Pan-Tilt-Zoom Camera Networks

A. Del Bimbo a F. Dini a A. Grifoni b F. Pernici a

a MICC University of Florence, Florence, Italy
b Thales Italia, Florence, Italy

Abstract

Pan–Tilt–Zoom (PTZ) camera networks play an important role in surveillance systems. They can direct attention to interesting events that occur in the scene. One method to achieve such behavior is a process known as sensor slaving: one master camera (or more) monitors a wide area and tracks moving targets to provide the positional information to one slave camera (or more). The slave camera can thus foveate at the targets in high resolution.

In this chapter we consider the problem of estimating online the time-variant transformation of a human’s foot position of a fixed camera relative to his head position in the image of PTZ camera. The transformation achieves high-resolution images by steering the PTZ camera at targets detected in a fixed camera view. Assuming a planar scene and modeling humans as vertical segments, we present the development of an uncalibrated framework that does not require any known 3D location to be specified, and that takes into account both zooming camera and target uncertainties. Results show good performances in localizing the target’s head in the slave camera view, degrading when the high zoom factor causes a lack of feature points. A cooperative tracking approach exploiting an instance of the proposed framework is presented.

Keywords: pan tilt zoom camera, rotating and zooming camera, camera networks, distinctive keypoints, tracking

8.1 Introduction

In realistic surveillance scenarios, it is impossible for a single sensor to monitor all areas at once or to acquire interesting events at high resolution. Objects become occluded by trees and buildings or by other moving objects, and sensors
themselves have limited fields of view. A promising solution to this problem is to use a network of PTZ cameras. A PTZ camera can be conceived as reconfigurable and fixed camera. In fact, through the pan, tilt, and zoom parameters, we can actually choose from a family of fixed cameras. The zoom capabilities of those cameras allow placement away from the scene to monitor, while the pan and tilt can be used to cooperatively track all objects within an extended area, and seamlessly track, at higher resolution, individual objects that could not be viewed by a single not zooming sensor. This allows for example, target identity to be maintained across gaps in observation.

Despite this huge potential, the complexity of such systems is considerable. We argue that three main issues worth exploring since they form the basic building blocks to create high level functionalities in PTZ-camera networks: (1) control laws for the trajectories of the PTZ system with respect to a given task; (2) determining the mapping between one camera and the PTZ system (master–slave) and (3) the scaling of a minimal system (i.e., two cameras) to one that includes multiple omnidirectional and PTZ systems.

In this chapter the focus is on establishing the mapping between a fixed camera and the PTZ camera. Once known, the mapping between a fixed and an active camera greatly simplifies peripherally guided active vision of events at a site.

To this end, cameras are settled in a master–slave configuration [1]: the master camera is set to have a global view of the scene so that it can track objects over extended areas using simple tracking methods with adaptive background subtraction. The slave camera can then follow the trajectory to generate closeup imagery of the object.

Despite the possibility of arranging an active and a fixed camera with a short baseline so as to promote feature matching between the two fields of view as in [1,2], here the aim is to propose a general framework for arbitrary camera topology. In our framework, any node in the camera network sharing a common FOV can exploit the mapping.

The proposed solution is able to estimate online the transformation between a fixed master camera viewing a planar scene and tracking the target, and a PTZ slave camera taking closeup imagery of it. The system computes the target foot position in the fixed camera view and transfers it to the correspondent head position in the PTZ camera view. Both the camera and target uncertainties are taken into account in estimating the transformation between the two cameras. Since we do not explicitly use direct camera calibration, no large offline learning stage is needed.
8.2 Related Work

A substantial number of papers in the literature concerning multi-camera surveillance systems had focused on object detection (Dalal and Triggs [3]), target tracking (Yilmaz et al. [4]) and data association for multiple target tracking (Vermaak et al. [5]). The importance of accurate detection, tracking, and data association is obvious, since tracking information is needed as the initial stage for controlling one or more PTZ cameras to acquire high-resolution imagery. However, in addition to detection and tracking, the acquisition of high-quality imagery, particularly for biometrics purposes, requires accurate calibration between the fixed and PTZ cameras in order to focus attention on interesting events that occur in the scene. For these cameras, however, precalibration is almost impossible. In fact, transportation, installation, and changes in temperature and humidity as present in outdoor environments typically affect the estimated calibration parameters. Moreover, it is impossible to recreate the full range of zoom and focus settings. A trade-off has to be made for simplicity against strict geometric accuracy and between online dynamic and offline batch methodologies.

A step in this direction has been made by combining fixed and PTZ cameras. This configuration is often termed in the literature as master–slave. Many researcher, use master–slave camera configuration with two [1,6,7,2,8,9] or more [10–15] cameras. In particular, most of the methods strongly rely on a direct camera calibration step [8,7,13,14]. Basically these approaches are not autonomous since they need a human to deal with calibration marks.

