Continuous Localization and Mapping of a Pan Tilt Zoom Camera for Wide Area Tracking

Giuseppe Lisanti · Iacopo Masi · Federico Pernici · Alberto Del Bimbo

Abstract Pan-tilt-zoom (PTZ) cameras are powerful to support object identification and recognition in far-field scenes. However, the effective use of PTZ cameras in real contexts is complicated by the fact that a continuous on-line camera calibration is needed and the absolute pan, tilt and zoom positional values provided by the camera actuators cannot be used because are not synchronized with the video stream. So, accurate calibration must be directly extracted from the visual content of the frames. Moreover, the large and abrupt scale changes, the scene background changes due to the camera operation and the need of camera motion compensation make target tracking with these cameras extremely challenging. In this paper, we present a solution that provides continuous on-line calibration of PTZ cameras which is robust to rapid camera motion, changes of the environment due to illumination or moving objects and scales beyond thousands of landmarks. The method directly derives the relationship between the position of a target in the 3D world plane and the corresponding scale and position in the 2D image, and allows real-time tracking of multiple targets with high and stable degree of accuracy even at far distances and any zooming level.

Keywords Rotating and Zooming Camera · PTZ Sensor · Localization and Mapping · Multiple Target Tracking

1 Introduction

Pan-tilt-zoom (PTZ) cameras are powerful to support object identification and recognition in far-field scenes. They are equipped with adjustable optical zoom lenses that can be manually or automatically controlled to permit both wide

Media Integration and Communication Center, University of Florence, Viale Morgagni 65, Florence, 50134, Italy
+39 055 275-1390

area coverage and close-up views at high resolution. This capability is particularly useful in surveillance applications to permit tracking of targets in high resolution and zooming in on biometric details of parts of the body in order to resolve ambiguities and understand target behaviors.

However, the practical use of PTZ cameras in real contexts of operation is complicate due to several reasons. First, the geometrical relationship between the camera view and the 3D observed scene is time-varying and depends on camera calibration. Unfortunately, the absolute pan tilt and zoom positional values provided by the camera actuators, even when they are sufficiently precise, in most cases are not synchronized with the video stream, and, for IP cameras, a constant frame rate cannot be assumed. So, accurate calibration must be extracted from the visual content of the frames. Second, the pan tilt and zooming facility may determine large and abrupt scale changes. This prevents the assumption of smooth camera motion. Moreover, since the scene background is continuously changing, some adaptive representation of the scene under observation becomes necessary. All these facts have significant impact also on the possibility of having effective target detection and tracking in real-time. Due to this complexity, there is a small body of literature on tracking with PTZ cameras and most of the solutions proposed were limited to either unrealistic or simple and restricted contexts of application.

In the following, we present a novel solution that provides continuous adaptive calibration of a PTZ camera and enables real-time tracking of targets in 3D world coordinates in general contexts of application. We demonstrate that the method is effective and is robust over long time periods of operation.

The solution has two distinct stages. In the off-line stage, we collect a finite number of keyframes taken from different viewpoints, and for each keyframe detect and store the scene landmarks and the camera pose. In the on-line stage, we

perform camera calibration by estimating the homographic transformation between the camera view and the 3D world plane at each time instant from the matching between the current view and the keyframes. Changes in the scene that have occurred over time due to illumination or objects are accounted with an adaptive representation of the scene under observation by updating the uncertainty in landmark localization. The relationship between target position in the 3D world plane and its position in the 2D image allows us to estimate the scale of target in each frame, compensate camera motion and perform accurate multi-target detection and tracking in 3D world coordinates.

2 Related work

In the following, we review the research papers that are most relevant for the scope of this work. In particular, we review separately solutions for self-calibration and target tracking with moving and PTZ cameras.

PTZ camera self-calibration

Hartley et al. [1] were the first to demonstrate the possibility of performing self-calibration of PTZ cameras based on image content. However, since calibration is performed off-line, their method cannot be applied in real-time contexts of operation. The method was improved in [2] with a global optimization of the parameters.

Solutions for on-line self-calibration and pose estimation of moving and PTZ cameras were presented by several authors. Among them, the most notable contributions were in [3,4,5,6,7,8,9]. Sinha and Pollefeys in [3] used the method of [2] to obtain off-line a full mosaic of the scene. Feature matching and bundle adjustment were used to estimate the values of the intrinsic parameters for different pan and tilt angles at the lowest zooming level, and the same process is repeated until the intrinsic parameters are estimated for the full range of views and zoomings. In [4] the same authors suggested that on-line control of a PTZ camera in closed loop could be obtained by matching the current frame with the full mosaic. However, their paper does not include any evidence of the claims nor provides any evaluation of the accuracy of the on-line calibration. Civera et al. [7], proposed a method that exploits real-time sequential mosaicing of a scene. They used Simultaneous Localization and Mapping (SLAM) with Extended Kalman Filter (EKF) to estimate the location and orientation of a PTZ camera and included the landmarks of the scene in the filter state. This solution cannot scale with the number of scene landmarks. Moreover, they only considered the case of camera rotations, and did not account for zooming. Lovegrove et al. [8] obtained the camera parameters between consecutive images by whole image alignment. As an alternative to using

EKF sequential filtering, they suggested to use keyframes to achieve scalable performance. They claimed to provide full PTZ camera self-calibration but did not demonstrate calibration with variable focal length. The main drawback of all these methods is that they assume that the scene is almost stationary and changes are only due to camera motion, which is a condition that is unlikely to happen in real contexts.

Wu and Radke [9] presented a method for on-line PTZ camera self-calibration based on a camera model that accounts for changes of focal length and lens distortion at different zooming levels. The authors claimed robustness to smooth scene background changes and drift-free operation, with higher calibration accuracy than [3,4] especially at high zoom levels. However, as reported by the authors, this method fails when a large component in the scene abruptly modifies its position or the background changes slowly. It is therefore mostly usable with stationary scenes. A similar strategy was also applied in [10], but accounts for pan and tilt camera movements, only.

Other authors developed very effective methods for pose estimation of moving cameras with pre-calibrated internal camera parameters [5,6]. In [5], Klein and Murray applied on-line bundle adjustment to the five nearest keyframes sampled every ten frames of the sequence. In [6], Williams et al. used a randomized lists classifier to find the correspondences between the features in the current view and the (pre-calculated) features from all the possible views of the scene, with RANSAC refinement. However both these approaches, if applied to a PTZ camera, are likely to produce over-fitting in the estimation of the camera parameters at progressive zoomings in.

Tracking with PTZ cameras

Solutions to perform general object tracking with PTZ cameras were proposed by a few authors. Hayman et al. [11] and Tordoff et al. [12] proposed solutions to adapt the PTZ camera focal length to compensate the changes of target size, assuming a single target in the scene and fixed scene background. In particular, in [11], the authors used the affine transform applied to lines and points of the scene background; in [12] the PTZ camera focal length is adjusted to compensate depth motion of the target. Kumar et al. [13] suggested to adapt the variance of the Kalman filter to the target shape changes. They performed camera motion compensation and implemented a layered representation of spatial and temporal constraints on shape, motion and appearance. However, the method is likely to fail in the presence of abrupt scale changes. In [14], Varcheie and Bilodeau addressed target tracking with IP PTZ cameras, in the presence of low and irregular frame rate. To follow the target, they commanded the PTZ motors with the predicted target position. A fuzzy clas-

sifier is used to sample the target likelihood in each frame. Since zooming is not managed, this approach can only be applied in narrow areas. The authors in [15] assumed that PTZ focal length is fixed and coarsely estimated from the camera CCD pixel size. They performed background subtraction by camera motion compensation to extract and track targets. This method is therefore unsuited for wide areas monitoring and highly dynamic scenes.

