Point Matching for PTZ Camera Auto-tracking

Li Sun, Quentin De Neyer, Christophe De Vleeschouwer
ICTEAM, Université catholique de Louvain, Belgium

***Abstract*—This paper considers the problem of PTZ camera auto-tracking. The system is composed of a controller and a tracker, which follows the target in consecutive frames captured by the camera. Since the target might potentially cover a large fraction of the image or might even be only partly visible, the tracking is done through points matching, using local features instead of global ones. Using a PI control scheme with adaptive gain factors, the controller steers the camera towards the target by smoothly changing the pan, tilt and zoom parameters of the camera. Results demonstrate the relevance of the approach in the typical surveillance application.**

I. Introduction and Overview

Most of the existing surveillance systems employ one or several fixed view cameras. The finite number of such static cameras limits the total area and the spatial resolution of the coverage. Furthermore, such systems are relatively passive, which means that they can not actively keep focusing on some particular targets of interest in the scene. Motorized cameras provide an alternative solution to those drawbacks. Most related works have investigated the use of motorized PTZ cameras to detect and track an intruding target. Murad *et al.* [1] propose to track the position of the target in the camera referential using an Extended Kalman Filter (EKF). A command is then sent to point the PTZ towards the estimated target position. Varcheie *et al.* [2] track people based on the fuzzy classification of color histograms, and point the PTZ directly towards the estimated target position. Both these works do not consider the smoothness of the camera displacement during the control, and can only handle pictures containing the entire target, which limits the zooming capabilities.

PTZ auto-tracking raises two problems, which are target position estimation and smooth camera steering. Accordingly, our proposed system has two major components: the *tracker* and the *controller*. In contrast to the above previous works, the proposed method treats the image tracking and the PTZ steering in a closed control loop. In that way, instead of directly commanding the parameter according to the directive, our system continuously adjust the speed of variation of the pan and tilt, and increments or decrements the zoom factor by very small steps. This allows for a better control of the smoothness of the camera displacements. The overview flowchart of the proposed control loop is shown in Figure 1. The image processing block estimates the 2D position of the target in the image. In order to deal with large zooming factors, this is done by matching points in two successive images, based on the texture observed in their local neighborhood. Then the controller steers the pan/tilt/zoom motion based on the position and size of the target, as given by the tracker.

The controller and the tracker are presented in II and III, respectively. We conclude with a few results in IV.

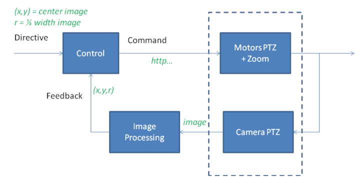

Fig. 1. Motorized PTZ control system.

II. PTZ commands and PI control

Off-the-shelf PTZ cameras (such as the P5512 by AXIS) give the opportunity to control either the actual PTZ parameters of the camera, or the speed of variation of those parameters. In order to change the pan/tilt/zoom parameters with a constrained smoothness, we have chosen to control the speed of the pan/tilt (in degrees/sec), and to restrict the variation of zoom factor to the smallest available variation at each step of the discrete control process. In practice, if the height of the target is estimated by the tracker to be smaller than a predefined threshold T_l, the camera zooms in, else if it is larger than T_h it zooms out, otherwise it dose not change the zoom factor.

Regarding the pan/tilt motion, the well-known PI feedback controller has been considered. We detail the control of the pan parameter. Extension to the tilt parameter is straightforward. The input of the pan controller is computed by the tracker described in Section III, and estimates the distance in pixels between the center of the target and the center of the image. We are thus first interested in deriving the relation that translates an image pixel displacement into a displacement of angle, so as to make the controller work in a uniform unit system. Let $\mathrm{d}x_\alpha(\mathrm{d}\theta)$ and $\mathrm{d}y_\alpha(\mathrm{d}\theta)$ denote the horizontal and vertical shifts of the image plane projection of a 3D point, when the zoom factor of the camera is set to α. The exact relation between the coordinates of the projected pixels, as measured before and after the panning, can be derived by multiplying the projection matrix (depending on α, via the focal length $f_x(\alpha)$, measured in unit of pixels along the x axis) by a rotation matrix (for $\mathrm{d}\theta$). Using Taylor's approximation of the rotation factors, we easily derive the shifts.

$$\mathrm{d}x_\alpha(\mathrm{d}\theta) \approx \mathrm{d}\theta f_x(\alpha) + \mathrm{d}\theta^2 x + \mathrm{d}\theta \frac{x^2}{f_x(\alpha)} \approx \mathrm{d}\theta f_x(\alpha) \quad (1a)$$

$$\mathrm{d}y_\alpha(\mathrm{d}\theta) \approx \mathrm{d}\theta \frac{xy}{f_y(\alpha)} \approx 0 \quad (1b)$$