The few exceptions are discussed [9], [1] and [2]. Kang et al. [9] track targets across a fixed and a PTZ camera. They use an affine transformation between consecutive pair of frames for stabilizing moving camera sequences, and an homography transformation for registering the moving and stationary cameras with the assumption that the scene is planar. No explicit zoom usage is performed. The works Zhou et al. [1] and Badri et al. [2] do not require direct calibration; however viewpoints between the master and slave cameras are assumed to be nearly identical so as to promote feature matching. In particular in Zhou et al. [1] use linear mapping computed from a lookup table of manually established pan and tilt correspondences. They actively control a zooming camera using a combination of motion detection algorithms. The Badri et al. [2] use a lookup table that takes the zoom into account, but still needs a batch learning phase.

More general methods exist to calibrate one or several PTZ cameras. These methods can be classified depending on the particular task performed. Since PTZ cameras, if they are stationary, play the same role as fixed cameras, standard methods for fixed cameras still apply. From the very definition of camera
calibration (internal and external), standard camera calibration techniques (sometimes also termed hard calibration) can be used. Past work on active camera calibration has mostly been done in a laboratory using calibration targets at least in controlled environments. One important work using active zoom lens calibration is Willson et. al. [16]. However, these methods are not flexible enough to be used in wide areas (especially in outdoor environments), because image measurements have to be well spaced in the image.

A more flexible method Hartley [17] can be used to self-calibrate (without calibration targets) a single PTZ camera by computing the homographies induced by rotating and zooming. In de Agapito et al. [18] the same approach was analyzed considering the effect of imposing different constraints on the intrinsic parameters of the camera. It was reported that best results are obtained when the principal point is assumed to be constant throughout the sequence although it is known to be varying in reality. In Sinha and Pollefeys [19] a very thorough evaluation of the same was method performed with more than one hundred images. Then the internal calibration of the two PTZ camera was used for 3D reconstruction of the scene through essential matrix and triangulation, using the mosaic images as a stereo pair.

A class of methods exploiting moving objects in order to achieve self-calibration also appears in the literature. For example Davis and Chen [20] and Svoboda et al. [21] use LEDs. As the LED is moved around and visits several points, these positions make up the projection of a virtual object (3D point cloud) with an unknown 3D position. However, camera synchronization here limits the flexibility of these approaches especially if they are applied to IP camera networks.

A further class of methods performs a weaker (self–) calibration by establishing a common coordinate frame using walking subjects. Sinha et al. [22] uses silhouettes of close–range walkers; Chen et al. [23] and Lee et al. [24] use far distance moving objects; and [25] use planar trajectories of moving targets.

The very last category over-simplifies camera self-calibration, maximizing flexibility over accuracy. Krahastoeover et al. [26] Lv et al. [27] and Bose and Grimson [28] especially belong to this category. They use basically single–view–geometry camera calibration (Liebowitz and Zisserman [29]) of pinhole cameras through vanishing points computed from inaccurate image features such as the imaged axis of walking people. A main problem with these approaches is that parallel lines used to compute vanishing points have to be viewed with a strong perspective effect. Moreover, the measured features computed using target/motion detection methods are in general very noisy for the estimation of the underlying projective geometric models.

However none of the presented methods provides a way to maintain calibration
PTZ camera network research can also benefit from simultaneous localization and mapping (SLAM) literature using visual landmarks Checkov et al. [30] and Barfoot [31]. In Se et al. [32] landmarks are detected in images using scale-invariant feature transform (SIFT), matched using efficient best–bin–first K-D tree search (Beis and Lowe [33]). With a similar method Barfoot, [31] is able to process pairs of 1024×768 images at 3Hz with databases of up to 200,000 landmarks. This result is even more encouraging because of recent efforts in local image descriptors [34–37]. Zhang et al. [34] propose a parallel SIFT implementation in 8-core system showing an average processing speed of 45 FPS for images with 640×480 pixels, which is much faster than implementation on GPUs (Sinha et al. [35] and Grabner et al. [36]). Finally, Brown et al. [37] demonstrates that descriptors with performance equal to or better than state–of–the–art approaches can be obtained with 5 to 10 times fewer dimensions.

8.3 Pan–Tilt–Zoom Camera Geometry

In this section, we give an overview of the mathematics (used by our proposed framework) by reviewing the basic geometry of PTZ cameras. [18].

