is assigned either as *global* or *target* camera. While *global* cameras ensure the general coverage of the scene, *target* cameras are assigned to individual targets.

CFA. Munishwar et al. [1] presented a series of algorithms for multi-camera object coverage. In their study, they define a *force* between each target and possible pan configurations of each camera as the fitness for camera-target assignment. Finally, given the attraction of the computed forces, each camera is assigned to a pan configuration in a greedy fashion.

Occupancy. In the previous work [11] we devised a PTZ camera reconfiguration system that does not require the detection of targets in camera views. By registering each PTZ camera, we compute the ground occupancy maps for PTZ cameras and optimize camera configurations directly on the occupancy map. We also utilize multiple time-step estimations for better camera configuration decisions.



(a) Static view

(b) PTZ: $(-5°, -5°, 1/3)$

(d) PTZ: $(-20°, 0°, 1/3)$

(c) PTZ: $(10°, 5°, 1/3)$

(e) PTZ: $(-15°, 10°, 1/2)$

Fig. 1. Sample virtual views for a synthesized PTZ camera: (b), (c), (d), and (e) show the view of the virtual PTZ camera with varying (*pan*,*tilt*, *zoom*) configurations. The unit for pan and tilt options is degrees, and the unit for the zoom option is scaling relative to the original field-of-view in (a).

3 Calibration of Camera Time-Step

Using PTZ cameras, we are bound to make decisions in advance, because a PTZ camera will issue a pan/tilt command not instantly but by spending a short amount of time. The natural assumption is that, the time spent by the camera to perform an action is not random and can be modeled. First, let us consider the case where we have modeled the camera latency properly and we know how

much milliseconds it would take to issue a given amount of pan/tilt operation. In this case, all our formulations using future time-steps would imply this calibrated latency value. Depending on the environment and scenario, a pan/tilt step may simply mean 5-10 degrees of movement around the camera center. Let us say, a pan step is 5 degrees of movement and 5 degrees of pan movement would take 100 ms for the camera to perform. Then in this case, the *next time-step* is simply 100 ms ahead of current time.

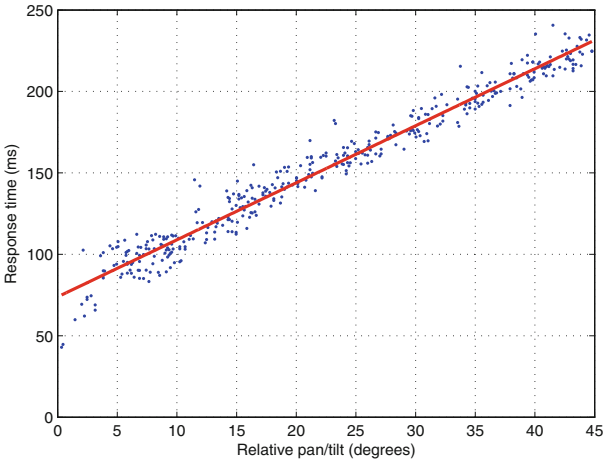


Fig. 2. Camera latency data

It is relatively easy to calibrate a PTZ camera for its latency and normalize time-steps using the latency information. For our Sony D-100 [2] cameras, we have measured pan/tilt latencies for various amount of rotations and the scatter data in Fig. 2 is gathered. The relationship is clearly modeled by a linear function.

For the rest of this work, the calibration of camera time-step is implicit and future time-steps are discretized by intervals of pan/tilt steps. For instance, next time-step is when the camera can complete a single pan/tilt step. Similarly, two time-steps into the future is when the camera can complete two pan/tilt steps. The length of pan/tilt step in angles is usually 5 or 10 degrees depending on the experimental environment.

4 PTZ Camera Synthesis

In this section, we describe our method for generating consistent views for PTZ cameras from static camera views. Fig. 1 demonstrates this process.

Initially we only require the rectifying homography H for the static camera view. This homography maps image coordinates to ground plane coordinates. For human tracking databases, such as PETS-2006, H is usually available because

tracking methods depend heavily on it. Otherwise, it can be computed by manually providing four point correspondences. Although we describe our method for a single camera, it can readily be applied to multiple cameras.