Solutions for tracking with PTZ cameras in specific domains of application were proposed in [16, 17, 18, 19]. All these methods exploit context-specific fiducial markers to obtain an absolute reference and compute the time-varying relationship between the positions of the targets in the 2D image and those in the 3D world plane. In [17], the authors used the a-priori known circular shape of the hockey rink and playfield lines to locate the reference points needed to estimate the world-to-image homography and compute camera motion compensation. The hockey players were tracked using a detector specialized for hockey players trained with Adaboost and particle filtering based on the detector's confidence [16]. The changes in scale of the targets was managed with simple heuristics using windows slightly larger/smaller than the current target size. Similar solutions were applied in soccer games [18, 19].

Beyond the fact that these solutions are domain-specific and have no general applicability, the main drawback is that fiducial markers are likely to be occluded and impair the quality of tracking.

2.1 Contributions and Distinguishing Features

The main contributions of the solution proposed are:

– We define a method for on-line PTZ camera calibration that jointly estimates the pose of the camera, the focal length and the scene landmark locations. Under reasonable assumptions, such estimation is Bayes-optimal, is very robust to zoom and camera motion and scales beyond thousands of scene landmarks. The method does not assume any temporal coherence between frames but only considers the information in the current frame.
– We provide an adaptive representation of the scene under observation that makes PTZ camera operations independent of the changes of the scene.
– From the optimally estimated camera pose we infer the expected scale of a target at any image location and compute the relationship between the target position in the 2D image and the 3D world plane at each time instant.

Differently from the other solutions published in the literature like [4], [7], [8] and [9] our approach allows performing on-line PTZ camera calibration also in dynamic scenes. Estimation of the relationship between positions in the 2D image and the 3D world plane permits more effective target detection, data association and real-time tracking.

Some of the ideas for calibration contained in this paper were presented with preliminary results under simplified assumptions in [20, 21]. Targets were detected manually in the first frame of the sequence and the scene was assumed almost static through time. Therefore we could not maintain camera calibration over hours of activity, neither support rapid camera motion.

3 PTZ Camera Calibration

In the following, we introduce the scene model and define the variables used. Then we discuss the off-line stage, where a scene map is obtained from the scene landmarks of the keyframes, and the on-line stage, where we perform camera pose estimation and updating of the scene map.

3.1 Scene model

We consider an operating scenario where a single PTZ camera is allowed rotating around its nodal point and zooming, while observing targets that move over a planar scene. The following entities are defined as time-varying random variables:

– The *camera pose* $\mathbf{c}$. Camera pose is defined in terms of the pan and tilt angles (ψ and ϕ, respectively), and focal length f of the camera. Since the principal point is a poorly conditioned parameter, it is assumed to be constant in order to obtain a more precise calibration [2]. Radial distortion was not considered since it can be assumed to be negligible for zooming operations [4].
– The *scene landmarks* $\mathbf{u}$. These landmarks account for salient points of the scene background. In the off-line stage SURF keypoints [22] are detected in keyframe images sampled at fixed intervals of pan, tilt and focal length. A SURF descriptor is associated to each landmark. These landmarks change during the on-line camera operation.
– The *view map* $\mathbf{m}$ and *scene map* $\mathbf{M}$. A view map is created for each keyframe that collects the scene landmarks (i.e. $\mathbf{m} = \{\mathbf{u}_i\}$). The scene map is obtained as the union of all the view maps and collects all the scene landmarks that have been detected at different pan, tilt and focal lengths values (i.e. $\mathbf{M} = \{\mathbf{m}_k\}$). Since the scene landmarks change through time, these maps will change accordingly.
– The *landmark observations* $\mathbf{v}$. These landmarks account for the salient points that are detected in the current frame. They can either belong to the scene background or to targets. The SURF descriptors of the landmark observations $\mathbf{v}$ are matched with the descriptors of the scene
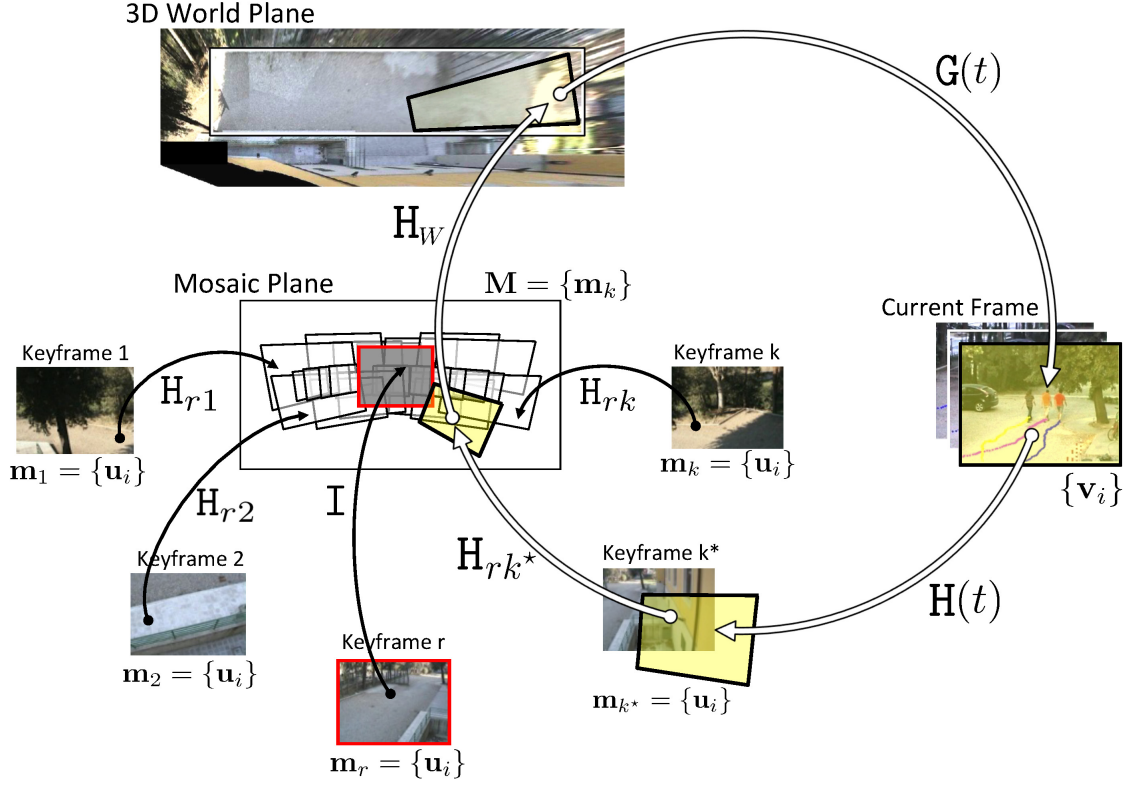
Fig. 1: Main entities and their relationships: the current frame and the landmark observations extracted $\mathbf{v}$; the view maps $\mathbf{m}$ including the scene landmarks $\mathbf{u}$; the initial scene map $\mathbf{M}$ obtained from the union of the view maps; the 3D scene; the functions that represent the relationships between these entities.

landmarks $\mathbf{u}$, in order to estimate the camera pose and update the scene map.