The projection of scene points onto an image by a perspective camera may be modeled by the central projection equation

\[ x = P X, \]

where \( x = [x, y, 1]^T \) are the image points in homogeneous coordinates, \( X = [X, Y, Z, 1]^T \) are the world points, and \( P \) is the 3×4 camera projection matrix. Note that this equation holds only up to scale. The matrix \( P \) can be decomposed as

\[ P = K[R|t] \tag{8.1} \]

where the rotation \( R \) and the translation \( t \) represent the Euclidean transformation between the camera and the world coordinate systems and \( K \) is an upper triangular matrix that encodes the internal parameters of the camera in the form

\[ K = \begin{pmatrix} \gamma f & s & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{pmatrix} \tag{8.2} \]

Here \( f \) is the focal length and \( \gamma \) is the pixel aspect ratio. The principal point
is \((u_0, v_0)\), and \(s\) is a skew parameter, which is a function of the angle between the horizontal and vertical axes of the sensor array. Without loss of generality, by choosing the origin of the camera reference frame at the optic center, the projection matrix for each possible view \(i\) of the PTZ camera may be written as:

\[
P_i = K_i[R_i][0]
\]  

(8.3)

The projection of a scene point \(X = [X, Y, Z, 1]^T\) onto an image point \(x = [x, y, 1]^T\) may now be expressed as \(x = K_i[R_i][0][X, Y, Z, 1]^T = K_iR_i[X, Y, Z]^T = K_iR_i d\), where \(d = [X, Y, Z]^T\) is the 3D ray emanating from the optical center to the image point \(x\).

Since the entries in the last column of the projection matrix of Equation 8.1 are zero, the depth of the world points along the ray is irrelevant and we only consider the projection of 3D rays \(d\). Therefore, in the case of a rotating camera, the mapping of 3D rays to image points is encoded by the 3×3 invertible projective transformation:

\[
P_i = K_iR_i
\]  

(8.4)

Given a 3D ray, \(d\) its projections onto two different images will be \(x_i = K_iR_i d\) and \(x_j = K_jR_j d\). Eliminating \(d\) from the equation it is easy to see that in the case of a rotating camera there exists a global 2D projective transformation (homography) \(H_{ij}\) that relates corresponding points in two views: \(x_i = H_{ij} x_j\), whose analytic expression is given by:

\[
H_{ij} = K_jR_jR_i^{-1}K_i^{-1} = K_jR_{ij}K_i^{-1}
\]  

(8.5)

### 8.4 PTZ Camera Networks with Master–Slave Configuration

PTZ cameras are particularly effective when in a master–slave configuration [1]: the master camera is set to have a global view of the scene so that it can track objects over extended areas using simple tracking methods with adaptive background subtraction. The slave camera can then follow a target’s trajectory to generate closeup imagery of the object. Obviously the slave and master roles can be exchanged, provided that both have PTZ capabilities. The master–slave configuration can then be extended to the case of multiple PTZ cameras. Figure 8.1 shows the pairwise relationship between two cameras in this configuration. \(H'\) is the homography relating the image plane of the master
Fig. 8.1. The pairwise relationship between two cameras in master–slave configuration. The camera $C'$ is tracking the target $A$ and camera $C$ is the slave. $H'$ is the homography from the stationary camera to a reference position of the PTZ camera in a wide angle view; $\Pi'$ is the image plane of the master camera; $\Pi$ is the reference plane of the slave camera. The homography $H_k$ relates the reference image plane $\Pi$ of image $I$ with the plane of current image $I_k$ of the slave camera.

camera $C'$ with the reference image plane $\Pi$ of the slave camera $C$ and $H_k$ is the homography relating the plane of the current image $I_k$ to the plane $\Pi$ of the reference image $I$. Once $H_k$ and $H'$ are known, the imaged location $a'$ of a moving target $A$ tracked by the stationary camera $C'$ can be transferred to the zoomed view of camera $C$ by

$$T_k = H_k \cdot H'$$

as $a = T_k \cdot a'$. With this pairwise relationship between cameras, the number of possible network configurations can be calculated. Given a set of PTZ cameras $C_i$ viewing a planar scene, we define $\mathcal{N} = \{C_i\}_{i=1}^n$ as a PTZ camera network with the master–slave relationship, where $n$ denotes the number of cameras in the network and $s$ defines the state of each camera. At any given time these cameras can be in one of two states $q_i = \{\text{master}, \text{slave}\}$.

The network $\mathcal{N}$ can be in one of $2^n - 2$ Possible state configurations. Not all cameras in a master state or all cameras in a slave state can be defined. It is worth noticing that from this definition more than one camera can act as a MASTER and/or SLAVE.

In principle, without any loss of generality, if all cameras in a network have an overlapping field of view (i.e., they are in a full connected topology) they can be set in a master–slave relationship with each other (not only in a one–to–one relationship). For example, in order to cover large areas, more master cameras can be placed with adjacent fields of view. In this case if it acts as a master camera, one slave camera can suffice to observe the whole area. Several master cameras can have overlapping fields of view so as to achieve higher tracking accuracy (multiple observations of the same object from different cameras.)
can be taken into account to obtain a more accurate measurement and to
determine a more accurate foveation by the slave camera). Similarly, more
than one camera can act as a slave, for example, for capturing high–resolution
images of moving objects from several viewpoints.