Given this relation, we now explain how to adapt the pan speed $\mathrm{d}\theta/\mathrm{d}t$ of the camera, based on a PI controller. The PI combines 2 weighted terms, which are proportional to the error and the integral of the error, respectively. This can be represented by (2). $e_x(t)$ is the error signal measured in deg/sec at time t. This error is calculated by $e_x(t) = (x_c(t) - x_i(t))/f_x(\alpha)$, where $x_i(t)$ is the estimated horizontal target position output from the tracker, and $x_c(t)$ is the center position of the image (directive). Here, the denominator $f_x(\alpha)$ is required to translate the image pixel displacement in degrees of pan angle, and adapts the gains of the scheduler to α. The K_p and K_i are defined based on the accuracy and stability of the system, as discussed in Section IV.

$$\mathrm{d}\theta/\mathrm{d}t = K_p e_x(t) + K_i \int_t e_x(t)\mathrm{d}t \tag{2}$$

III. Points matching for auto-tracking

The proposed approach builds on the matching of points between consecutive frames of the video sequence to infer the displacements of the target of interest in the image referential. The basic idea consists in describing the object through a set of points characterized by their associate local neighborhood representation, and then in finding correspondences between consecutive frames, based on the matching of points in their neighborhoods. Assuming small 3D displacements between consecutive images, it is possible to restrict the search for correspondence to a search window around the initial position of the point, compensated to account for the camera motion. A matching metric is used to compare the local neighborhood of the reference point with the one of every candidate of the search window, generating a distance map. Then, a weighted sum of all distance maps (one for each point) is computed to infer the target displacement. The global displacement of the target between the two frames, that will guide the control of the camera, is computed as the minimum of this weighted sum. The weight associated to a point depends on two factors. The first one measures the likelihood that the point belongs to the target. In practice, it is estimated based on the past observations of the consistencies between the point distance map and the estimated target and background motions. The second factor reflects the point matching reliability. It increases when the texture around the point is such that the matching remains correct, even in case of small local deformations. In practice, this factor is estimated through the off-line training, which computes distance maps between reference and deformed texture patches to learn how the matching correctness is affected by some of the texture features. Typically, a reliable or non-ambiguous point matching corresponds to a distance map that remains peaked despite small deformations. In addition to weighting the distance map, this 'peakedness' factor also supports a dynamic update of the points included in the target model. Points with ambiguous matching are progressively removed, while new points are incorporated into the model. Finally, the average position of the points, that are likely to be in the target, is used as the target location, and the envelope of those is used to estimate the target height.

IV. Results and Conclusion

Two kinds of setups are considered to assess our control system. The first setup is depicted in Figure 2. The PTZ camera lies in front of a screen on which a scene is projected. The scene can be a natural or synthetic scene, but is composed of some visual object in motion. The PTZ camera has to capture close-shot images of the object moving on the screen. The advantage of this setup is that it offers reproducible experimental conditions, which is quite useful to tune the parameters of the PI controller. It also allows to consider simple object appearance and motion scenarios, which allows to assess the controller independently of the tracker limitations.

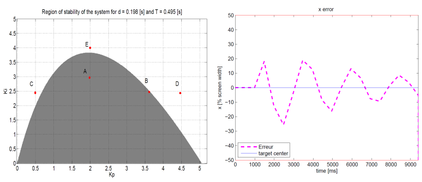

Fig. 2. The stability region of K_p and K_i is shown on the left figure of the bottom row, while the error with point A is shown on the right.

This setup has been used to study the system stability. Given a system sampling period T and a delay d, the transfer function of our system (which is supposed to be Linear Time-Invariant) can be represented in the z-domain, and the value range of (K_p, K_i) that generates the stable results can be derived by jury test conditions based on the transfer function. The stability region is shown in gray in Figure 2. For the point A in the stability region with $(K_p, K_i) = (1.94, 2.9)$, the error of the step response for the synthetic object is shown as the pink curve in Figure 2. A deeper assessment of the tracking performance (e.g. the rise time, overshoot, steady state error) has been accomplished based on this setup.

In contrast, the second setup considers real-life scenarios. In this case, there is little control on the reproducibility of the scenario. This assessment evaluates whether the system most often works or not, given the constraints of a real-life environment. Figure 3 presents the distribution of points considered by the point matching tracker in the case of a real-life deployment. We observe that the number of points is relatively small, so as to maintain the frame rate reasonable (around 5 fps). Demo videos have been uploaded as the supplementary material, and a running system will be deployed during the conference.

Fig. 3. Selected frames for video surveillance.

References

[1] M. Al Haj, A.D. Bagdanov, J. Gonzalez, and F.X. Roca, "Reactive object tracking with a single ptz camera," in *ICPR*. IEEE, 2010, pp. 1690–1693.

[2] P.D.Z. Varcheie and G.A. Bilodeau, "Active people tracking by a ptz camera in ip surveillance system," in *ROSE*. IEEE, 2009, pp. 98–103.