- The *target state* $\mathbf{s}$. The target state is represented in 3D world coordinates and includes both the position and speed of a target. It is assumed that targets move on a planar surface, i.e. $Z = 0$, so that $\mathbf{s} = [X, Y, \dot{X}, \dot{Y}]$.
- The *target observations* in the current frame, $\mathbf{p}$. This is a location in the current frame that is likely to correspond to the location of a target. At each time instant t there is a non-linear and time varying function $\mathbf{g}$ relating the position of the target in world coordinates $\mathbf{s}$ to the location $\mathbf{p}$ of the target in the image. Its estimation depends on the camera pose $\mathbf{c}$ and the scene map $\mathbf{M}$ at time t.

Fig. 1 provides an overview of the main entities of the scene model and their relationships.

3.2 Off-line Scene Map Initialization

In the off-line stage, image views (keyframes) are taken at regular samples of pan and tilt angles and focal length, and view maps $\mathbf{m}_k$ are created so to cover the entire scene. SURF keypoints [22] are organized in a k-d tree for each view map.

Given a reference keyframe and the corresponding view map $\mathbf{m}_r$, the homography that maps each $\mathbf{m}_k$ to $\mathbf{m}_r$ can be

estimated as in the usual way of planar mosaicing [1]:

$$\mathtt{H}_{rk} = \mathtt{K}_r \mathtt{R}_r \mathtt{R}_k^{-1} \mathtt{K}_k^{-1} \tag{1}$$

The optimal values of both the external camera parameter matrix $\mathtt{R}_k$ and the internal camera parameter matrix $\mathtt{K}_k$ are estimated by bundle adjustment for each keyframe k.

Differently from [5], we use bundle adjustment for off-line scene map initialization and use the whole set of keyframes of the scene at multiple zoomings. Since keyframes were taken by uniform sampling of the parameter space, over-fitting of camera parameters is avoided. This results in a more accurate on-line estimation of the PTZ parameters. The difference in the accuracy of the estimation is especially sensible in the case in which PTZ operates at high zooming. Fig. 2 shows an example of estimation of the focal length with the two approaches for a sample sequence with right panning and progressive zooming-in.

The pan, tilt, zoom values of the camera actuators are stored in order to uniquely identify each view map. The complete scene map $\mathbf{M}$ is obtained as the union of all the view maps. Differently from [20], a forest of k-d trees is used for matching.

Fig. 2: Estimations of the camera focal length of the last frame of a sequence with right panning and progressive zooming in: a) using the on-line bundle adjustment of [5]; b) using our off-line solution with keyframes obtained by uniform sampling of the camera parameter space and the last frame. The focal length of the last frame is represented with a square box on the scene mosaic. Focal length estimation is respectively 741.174 pixels and 2097.5 pixels. The true focal length is 2085 pixels.

3.3 On-line camera pose estimation and mapping

The positional values provided by the camera actuators at each time instant, although not directly usable for on-line camera calibration, are nevertheless sufficiently precise to retrieve the view map $\mathbf{m}_{k^\star}$ with the closest values of pan, tilt and focal length. This map is likely to have almost the same content as the current frame and many landmarks will match. The landmarks matched can be used to estimate the homography $\mathtt{H}(t)$ from the current view to $\mathbf{m}_{k^\star}(t)$. Matching is performed according to Nearest Neighbor distance ratio as in [23] and RANSAC. To reduce the computational effort of matching, only a subset of the landmarks in $\mathbf{m}_{k^\star}$ is taken by random sampling. The descriptors of the landmarks matched are updated using a running average with a forgetting factor.

The optimal estimation of $\mathtt{H}(t)$ on the basis of the correspondences between landmark observations $\mathbf{v}_i(t)$ and scene landmarks $\mathbf{u}_i(t)$ is fundamental for effective camera pose (pan, tilt, focal length) estimation and mapping in real conditions. However, changes of the visual environment due to illumination or to objects entering, leaving or changing position in the scene induce modifications of the original scene map as time progresses. Moreover, imprecisions in the detection and estimation process might affect scene landmark estimation and localization. To this end, under reasonable assumptions, we derive a linear measurement model that accounts for all the sources of error of landmark observations, that permits to obtain the optimal localization of the scene landmarks. Permanent modifications of the scene are accounted through a landmark birth-death process that includes new landmarks and discards temporary changes.

Closed-form recursive estimation of scene landmarks

Camera pose estimation and mapping requires inference of the joint probability of the camera pose $\mathbf{c}(t)$ and scene landmark locations in the map $\mathbf{M}(t)$, given the landmark observations $\mathbf{v}$ until time t and the initial scene map $\mathbf{M}(0)$:

$$p\big(\mathbf{c}(t), \mathbf{M}(t) | \mathbf{v}(0:t), \mathbf{M}(0)\big). \qquad (2)$$

In order to make the problem scalable with respect to the number of landmarks, Eq. (2) is approximated by decoupling camera pose estimation from map updating:

$$\underbrace{p\big(\mathbf{c}(t)|\mathbf{v}(t), \mathbf{M}(t-1)\big)}_{\text{camera pose estimation}} \underbrace{p\big(\mathbf{M}(t)|\mathbf{v}(t), \mathbf{c}(t), \mathbf{M}(t-1)\big)}_{\text{map updating}} \quad (3)$$

Considering the the view map $\mathbf{m}_{k^\star}$ with the closest values of pan, tilt and focal length and applying Bayes theorem to the map updating term, Eq. (3) can be rewritten as:

$$p\big(\mathbf{m}_{k^\star}(t)|\mathbf{v}(t), \mathbf{c}(t), \mathbf{m}_{k^\star}(t-1)\big) =$$
$$p\big(\mathbf{v}(t)|\mathbf{c}(t), \mathbf{m}_{k^\star}(t)\big) p\big(\mathbf{m}_{k^\star}(t)|\mathbf{m}_{k^\star}(t-1)\big), \quad (4)$$

where the term $p\big(\mathbf{m}_{k^\star}(t)|\mathbf{m}_{k^\star}(t-1)\big)$ indicates that view map $\mathbf{m}_{k^\star}(t)$ at time t depend only on $\mathbf{m}_{k^\star}(t-1)$. Assuming that for each camera pose the observation landmarks $\mathbf{v}_i$ that match the scene landmarks $\mathbf{u}_i$ in $\mathbf{m}_{k^\star}(t)$ are independent of each other, i.e.:

$$p\big(\mathbf{v}(t)|\mathbf{c}(t), \mathbf{m}_{k^\star}(t)\big) = \prod_i p\big(\mathbf{v}_i(t)|\mathbf{c}(t), \mathbf{u}_i(t)\big), \quad (5)$$