8.4.1 Minimal PTZ Camera Model Parameterization

In the case of real–time tracking applications, a trade–off can be made for
simplicity against strict geometric accuracy because of the different nature
of the problem. Whereas in 3D reconstruction, small errors in the internal
camera parameters can lead to nonmetric reconstruction, in tracking, weaker
Euclidean properties of the world are needed. Thus, more simple camera mod-
els can be assumed. It is also evident that the low number of features in the
use of high parametric models can lead to fit noise [38]. For these reasons we
adopt a minimal parameterization where only the focal length is allowed to
vary.

Because of the mechanical nature of PTZ cameras it is possible to assume that
there is no rotation around the optical axis. We also assume that the principal
point lies at the image center and that the CCD pixels are squared (i.e., their
aspect ratio equals 1). Under these assumptions we can write for $R_{ij}$:

$$R_{ij} = R_{ij}^\psi \cdot R_{ij}^\phi$$

(8.7)

with

$$R_{ij}^\psi = \begin{bmatrix}
\cos(\psi_{ij}) & 0 & -\sin(\psi_{ij}) \\
0 & 1 & 0 \\
\sin(\psi_{ij}) & 0 & \cos(\psi_{ij})
\end{bmatrix}$$

(8.8)

and

$$R_{ij}^\phi = \begin{bmatrix}
1 & 0 & 0 \\
0 & \cos(\phi_{ij}) & -\sin(\phi_{ij}) \\
0 & \sin(\phi_{ij}) & \cos(\phi_{ij})
\end{bmatrix}$$

(8.9)
where $\psi_{ij}$, $\phi_{ij}$ are the pan and tilt angles from image $i$ to image $j$, respectively. Under the same assumptions we can write

$$K_j = \begin{bmatrix} f_j & 0 & p_x \\ 0 & f_j & p_y \\ 0 & 0 & 1 \end{bmatrix}$$

(8.10)

and a similar expression for $K_i$ as

$$K_i = \begin{bmatrix} f_i & 0 & p_x \\ 0 & f_i & p_y \\ 0 & 0 & 1 \end{bmatrix}$$

(8.11)

where $(p_x, p_y)$ are the coordinates of the image center.

The homography $H_{ij}$ relating current view and reference view is thus only dependent on four parameters: $\psi_{ij}$, $\phi_{ij}$, $f_i$, and $f_j$. In our framework they are reduced to three because $f_i$ can be computed in closed form by using two images with different focal lengths obtained from the PTZ camera [39]. In the next section we illustrate our method to recursively track these parameters while the active camera is moving.

### 8.5 Cooperative Target Tracking

Real–time tracking applications are required to work properly even when camera motion is discontinuous and erratic due to command operations issued to the PTZ system. If surveillance applications are to be usable, then the system must recover quickly and in a stable manner following such movements. Unfortunately, methods that use only recursive filtering can fail catastrophically under such conditions, and although the use of more general stochastic filtering methods, such as particle filtering, can provide increased resilience, the approach is difficult to transfer to surveillance PTZ cameras operating with zoom.

In this section we show how to compute the time–variant homography $H_k$ using a novel combined tracking that addresses the difficulties described previously.

We adopt a SIFT–based matching approach to detect the relative location of the current image with respect to the reference image. At each time step we extract the SIFT features from the current image and match them with
those extracted from the reference frame, obtaining a set of points’ pairs. The SIFT features extracted in the reference image can be considered as visual landmarks. Once these visual landmarks are matched to the current view, the registration errors between these points are used to drive a particle filter whose state vector includes the parameters defining $H_k$. This allows stabilizing the recovered motion, characterizing the uncertainty and reducing the area where matches are searched. Moreover, because the keypoints are detected in scale space, the scene does not necessarily have to be well textured, which is often the case in an urban planar human made scene.

8.5.1 Tracking Using SIFT Visual Landmarks

Let us denote with $H_k$ the homography between the PTZ camera reference view and the frame grabbed at time step $k$. What we want to do is to track the parameters that define $H_k$ using a Bayesian recursive filter. Under the assumptions we made, the homography of Equation 8.6 is completely defined once the parameters $\psi_k$, $\phi_k$, and $f_k$ are known. We used this model to estimate $H_k$ relating the reference image plane $\Pi'$ with the current image at time $k$ (see Figure 8.1).

The focal length $f_i$ of the reference image can be computed in closed form using a further image $I_0$ matching with the reference image $I$ through the homography

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}. \tag{8.12}$$

This is achieved by Equation 8.5 with the assumptions made in Equation 8.10 and Equation 8.11:

$$HK_0K_0^T = K_iK_i^T$$

so as to obtain three equations to compute the focal length $f_0$ of $I_0$ and the focal length $f_i$ of the reference $I$:

$$f_0^2(h_{11}h_{21} + h_{12}h_{22}) = -h_{13}h_{23} \tag{8.13}$$

$$f_0^2(h_{11}h_{31} + h_{12}h_{32}) = -h_{13}h_{33} \tag{8.14}$$
\[ f_0^2(h_{21}h_{31} + h_{22}h_{32}) = -h_{23}h_{33} \] (8.15)

and two equations for \( f_i \):

\[ f_i^2 = \frac{f_0^2(h_{11}^2 + h_{12}^2)}{f_0^2(h_{31}^2 + h_{32}^2) + h_{33}^2} \] (8.16)

\[ f_i^2 = \frac{f_0^2(h_{21}^2 + h_{22}^2)}{f_0^2(h_{31}^2 + h_{32}^2) + h_{33}^2} \] (8.17)

In this computation, care must be taken to avoid pure rotation about the pan and tilt axis.