PTZ camera views are simply rotated/zoomed views of the static camera with a smaller field-of-view. The rotation is around the camera center, and thus, can be performed with a linear image transformation once the camera matrix K is computed. Zoom and viewpoint alterations are possible by similarly utilizing the camera matrix. With some care, we can compute K directly from H. We begin by extracting vanishing points on the ground plane from H using the following equations:

$$Hv_x = [1\ 0\ 0]^T, \tag{1}$$

$$Hv_y = [0\ 1\ 0]^T. \tag{2}$$

Because the ortho-center of the vanishing points is the principle point, c, on the image plane, we can compute the last vanishing point v_z by constructing a triangle of vanishing points from v_x, v_y, and c. Initially, we estimate c as the center of the image, $(w/2, h/2)$. This gives us a very good estimate of v_z. Next, we collect edges in the direction of v_z and re-estimate v_z from these edges and c from the new set of vanishing points.

Finally, we can compute K directly from the three vanishing points [6]. Any rotation around the camera center can now be represented as an image homography, $H_r = KRK^{-1}$, where R is the rotation matrix in 3D. New views from the static camera view are computed by perspectively warping the image with H_r followed by a clipping and scaling around c to adjust the field-of-view. If we represent the clipping by homography H_{fov}, the rectifying homography for the new view can be given as $H_{new} = HH_r^{-1}H_{fov}^{-1}$.

Given a static camera view and the corresponding H, we compute PTZ camera views by applying the perspective warp $T = H_{fov}H_r$ to the static camera view. In practice, we precompute and store T and H_{new} matrices for each PTZ camera configuration.

Fig. 1 illustrates sample outputs of our PTZ camera view synthesis. See the figure caption for details.

4.1 PTZ Camera Synthesis Discussion

Main input to our view synthesis method is a human tracking database recorded with static cameras. We also require the ground rectifying homographies for each view. Generally these homographies are readily present for tracking databases, however they can easily be computed by manually marking 4 points on the view of the ground plane. Given recorded video of a static camera and its ground rectifying homography, our method synthesizes views of a virtual PTZ camera for any given pan, tilt and zoom configuration. New view generation consists of a planar image warping and is performed online with virtually free of computational cost. Thus, we do not need to store any external files for the newly generated database as new views are generated on demand with high consistency and speed.

Necessary delay is applied automatically consistent with the simulated PTZ camera. Computing the amount of delay is discussed in Sect. 3. During benchmarking, our system computes the required delay for a pan/tilt/zoom command and provides new images for PTZ camera reconfiguration methods only after the required delay has elapsed.

5 Experiments

We implemented the PTZ camera methods described in Sect. 2 using the information available in respective papers. We manually optimized the necessary parameters of the methods to obtain the best accuracy for our implementations and compared their performance on our synthetic PTZ camera databases created from the PETS-2006 and POM databases. Although the PETS-2006 database has a relatively low human density in the videos, the human movements in this database are quite natural. Conversely, the POM database includes unnatural human movements with a higher human density than in the PETS-2006 database. We chose these two databases because of the contrast in their density and motion properties.

While the POM database provides ground truth locations of humans for some intervals, the PETS-2006 database does not have any ground truth information. Thus, we marked the intermediate frames of the POM database and all the frames of the PETS-2006 database to provide a basis for fair comparison of the PTZ camera reconfiguration methods.

In all our experiments, we compared the competing methods in terms of accuracy and execution times. The unit of accuracy is *coverage*, which is defined as the ratio of the number of targets in the camera view(s) to the total number of targets in the scene. The unit of execution time is milliseconds per frame. We computed accuracy and execution times for all frames and report the mean values.

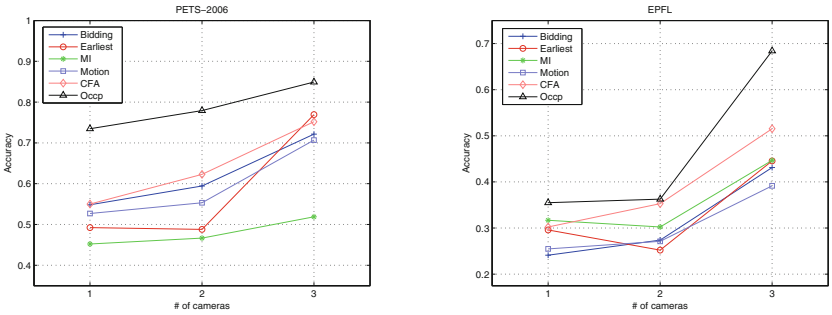
Table 1. Accuracy of different methods on PETS-2006

N-cams	Bidding [8]	Earliest [3]	MI [10]	Motion [7]	CFA [1]	Occp. [11]
1	0.5483	0.4922	0.4521	0.5268	0.5501	0.7347
2	0.5944	0.4879	0.4665	0.5532	0.6227	0.7792
3	0.7214	0.7693	0.5187	0.7069	0.7519	0.8493

Tables 1 and 2 give the accuracy of the PTZ camera methods on the PETS-2006 and POM databases, respectively. It is evident that the POM database is a more complex database for multiple PTZ cameras because the human density is relatively high. In contrast, the PETS-2006 database includes videos with low human density and PTZ cameras can capture these people with relatively high accuracy. In Fig. 3 the results are shown as graph plot. It is more evident on the graph that the method we devised outperforms other competing methods.