Eq. (4) modifies in:

$$p\big(\mathbf{m}_{k^\star}(t)|\mathbf{v}(t), \mathbf{c}(t), \mathbf{m}_{k^\star}(t-1)\big) =$$
$$\prod_i p\big(\mathbf{v}_i(t)|\mathbf{c}(t), \mathbf{u}_i(t)\big) p\big(\mathbf{u}_i(t)|\mathbf{u}_i(t-1)\big), \quad (6)$$

where $p\big(\mathbf{u}_i(t)|\mathbf{u}_i(t-1)\big)$ is the prior pdf of the i-th scene landmark at time t given its state at time $t-1$. Under the assumptions that both scene landmarks $\mathbf{u}_i(t)$ and the keypoint localization error have a Gaussian distribution, and that Direct Linear Transform is used, the observation model $p\big(\mathbf{v}_i(t)|\mathbf{c}(t), \mathbf{u}_i(t)\big)$ can be expressed as:

$$\mathbf{v}_i(t) = \mathbf{H}_i(t)\mathbf{u}_i(t) + \boldsymbol{\lambda}_i(t), \qquad (7)$$

where $\mathbf{H}_i(t)$ is the 2×2 matrix obtained by linearizing the homography $\mathtt{H}(t)$ at $\mathbf{v}_i(t)$ and $\boldsymbol{\lambda}_i(t)$ is an additive Gaussian noise term with covariance $\boldsymbol{\Lambda}_i(t)$ that represents the whole error in the landmark mapping process. This covariance can be expressed in closed form and in homogeneous coordinates as:

$$\Lambda_i(t) = \mathtt{B}_i(t)\,\Sigma_i(t)\mathtt{B}_i(t)^\top + \Lambda_i' + \mathtt{H}(t)^{-1}\mathtt{P}_i(t)\mathtt{H}(t)^{-\top}, \quad (8)$$

where the three terms account respectively for the spatial distribution of the matched landmarks, the covariance of keypoint localization in the current frame and the uncertainty associated to the scene landmark positions in the view map. In Eq. (8), $\Sigma_i(t)$ is the 9×9 homography covariance matrix (calculated in closed form according to [24]) and $\mathtt{B}_i(t)$ is the 3×9 block matrix of landmark observations; Λ_i' models the keypoint detection error covariance; $\mathtt{P}_i(t)$ is the covariance of the estimated landmark position on the nearest view map, and $\mathtt{H}$ is obtained from the Direct Linear Transform. Covariance $\mathbf{\Lambda}_i(t)$ can be directly obtained as the 2×2 principal minor of $\Lambda_i(t)$.

The optimal localization of the scene landmarks is therefore obtained in closed form through multiple applications of the Extended Kalman Filter to each landmark observation, with the Kalman gain being computed as:

$$\mathbf{K}_i(t) = \mathbf{P}_i(t|t-1)\mathbf{H}_i(t)^{-1}\left[\mathbf{H}_i(t)^{-1}\mathbf{P}_i(t|t-1)\mathbf{H}_i(t)^{-\top} + \mathbf{\Lambda}_i(t)\right]^{-1}, \tag{9}$$

where $\mathbf{P}_i$ is the Kalman covariance of the i-th scene landmark.

Birth-death of scene landmarks

Objects that enter or leave the scene introduce modifications of the original scene map. Their landmarks are not taken into account in the computation of $\mathtt{H}(t)$ at the current time (they are the RANSAC outliers in the matching process), but are taken into account in the long term, in order to avoid that the representation of the original scene becomes drastically different from that of the current scene. We assume that new landmarks that persist in 20 consecutive frames and are closest to the already matched landmarks have higher probability of belonging to a new scene element (they have smaller covariance according to Eq. (8)). According to this, we implemented a *proximity check* (Fig. 3) that computes such probability as the ratio between the bounding box of the landmarks matched and the extended bounding box of the new landmark (respectively box A and B in Fig. 3). Such candidate landmarks are included in $\mathbf{m}_{k^\star}$ using the homography $\mathtt{H}(t)$. Landmarks are terminated when they are no more matched in consecutive frames.

Since the transformation between two near frames under pan tilt and zoom can be locally approximated by a similarity transformation, the asymptotic stability of the updating procedure is guaranteed by the Multiplicative Ergodic Theorem [25]. Therefore, we can assume that no sensible drifting is introduced in the scene landmark updating.
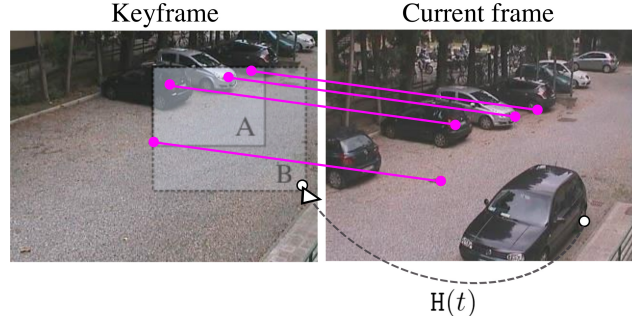

Keyframe   Current frame

$\mathtt{H}(t)$

Fig. 3: *Proximity check* for scene map updating. Current frame and its nearest keyframe in the scene map. Matched landmarks and a new landmark are shown in magenta and white, respectively, together with their bounding boxes.

Localization in world coordinates

Looking at Fig. 1, the time varying homography $\mathtt{G}(t)$ (in homogeneous coordinates), mapping a target position in the world plane to its position $\mathbf{p}$ in the current frame, can be represented as:

$$\mathtt{G}(t) = \left(\mathtt{H}_W \mathtt{H}_{rk^\star} \mathtt{H}(t)\right)^{-1}, \tag{10}$$

where $\mathtt{H}_W$ is the stationary homography from the mosaic plane to the 3D world plane:

$$\mathtt{H}_W = \mathtt{H}_s \mathtt{H}_p, \tag{11}$$

that can be obtained as the product of the rectifying homography $\mathtt{H}_p$ (derived from the projections of the vanishing points by exploiting the single view geometry of the planar mosaic[1] [26]) and transformation $\mathtt{H}_s$ from pixels in the mosaic plane to 3D world coordinates (estimated from the projection of two points at a known distance L in the world plane onto two points in the mosaic plane as in Fig. 4).

4 Target tracking with PTZ cameras

We perform multi-target tracking in 3D world coordinates using the Extended Kalman Filter. Data association to discriminate between target trajectories is implemented according to the Cheap-JPDAF model [27].

The relationship between the image plane and the 3D world plane of Eq. (10) allows us to obtain the target scale and perform tracking in the 3D world plane. As it will be shown in Section 5, tracking in the 3D world plane allows a better discrimination between targets.