Thus we adopt the state vector \( x_k \), which defines the camera parameters at time step \( k \):

\[ x_k = (\psi_k, \phi_k, f_k) \] (8.18)

We use a particle filter to compute estimates of the camera parameters in the state vector. Given a certain observation \( z_k \) of the state vector at time step \( k \), particle filters build an approximated representation of the posterior probability density function (pdf) \( p(x_k|z_k) \) through a set of weighted samples \( \{ (x_i, w_i) \}_{i=1}^{N_p} \) (called particles), where the weights sum to 1. Each particle is thus a hypothesis on the state vector value (i.e. a homography), with an associated probability. The estimated value of the state vector is usually obtained through the weighted sum of all particles.

Like any other Bayesian recursive filter, the particle filter algorithm requires a probabilistic model for the state evolution between time steps, from which a prior pdf \( p(x_k|x_{k-1}) \) can be derived, and an observation model, from which a likelihood \( p(z_k|x_k) \) can be derived. Basically there is no prior knowledge of the control actions that drive the camera through the world, so we adopt a simple random walk model as a state evolution model. This is equivalent to assuming the actual value of the state vector to be constant in time and relying on a stochastic noise \( v_{k-1} \) to compensate for unmodeled variations: \( x_k = x_{k-1} + v_{k-1} \). \( v_{k-1} \sim \mathcal{N}(0, Q) \) is a zero mean Gaussian process noise, with the covariance matrix \( Q \) accounting for camera maneuvers (i.e., discontinuous and erratic motion).

The way we achieve observations \( z_k \) of the actual state vector value \( x_k \) is a little more complex and deserves more explanation. Let us denote with \( S_0 = \{ s_j^0 \}_{j=0}^N \) the set of SIFT points extracted from the reference view of
the PTZ camera (let us assume for the moment a single reference view), and denote with \( S_k = \{ s^j_k \}_{j=0}^{N^r} \) the set of SIFT points extracted from the frame grabbed at time step \( k \).

From \( S_0 \) and \( S_k \) we can extract pairs of SIFT points that match (through their SIFT descriptors) in the two views of the PTZ camera. After removing outliers from this initial set of matches through a RANSAC algorithm, what remains can be used as an observation for the particle filter. In fact, the set of remaining \( \tilde{N} \) pairs \( \mathcal{P}_k = \{(s^1_0, s^1_k), \ldots, (s^N_0, s^N_k)\} \) implicitly suggests a homography between the reference view and the frame at time step \( k \), one that maps the points \( \{s^1_0, \ldots, s^N_0\} \) into \( \{s^1_k, \ldots, s^N_k\} \). Thus, there exists a triple \((\hat{\psi}_k, \hat{\phi}_k, \hat{f}_k)\) that, in the above assumptions, uniquely describes this homography and that can be used as a measure \( z_k \) of the actual state vector value. To define the likelihood \( p(z_k|x^i_k) \) of the observation \( z_k \) given the hypothesis \( x^i_k \), we take into account the distance between the homography \( H^i_k \) corresponding to \( x^i_k \) and the one associated with the observation \( z_k \):

\[
p(z_k|x^i_k) \propto e^{-\frac{1}{\lambda} \sqrt{\sum_{j=1}^{\tilde{N}} (H^i_k s^j_0 - s^j_k)^2}}
\]

where \( H^i_k \cdot s^j_0 \) is the projection of \( s^j_0 \) in the image plane of frame \( k \) through the homography \( H^i_k \), and \( \lambda \) is a normalization constant.

It is worth noting that the SIFT points on frame \( k \) do not need to be computed on the whole frame. In fact, after the particle filter prediction step it is possible to reduce the area of the image plane where the SIFT points are computed close to the area where the particles are propagated. This reduces the computational load of the SIFT points computation and of the subsequent matching with the SIFT points of the reference image.

### 8.6 Extension to Wider Areas

We have seen that in order to maintain a consistent estimation of the transformation \( H_k \), the pan and tilt angles and the focal length must be estimated with respect to a reference plane. A problem arises when the slave camera is allowed to move outside the reference view.