Table 2. Accuracy of different methods on POM

N-cams	Bidding [8]	Earliest [3]	MI [10]	Motion [7]	CFA [1]	Occp. [11]
1	0.2411	0.2958	0.3167	0.2547	0.3023	0.3549
2	0.2737	0.2521	0.3022	0.2711	0.3528	0.3627
3	0.4312	0.4456	0.4467	0.3914	0.5153	0.6842



(a) PETS-2006 dataset (b) POM dataset

Fig. 3. Accuracy of different methods on PETS-2006 and POM datasets

The results in Tables 1 and 2 give a fair comparison of the competing PTZ camera methods. Thus, we can safely conclude that, while in relatively low density scenarios with natural human movements (PETS-2006 database) our *Occupancy* method outperforms other methods, in a more dense scenario with unnatural human movements (POM database) all methods perform similarly.

Table 3. Execution times of different methods (ms/frame)

N-cams	Bidding [8]	Earliest [3]	MI [10]	Motion [7]	CFA [1]	Occp. [11]
1	108.01	108.78	125.55	122.06	98.06	10.64
2	180.27	171.08	265.10	218.92	175.62	17.72
3	250.34	210.56	630.22	341.77	223.67	32.76

Table 3 gives the execution times of the evaluated methods in milliseconds per frame. Note that different methods utilize different types of optimizations. For instance the *Bidding* method makes decisions for all cameras at the same time, whereas the *Earliest* method makes independent decisions for each camera, thus running faster. In such cases, making more accurate decisions appears to be vital, however, making faster decisions allows the methods to make faster acquisitions. A very slow running PTZ camera method can receive new information only at larger intervals, thus reducing its awareness of the environment.

6 Conclusions

There are publicly available databases for the evaluation of various vision tasks, such as stereo, optical flow, human tracking, and so on. These databases make it possible for fair comparison of competing methods, which encourages research and allows algorithms to evolve faster with increasing accuracy and speed. However, no evaluation database is available for PTZ camera reconfiguration. The reason for this is that PTZ camera experiments are not repeatable in a straightforward manner.

In this study, we devised a simple method for generating PTZ camera views from static camera views on demand to evaluate PTZ camera reconfiguration methods. The original databases included both natural and controlled human motion, which is desirable for the evaluation of competing methods. We tested several methods from the PTZ camera tracking literature and compared with our PTZ camera reconfiguration method [11].

References

1. Abu-Ghazaleh, V.M.N.: Coverage algorithms for visual sensor networks. ACM Trans. Sens. Netw. **9**(4), 1–31 (2013)
2. Camera, S.E.D.P.: http://pro.sony.com/bbsc/ssr/cat-industrialcameras/catrobotic/product-EVID100/
3. Costello, C.J., Diehl, C.P., Banerjee, A., Fisher, H.: Scheduling an active camera to observe people. In: Proceedings of the ACM 2nd International Workshop on Video Surveillance and Sensor Networks. pp. 39–45 (2004)
4. Data, P.B.: http://www.cvg.rdg.ac.uk/~pets2006/data.html
5. Fleure, F., Berclaz, J., Lengagne, R., Fua, P.: Multicamera people tracking with a probabilistic occupancy map. IEEE Trans. Pattern Anal. Mach. Intell. **36**(8), 1614–1627 (2008)
6. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, Cambidge (2004)
7. Konda, K.R., Conci, N.: Real-time reconfiguration of ptz camera networks using motion field entropy and visual coverage. In: Proceedings of the International Conference on Distributed Smart Cameras (2014)
8. Li, Y., Bhanu, B.: Camera pan/tilt control with multiple trackers. In: International Conference on Pattern Recognition (2012)
9. Sigal, L., Balan, A.O., Black, M.J.: Humaneva: synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. Int. J. Comput. Vis. **87**(1–2), 4–27 (2010)
10. Sommerlade, E., Reid, I.D.: Probabilistic surveillance with multiple active cameras. In: Proceedings of the IEEE International Conference on Robotics and Automation May 2010
11. Yildiz, A., Takemura, N., Hori, M., Iwai, Y., Sato, K.: Tracking people with active cameras using variable time-step decisions. IEICE Trans. Inform.Sys. **E97**(8), 1952–2216 (2014)