---

[1] In the case of a PTZ sensor, the homography between each keyframe and the reference keyframe is the infinite homography $\mathtt{H}_\infty$ that puts in relation vanishing lines and vanishing points between the images.
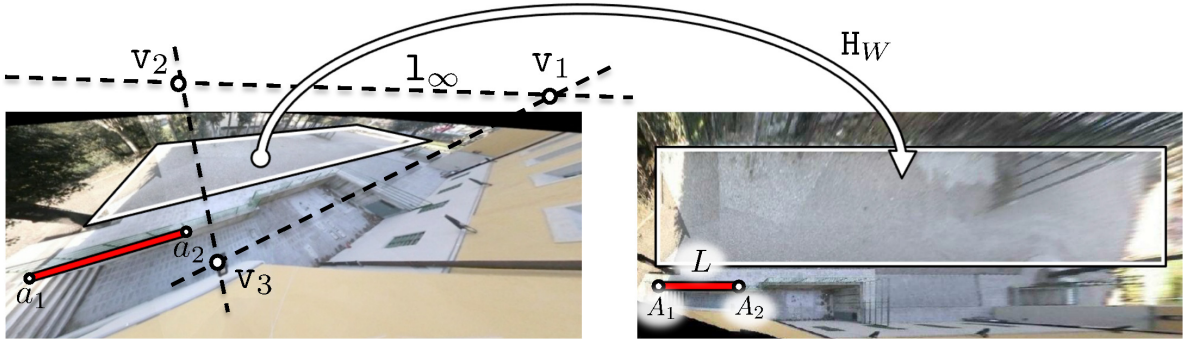
Fig. 4: The transformation from the 2D mosaic plane (*Left*) to the 3D world plane (*Right*). The vanishing points and the vanishing lines are used for the computation of matrix $\mathtt{H}_p$. A pair of corresponding points to compute $\mathtt{H}_s$ is shown.

4.1 Target scale estimation

At each time instant t, the homography $\mathtt{G}(t)$ permits to derive the homology relationship that directly provides the scale at which the target is observed in the current frame:

$$\mathbf{h}(t) = \mathtt{W}(t)\mathbf{p}(t) \tag{12}$$

where $\mathbf{h}(t)$ and $\mathbf{p}(t)$ are respectively the position of the target top and bottom in the image plane and $\mathtt{W}(t)$ is defined as:

$$\mathtt{W}(t) = \mathtt{I} + (\mu - 1)\frac{\mathbf{v}_\infty(t) \cdot \mathbf{l}_\infty^\top(t)}{\mathbf{v}_\infty^\top(t) \cdot \mathbf{l}_\infty(t)}, \tag{13}$$

where $\mathtt{I}$ is the identity matrix, $\mathbf{l}_\infty(t)$ is the world plane vanishing line, $\mathbf{v}_\infty(t)$ is the vanishing point of the world normal plane direction, and μ is the cross-ratio. The vanishing point $\mathbf{v}_\infty(t)$ is computed as $\mathbf{v}_\infty(t) = \mathtt{K}(t)\mathtt{K}(t)^\top \cdot \mathbf{l}_\infty(t)$, with $\mathbf{l}_\infty(t) = \mathtt{G}(t) \cdot [0,\,0,\,1]^\top$ and $\mathtt{K}(t)$ is derived from $\mathtt{H}(t)$ as in [20]. Estimation of the target scale allows us to apply the detector at a single scale instead of multiple scales and improve in both recall and computational performance for detection and tracking.

4.2 Multiple Target Tracking

The Extended Kalman filter observation model for each target is defined as:

$$\mathbf{p}(t) = \mathbf{g}\big(s(t), t\big) = \big[\, \mathbf{G}(t)\ \mathbf{0}_{2\times2} \,\big]\mathbf{s}(t) + \zeta(t), \tag{14}$$

where $\zeta(t)$ is a Gaussian noise term with zero mean and diagonal covariance that models the target localization error in the current frame; $\mathbf{s}(t)$ is the target state, represented in 3D world coordinates, $\mathbf{G}(t)$ is the homography $\mathtt{G}(t)$ linearized at the predicted target position and $\mathbf{0}_{2\times2}$ is the 2×2 zero matrix. Assuming constant velocity, the motion model in the 3D world plane is defined as:

$$p(\mathbf{s}(t)|\mathbf{s}(t-1)) = \mathcal{N}(\mathbf{s}(t); \mathbf{A}\mathbf{s}(t-1), \mathbf{Q}), \tag{15}$$

where $\mathbf{A}$ is the 4×4 constant velocity transition matrix and $\mathbf{Q}$ is the 4×4 process noise matrix. For multiple target tracking, $\mathbf{G}(t)$ influences the target covariance of the Cheap-JPDAF respectively for the Kalman gain expression:

$$\mathbf{W}(t) = \mathbf{P}(t|t-1)\mathbf{G}(t)\mathbf{S}(t|t)^{-1}, \tag{16}$$

and the target covariance on the image plane:

$$\mathbf{S}(t|t) = \mathbf{G}(t)\mathbf{P}(t|t-1)\mathbf{G}(t)^\top + \mathbf{V}(t), \tag{17}$$

where $\mathbf{V}(t)$ is the covariance matrix of the measurement error of Eq. (14).

5 Experimental results

In this Section we report on an extensive set of experiments to assess the accuracy of our PTZ camera calibration method and its effective exploitation for real-time multiple target tracking.

5.1 PTZ camera calibration

In the following, we summarize the experiments that validate our approach for camera calibration. We justify the use of motor actuators to retrieve the closest scene map; we report on the precision of the off-line scene map initialization and the on-line camera pose estimation and mapping.

Accuracy of PTZ motor actuators

We validated the use of pan tilt and zoom values provided by the camera motor actuators to retrieve the closest view map, by checking their precision with the same experiment as in [9]. We placed four checkerboard targets at different positions in a room. These positions corresponded to different pan, tilt and zoom conditions. A SONY SNC-RZ30P PTZ camera was moved to a random position every 30 seconds and returned at the initial positions every hour. For each

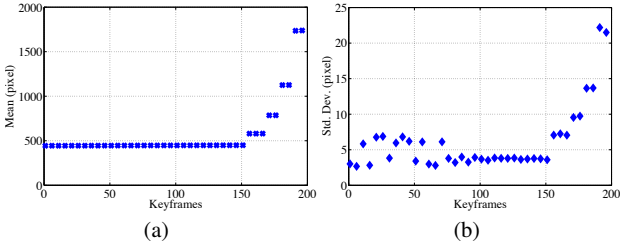(a)   (b)

Fig. 6: Average (a) and standard deviation (b) of the bundle-adjusted focal length for the keyframes used in scene map initialization. Keyframes are ordered for increasing values of focal lenght.

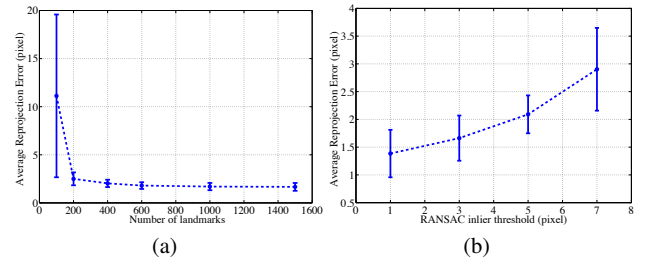(a)   (b)

Fig. 7: Reprojection error as a function of (a) the number of landmarks extracted (b) inlier threshold in the RANSAC algorithm, for Sequence 1 under test.

image view the corners of the checkerboard were extracted and compared to the reference image. The errors were collected for 200 hours. We have measured an average error of 2 pixels at the lowest zooming and 9 pixels for the maximum zooming. Fig. 5 shows the plots of the errors and the initial and final camera view for each target.