As shown in Figure 8.2(a), matching with the reference view does not allow exploiting the whole parameter space of the pan and tilt angles or even zoom. The current zoomed view of the slave camera can only be matched when it has an overlap with the reference image. Figure 8.2(b) shows this limitation. The rectangles represent views that can be matched to acquire zoomed closeups of moving targets. Moreover, in the region where features are scarce, detection
is limited and/or inaccurate. To overcome these difficulties and to increment the applicability of the recursive tracking described in Section 8.5.1 to wider areas, a database of the scenes feature points is built during a learning stage.

SIFT keypoints extracted to compute the planar mosaic are merged into a large KD–Tree together with the estimated mosaic geometry (see Figure 8.3). The mosaic geometry in this case consists of one or more images taken so as to cover the whole field of regard\(^1\) of the PTZ system at several different pan and tilt angle and zoom settings. This can be computed by following the basic building blocks described in Equations 8.13–8.17. A more general and accurate solution can be obtained using a global refinement step with a bundle adjustment [40] [41]. What is obtained is shown in Figure 8.3.

The match for a SIFT feature extracted from the current frame is searched according to the Euclidean distance of the descriptor vectors. The search is performed so that bins are explored in the order of their distance from the query description vector, and stopped after a given number of data points have been considered [42]. Once the image \(I_l\) closest to the current view \(I_k\) is found, the homography \(G\) relating \(I_k\) to \(I_l\) is computed at run–time with RANSAC. The homography \(H_{lm}\) that relates \(I_l\) with the mosaic plane \(\Pi\) retrieved in the database is used to finally compute the likelihood. Equation 8.19 becomes:

\(^1\) The camera field of regard is defined as the union of all field of view over the entire range of pan and tilt rotation angles and zoom values.
Fig. 8.3. Each landmark in the database has a set of descriptors that corresponds to location features seen from different vantage points. Once the current view of the PTZ camera matches an image $I_l$ in the database, the inter-image homography $H_{lm}$ is used to transfer the current view into the reference plane $\Pi$. $I_m$ is the reference image used to compute the mosaic.

$$p(z_k|x_k^l) \propto e^{-\frac{1}{2} \sqrt{\sum_{i=1}^{N_i} (H_{i_k}^{-1} \cdot s_{i_k} - H_{lm} \cdot G \cdot s_{i_k})^2}}$$ (8.20)

As shown in Figure 8.3 the image points of the nearest neighbor image $I_l$ with respect to current view $I_l$ and the current view (i.e. the query to the database) are projected in $\Pi$ to compute the likelihood of Equation 8.20. In particular $I_m$ in the figure is the reference image used to compute the mosaic.

### 8.7 Vanishing Line for Zoomed Head Localization

The uncertainty characterization of recursive tracking is further used to localize the target head in the active camera by taking into account both the target position sensed by the master camera and the pdf parameters of the slave camera.

Assuming subjects closely vertical in the scene plane, the position of foot and head can be related by a planar homology [43] [44]. This transformation can be parameterized as

$$W = I + (\mu - 1) \frac{v_\infty^T \cdot I^T}{v_\infty^T \cdot I_\infty}$$, (8.21)

where $I$ is the 3×3 identity matrix, $v_\infty$ is the vanishing point of the directions orthogonal to the scene plane where targets are moving, $I_\infty$ is the correspond-
Fig. 8.4. The master vanishing lines is computed once in the master camera and transferred to the current view of slave camera through $T_k$.

ing vanishing line of the plane, and $\mu$ is the characteristic cross-ratio of the transformation.

According to this, at each time step $k$, the probability density function of the planar homology $H_k$ should be computed once the probability density function of, respectively, the vanishing point $v_{\infty,k}$ and the vanishing line $l'_{\infty,k}$ in the active camera views at time $k$ are known. In what follows we show how sampling from $p(x_k|z_k)$ allows estimating $p(v_{\infty,k}|z_k)$ and $p(l'_{\infty,k}|z_k)$ once the vanishing line $l_\infty$ in the master camera is known, as shown in Figure 8.4.

For each particle $i$ in the set of weighted samples $\{(x^i_k, w^i_k)\}_{i=1}^{N_p}$ modeling $H_k$, we calculate:

$$l'^i_{\infty,k} = T_k^{-T} \cdot l_\infty = [H^i_k \cdot H']^{-T} \cdot l_\infty$$  \hspace{1cm} (8.22)

$$v^i_{\infty,k} = \omega^i_k \cdot l'^i_{\infty,k}$$  \hspace{1cm} (8.23)

where $l_\infty$ in Equation 8.22 is the vanishing line in the master camera view (see Figure 8.4) and $\omega^i_k$ in Equation 8.23 is the dual image of the absolute conic \cite{45}:

$$\omega^i_k = K^i_k \cdot K_k^T$$  \hspace{1cm} (8.24)
The intrinsic camera parameters matrix

\[ K_i^k = \begin{bmatrix}
    f_i^k & 0 & p_x \\
    0 & f_i^k & p_y \\
    0 & 0 & 1
\end{bmatrix} \]

is computed with reference to the \( i \)-th particle. The estimated focal length \( f_i^k \) is extracted from the state component of Equation 8.18.

From the samples of Equations 8.22, 8.23 and 8.24 the respective pdfs are approximated as

\[
p(l^i_\infty,k | z_k) \approx \frac{1}{N} \sum_{i=1}^{N} \delta(l^i_\infty,k - l^i_\infty,k)
\]

\[
p(v_\infty,k | z_k) \approx \frac{1}{N} \sum_{i=1}^{N} \delta(v_\infty,k - v^i_\infty,k)
\]

\[
p(\omega_k | z_k) \approx \frac{1}{N} \sum_{i=1}^{N} \delta(\omega_k - \omega^k_i)
\]

In the same way, at time \( k \) the pdf \( p(W_k | z_k) = \frac{1}{N} \sum_{i=1}^{N} \delta(W_k - W_i^k) \) is computed as:

\[
W_i^k = I + (\mu - 1) \frac{v^i_\infty,k \cdot l^i_\infty,k^T}{v^i_\infty,k^T \cdot l^i_\infty,k}. \quad (8.25)
\]

using Equation 8.23 and Equation 8.22.

The cross–ratio \( \mu \), being a projective invariant, is the same in any image obtained with the active camera, while only the vanishing line \( l'_\infty,k \) and the vanishing point \( v_\infty,k \) vary when the camera moves, thus it must be observed only once. The cross–ratio \( \mu \) can be evaluated accurately by selecting the target foot location \( a \) and the target head location \( b \) in one of the frames of the zooming camera (see Figure 8.5) as

\[
\mu = \text{Cross}(v, a, b, v_\infty,k) \quad (8.26)
\]

where \( v \) is computed as the intersection of the mean vanishing lines \( \hat{l}'_\infty,k \), with the line passing from the mean vanishing point \( \hat{v}'_\infty,k \) to \( a \). Using the
Fig. 8.5. Foot and head positions in a frame taken from one frame of the slave camera to compute the cross-ratio $\mu$. The relationship between the involved entities used to compute $\mu$ is shown.

homogeneous vector representation and the cross–product operator $\times$ it can be estimated as $v = \hat{I}_{\infty,k} \times (\hat{v}_{\infty,k} \times a)$.

The pdf $p(M_k|z_k)$ of the final transformation $M_k$ which maps the target foot observed in the image of the master camera to the target head in the current image of the slave camera, is computed as

$$M_k^i = W_k^i \cdot H_k^i \cdot H^i$$

where $W_k^i$ represents an entire family of transformations. Given the estimated $p(x_k|z_k)$ of the slave camera and the imaged position of the target $p(a'_k|z'_k)$ as tracked from the master camera (see Figure 8.6) the distribution of possible head locations $p(b_k|z_k, z'_k)$ as viewed from the slave camera is estimated. We sample $L$ homographies from $p(x_k|z_k)$ and the same number of particles from the set of particles tracking the foot position in the master camera view $p(a'_k|z'_k)$ to obtain

$$b_k^j = M_k^i \cdot a_k^j \quad j = 1 \ldots L \quad (8.27)$$

Figure 8.6 shows the particle distribution of head localization uncertainty in the slave camera computed with Equation 8.27. It is worth noting that Equation 8.27 jointly taking into account both zooming camera sensor error and target–tracking error.
Fig. 8.6. Particles indicate uncertainty in head localization due to both active camera tracking and target tracking uncertainty.

8.8 Experimental Results

For initial testing purposes, a simple algorithm has been developed to automatically track a single target. The target is localized with the wide–angle stationary camera, and its motion is detected using a single Gaussian background subtraction algorithm within the image, and tracked using a particle filter. The experiments described here were run using two PTZ cameras in a master–slave configuration. The first acted as a master camera (i.e. it was not steered during the sequence), grabbing frames of the whole area at PAL resolution. The second acted as a slave camera, grabbing frames at a resolution of 320×240 pixels. Because of the limited extension of the monitored area, in our experiments SIFT feature points were extracted from only two images, since they sufficed to populate the features database, and to compute the focal length $f_i$ of the reference image (see Section 8.5.1).

In Figure 8.7 are shown some frames extracted from an execution of the proposed system. The particles show the uncertainty on the position of the target foot. Since the slave camera does not explicitly detect the target, the similarity between background and foreground appearance does not influence the estimated localization of the target. As shown in the last frame of the sequence, even if the target is outside the current FOV of the slave camera, its imaged position is still available in image coordinates.

A quantitative result for the estimated camera parameters is depicted in Figure 8.8, which shows how the state density evolves as tracking progresses. In each box are shown, clockwise from the upper left corner: the slave camera view, the pan angle pdf, the focal length pdf, and the tilt angle pdf. Time increases from left to right and top to bottom. During the sequence the slave
Fig. 8.7. Showing a master–slave camera tracking a human target. 12 frames (shown in two column) extracted from an execution of the proposed master–slave camera system (left is the master view, right the slave view). The particles show the uncertainty on the foot position of the target as the camera zoom in. Since the slave camera does not explicitly detect the target, background appearance similar to foreground appearance does not influence the estimated localization of the target.