Scene map initialization

Off-line scene map initialization as discussed in Sect. 3.2 is accurate and produces repeatable results. Fig. 6 reports the mean and standard deviation of the focal length estimated during the scene map initialization. In this experiment, we acquired images of the same outdoor scene in 43 consecutive days at different time of the day, at 202 distinct values of pan tilt zoom. The PTZ camera was driven using motor actuators. We can notice that the standard deviation of the focal length that is estimated through off-line bundle adjustment increases almost proportionally with focal length. The maximum standard deviation value observed is 23 pixels at focal length of about 1700 pixels.

On-line PTZ camera pose estimation and mapping

In this experiment, we report on the average reprojection error and calibration errors with our method. We discuss the influence of the number of landmarks and RANSAC inlier threshold on the reprojection error and the effectiveness of scene landmark updating.

As in [9], we recorded 10 outdoor video sequences of 8 hours each (80 hours in total). Due to the long period of observation, all the sequences include slow background changes due to shadows or illumination variations, as well as large changes due to moving objects entering or exiting the scene. The PTZ camera was moved continuously using the motor actuators and stopped for a few seconds at the same pan tilt zoom values, so to have a large number of keyframes at the same scene locations and different conditions, in all the sequences. On average we performed about 34000 measurements per sequence. For each keyframe, a grid of points

was superimposed and the average reprojection error was measured between the grid points as obtained by the estimated homography and the same points by the off-line bundle adjustment.

Tab. 1 shows the average reprojection error, the errors in the estimation of pan, tilt and focal length and the improvements that are obtained with the *proximity checking*, for the outdoor sequences under test. As in [9], the errors in pan and tilt angles were computed as $e_\psi(t) = |\psi(t) - \psi_{rk}|$ and $e_\phi(t) = |\phi(t) - \phi_{rk}|$, respectively, and the focal length error as $e_f(t) = \left| \frac{f(t) - f_{rk}}{f_{rk}} \right|$ (in percentage). Pan and tilt angles estimated and those calculated with bundle adjustment were obtained from the rotation matrices $R(t) = K_r^{-1} H_{rk\star} H(t) K_k$ (see Eq. (10)) and $R_{rk} = K_r^{-1} H_{rk} K_k$ (see Eq. (1)), respectively. The results confirm that *proximity checking* avoids to select landmarks that introduce drifting in the homography estimation. It can be observed that errors in focal length measured with our method over a long period in an outdoor scenario are similar to those obtained in [9], and lower than those in [4] (as reported in [9]), for an indoor experiment with a few keyframes.

The reprojection error depends on both the number of landmarks extracted and the RANSAC threshold for inliers as shown in Fig. 7 for one of the sequences under test (Sequence 1). It can be observed that a large reprojection error with high standard deviation (plotted at one sigma) is present below 200 landmarks. Instead, such error is low when the number of landmarks is between 200 and 1500 (Fig. 7(a)). Fig. 7(b) shows that a RANSAC thresholds between 1 and 3 pixels for the inliers used in the homography estimation assures small reprojection errors. Values of 1000 and 3 pixels were used respectively for the number of landmarks extracted and RANSAC threshold in our experiments.

Scene map updating significantly contributes to the robustness of our camera calibration to both slow and sudden variations of the scene, maintaining a high number of RANSAC inliers through time. Fig. 8(a) shows the cumulative sum of the inliers with and without scene landmark updating. It is possible to observe that without scene land-
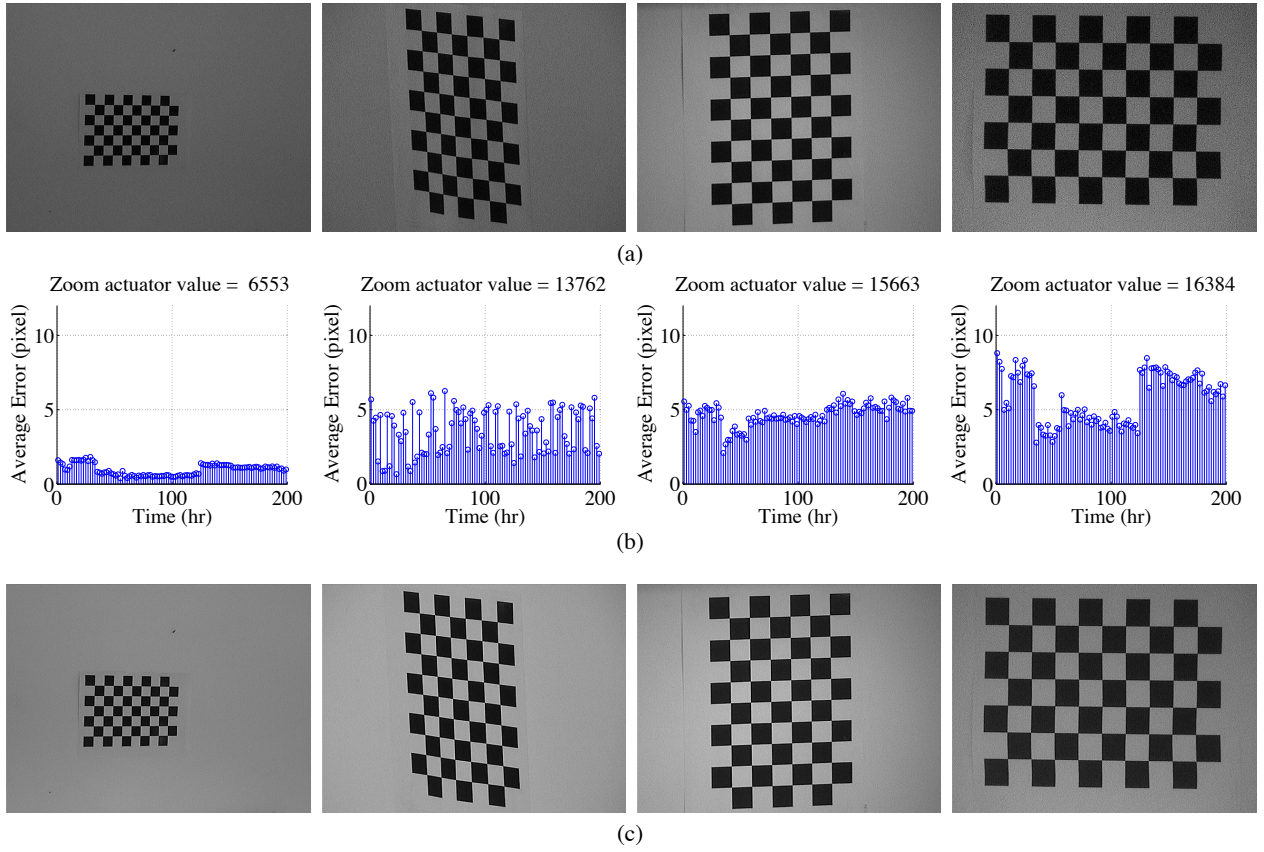
Fig. 5: (a) Checkerboard images at the initial camera pose. (b) Average Errors over 200 hours. (c) Checkerboard images after the camera has returned in the same initial pose after 200 hours.