The camera is panning, tilting, and zooming to follow the target. The focal length advances from 400 to about 1000 pixels, which corresponds approximately to a 3× zoom factor.

It can be seen that extending the focal length causes a significant increase
in the variance of parameter densities, which means that the estimated homography between the two cameras becomes more and more inaccurate. This is mainly caused by the fact that the number of features at high resolution that match those extracted from the reference image obviously decreases when zooming in, causing the SIFT match to be less effective. Figure 8.8 also shows how the focal length is much more sensitive to measurement when the camera zooms in. This is especially evident in the last rows of the figure, where the densities of the pan and tilt remain more peaked than the focal length density. This effect is absent in the first two rows, where the camera has a wide-angle view.

In Figure 8.9 are shown frames extracted from an execution of the proposed system on a different sequence. In particular Figure 8.9(a) shows the position of the target observed with the master camera. Figure 8.9(a) shows the correspondent mean head position (marked with a cross) in the slave camera computed with the time-variant homography estimated by the particle filter. The figure also shows the imaged line orthogonal to the scene plane as time progresses. As can be seen, increasing the zoom factor may bias the estimation of the head location.

We measured this error, calculating the Euclidean distance in pixels between the estimated \( \{\hat{x}_k, \hat{y}_k\} \) and the ground-truth head positions \( \{\bar{x}_k, \bar{y}_k\} \) respectively, with

\[
\epsilon_k = \sqrt{(\hat{x}_k - \bar{x}_k)^2 + (\hat{y}_k - \bar{y}_k)^2}
\]  

Figure 8.10(a) shows the estimated advancement in focal length. Figure 8.10(b) shows the relative mean error (dark gray curve) and the standard deviation (light gray curve) in head error localization. The mean error increases moderately, while a bias is present for high values of the focal length. The standard deviation increases almost linear, showing graceful degradation performance.

We also investigated the effect of the number of particles on the error calculated in Equation (8.28) and depicted in Figure 8.11. As can be seen, the error decreases while the number of particles increases, but once there are about 1000 particles or more, a further increase does not produce relevant improvements in filter performance. After this experiment, we found 1000 particles to represent a good trade-off between accuracy and computational load.

### 8.9 Conclusions

We proposed an uncalibrated framework for cooperative camera tracking that is able to estimate the time variant homography between a fixed wide-angle camera view and a PTZ camera view capable of acquiring high-resolution images of the target.
In particular, this method is capable of transferring a person’s foot position from the master camera view to the correspondent head position in the slave camera view.

By utilizing the discriminatory power of SIFT-like features in an efficient top-down manner we introduced new standards of robustness for real-time active zooming cameras cooperating in tracking targets at high resolution. Such robustness is critical to real-time applications such as wide-area video surveillance where human target identification at long distances is involved.

Since no direct camera calibration is needed, any camera in a reconfigurable sensor network can exploit the proposed method. Indeed, camera roles can be exchanged: if all cameras in a network have an overlapping field of view, they can be set in a master–slave relationship with each other (not only in a one to one relationship) without any loss of generality.

The proposed solution is capable of managing both transformation and target position uncertainty. Thus, camera management/control approaches using information gain (i.e., based on entropy) could be developed within the framework.

Acknowledgment

The authors would like to thank Giuseppe Lisanti for fruitful discussions that led to the extension of this work. This research was partially supported by the IST Program of the European Commission as part of the VIDI-Video project (Contract FP6-045547) and by the FREE SURF project funded by the Italian MIUR Ministry.

References


[34] Q. Zhang, Y. Chen, Y. Zhang, Y. Xu, SIFT implementation and optimization for multi-core systems, In the Proceedings of the Workshop on Advances in Parallel and Distributed Computational Models in conjunction with IPDPS 2008.


[38] B. Tordoff, Active control of zoom for computer vision, DPhil thesis, Oxford University.


Fig. 8.8. Frames showing probability distributions (histograms) of the slave camera parameters. Shown in each frame are (left to right and top to bottom) current view, pan, tilt, and focal length distributions. As expected, uncertainty increases with zoom factor.
Fig. 8.9. Screenshots from one experiment. (a) Fixed camera view. (b) PTZ camera view. The line in Figure (b) is the imaged line orthogonal to the scene plane passing through head and foot. The crosses in Figure (b) show estimated mean foot and head image locations.
Fig. 8.10. Error evaluation in head localization. (a) Estimated focal length advancement as the slave camera zooms in. (b) Related head localization mean error (dark gray curve) and its standard deviation (light gray curves). As can be seen, the uncertainty in the head localization increase with the focal length.
Fig. 8.11. Effects of the number of particles on head localization error. As expected, the higher the particles number, the better the filter’s performance.