Sequence	#measurements	Avg. reproj. error		Pan		Tilt		Focal Length	
–	–	Ours	Ours w/o p.	Ours	Ours w/o p.	Ours	Ours w/o p.	Ours	Ours w/o p.
Seq. 1	34,209	**2.83**	2.96	**1.18**	1.55	**0.39**	0.42	**0.96**	1.06
Seq. 2	34,605	**6.69**	6.90	2.47	**2.09**	**0.68**	0.94	4.41	**3.65**
Seq. 3	33,102	**3.26**	3.30	1.26	**1.17**	0.33	0.33	**0.84**	0.91
Seq. 4	33,939	**6.88**	7.09	**2.11**	2.58	1.93	**1.73**	**2.78**	3.79
Seq. 5	33,974	**22.54**	60.04	**11.14**	11.53	**9.51**	9.85	**12.49**	14.21
Seq. 6	33,570	**3.21**	4.26	**1.91**	2.84	**0.49**	0.54	**1.26**	3.05
Seq. 7	34,157	3.62	**3.59**	1.71	**1.27**	**0.35**	0.43	**1.81**	2.15
Seq. 8	33,932	**21.76**	21.99	**7.08**	7.41	10.07	**9.23**	11.91	**11.81**
Seq. 9	34,558	**8.78**	12.26	**3.35**	5.48	**1.37**	2.70	**3.47**	4.80
Seq. 10	34,405	**8.47**	9.26	7.20	**5.71**	**5.28**	6.59	**8.99**	9.54
Average	34,032	**8.80**	13.17	**3.94**	4.16	**3.04**	3.28	**4.89**	5.50

Table 1: Average reprojection error and calibration errors of pan, tilt and focal length with and without *proximity check* evaluated at the keyframes during the period of observation.

mark updating the number of inliers decreases (the cumulative curve is almost flat) as the initial landmarks do not match anymore with the landmarks observed due to scene changes. Fig. 8(b) shows the distribution of the inliers in the two cases. With no scene landmark updating, typically only few of the original landmarks are taken as inliers for each keyframe, that is insufficient to assure a robust calibration over time. With scene landmark updating, a higher number of inliers is taken for each frame that include both the original and the new scene landmarks. As can be inferred from

Fig. 8, in a dynamic scene few of the original scene landmarks survive at the end of the observation period. Fig. 9 highlights the scene landmark lifetime over a 20 minutes window, for one keyframe (randomly chosen). The scene landmarks with ID $\in [0..2000]$ are the original landmarks. Landmarks with ID ≥ 2000 are those observed during the 20 minutes.

Our PTZ camera calibration keeps sufficiently stable over long periods of observation. Fig. 10 shows a typical plot of the reprojection error over 8-hour operation for a sam-
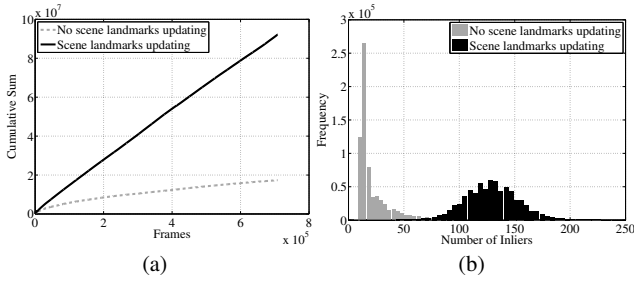
Fig. 8: (a) Cumulative Sum of number of inliers as a function of time: without and with scene landmark updating (dashed and solid curve respectively). (b) Distributions of the number of inliers without and with scene landmark updating (grey and black bins respectively).
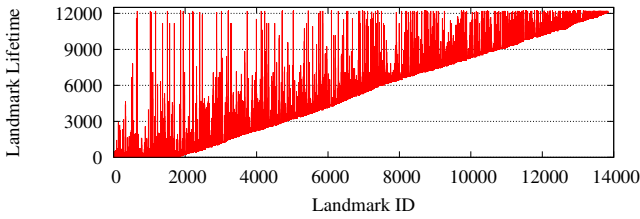

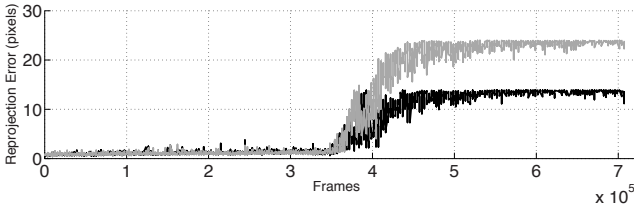Fig. 9: Lifetime of scene landmarks observed for a sample keyframe.

Fig. 10: Reprojection error over 8-hour operation for a sample keyframe without (light plot) and with (dark plot) *proximity checking*.

ple keyframe. Camera calibration at different time of the day without and with scene landmark updating is shown in Fig. 11(a-b) for a few sample frames. It can be observed that with scene landmark updating, camera calibration (represented by the superimposed grid of points) is still accurate despite of the large illumination changes occurred in the scene.

5.2 Multi-Target Tracking with PTZ cameras

In the following, we summarize experiments on multi-target tracking in 3D world coordinates using our on-line PTZ camera calibration, and compare our method with a few methods that appeared in the literature on a standard PTZ video sequence. In our experiments targets were detected automatically using the detector in [28].

Influence of camera calibration

To evaluate the impact of our PTZ calibration on tracking, we recorded a 8-hour sequence in a parking area during a working day and extracted three videos with one, two and three targets. This is a dynamic condition, with both smooth and abrupt scene changes. Multi-target tracking performance was evaluated according to both the CLEAR MOT [29] and USC metrics [30]. The CLEAR MOT metrics measures tracking accuracy (MOTA):

$$\text{MOTA} = 1 - \frac{\sum_t (\text{FN}_t + \text{FP}_t + \text{ID\_SW}_t)}{\sum_t \text{n}_t} \qquad (18)$$

and precision (MOTP):

$$\text{MOTP} = \frac{\sum_{i,t} \text{VOC}_{i,t}}{\sum_t \text{TP}_t}, \qquad (19)$$

where FN_t and FP_t are respectively the false negatives and positives, ID_SW_t are the identity switches, n_t is the number of targets and $\text{VOC}_{i,t}$ is the VOC score of the i-th target at time t. The USC metric reports the ratio of the trajectories that were successfully tracked for more than 80% (MT), the ratio of mostly lost trajectories that were successfully tracked for less than 20% (ML), the rest partially tracked (PT) and the average count of false alarms per frame (FAF). We measured the performance for the method with no scene map updating, with no *proximity checking* and for the full method.

From Tab. 2 it is apparent that scene map updating has a major influence on the number of false negatives and false positives and therefore on the tracking accuracy. *Proximity checking* has also a positive impact on the reduction of false positives and determines an average increase of the accuracy of about 10%.

Influence of tracking in 3D world coordinates

To analyze the effect of using 3D world coordinates we run our method in 2D image coordinates (not applying mapping in the 3D world plane). In this case, the target scale could not be evaluated directly and was estimated within a range from the scale at the previous frame. Tab. 3 reports the performance of our multi-target tracking performed in the two cases.

It can be observed that tracking in 3D world coordinates lowers the number of false positives and contributes to a sensible improvement in both accuracy and precision, with respect to tracking in the 2D image plane. This improvement is even greater as the number of targets increases since the tracker has to discriminate between them.

We compared our calibration and tracking against the results reported by a few authors, namely [16], [32] and [33],
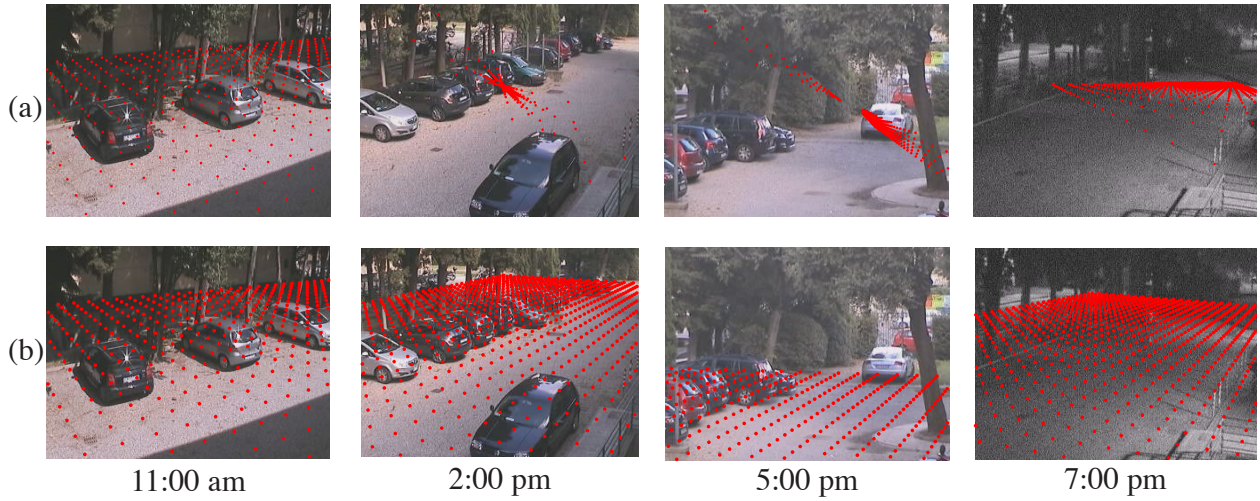
Fig. 11: Camera calibration without (a) and with scene map updating (b) at different time of the day.

Sequence and Method	CLEAR MOT						USC Metric			
	MOTA%	MOTP%	FN%	FP%	ID_SW	TR_FR	MT%	PT%	ML%	FAF
Seq. #1 (1 target)										
Our method w/o map updating	-89.9	58.4	70.8	118.2	0	23	0.0	100.0	0.0	1.17
Our method w/o proximity check	80.4	60.4	10.9	8.6	0	**1**	100.0	0.0	0.0	0.09
Our method	**88.2**	**66.7**	**10.9**	**0.7**	**0**	3	**100.0**	**0.0**	**0.0**	**0.01**
Seq. #2 (2 target)										
Our method w/o map updating	-130.0	52.1	96.1	133.0	0	27	0.0	0.0	100.0	2.49
Our method w/o proximity check	70.4	61.5	25.7	3.6	0	10	50.0	50.0	0.0	0.07
Our method	**78.8**	**64.2**	**19.4**	**1.6**	**0**	8	**50.0**	**50.0**	**0.0**	**0.03**
Seq. #3 (3 target)										
Our method w/o map updating	-51.5	59.4	81.9	69.1	0	20	0.0	66.7	33.3	2.06
Our method w/o proximity check	67.5	**67.3**	26.9	5.4	0	6	33.3	66.7	0.0	0.16
Our method	**74.6**	65.0	**24.3**	**1.0**	**0**	3	33.3	66.7	**0.0**	**0.03**

Table 2: Multi-Target Tracking Performance in different settings: with one, two, three moving targets.

Sequence and Method	CLEAR MOT						USC Metric			
	MOTA%	MOTP%	FN%	FP%	ID_SW	TR_FR	MT%	PT%	ML%	FAF
Seq. #1 (1 target)										
Our method in 2D	79.9	**70.6**	15.1	4.9	0	3	100.0	0.0	0.0	0.05
Our method in 3D	**88.2**	66.7	**10.9**	**0.7**	**0**	3	**100.0**	**0.0**	**0.0**	**0.01**
Seq. #2 (2 target)										
Our method in 2D	42.7	57.5	36.6	20.3	1	9	0.0	100.0	0.0	0.38
Our method in 3D	**78.8**	**64.2**	**19.4**	**1.6**	**0**	8	**50.0**	**50.0**	**0.0**	**0.03**
Seq. #3 (3 target)										
Our method in 2D	59.5	62.5	31.8	8.5	0	7	0.0	100.0	0.0	0.25
Our method in 3D	**74.6**	**65.0**	**24.3**	**1.0**	**0**	3	**33.3**	**66.7**	**0.0**	**0.03**

Table 3: Multi-Target Tracking Performance in 2D and 3D world coordinates.

on the *UBC Hockey* sequence [16]. This is the only publicly available dataset recorded from a PTZ camera. It is very short and includes frames of a hockey game. All these authors performed tracking in the 2D image plane. For the sake of completeness we have compared both the 2D and 3D version of our tracking method. The scene map was obtained by uniformly sampling the video sequence every ten frames so to have a full coverage of the scene. For a fair comparison, in a first experiment we compared our method against [16] using the original detections provided by Okuma. In a second experiment we compared with [32] and [33] using the ISM detector [31]. The results are reported in Tab. 4. As it is possible to observe, in the first experiment our calibration and tracking in 2D coordinates obtains comparable performance

Sequence and Method	CLEAR MOT						USC Metric			
	MOTA%	MOTP%	FN%	FP%	ID_SW	TR_FR	MT%	PT%	ML%	FAF
UBC Hockey (Okuma's detector)										
Okuma [16]	67.8	51.0	31.3	0.0	11	0	–	–	–	–
Our method in 2D	67.9	**62.3**	8.8	23.2	0	1	91.7	8.3	0	2.47
Our method in 3D	**90.3**	60.4	**6.5**	**3.1**	**0**	1	**91.7**	**8.3**	**0**	**0.35**
UBC Hockey (ISM [31] detector)										
Breitenstein [32]	76.5	57.0	22.3	1.2	0	–	–	–	–	–
Brendel [33]	79.7	60.0	19.5	**1.1**	0	–	–	–	–	–
Our method in 2D	72.6	61.0	18.7	8.6	0	1	58.3	33.3	8.3	0.93
Our method in 3D	**83.6**	**63.8**	**14.5**	1.9	**0**	**0**	75	16.7	8.3	**0.21**

Table 4: Multi-Target Tracking performance on UBC Hockey dataset.

as [16], while tracking in 3D world coordinates has significantly superior performance. In the second experiment, we observed that the ISM detector fails to detect a target in the entire sequence and determines a large number of false negatives in all the methods. Notwithstanding calibration and tracking in 3D coordinates still reports some improvement in performance with respect to the 2D solutions.

5.3 Operational Constraints and Computational requirements

We analyzed the operational constraints and computational requirements of our solution using a SONY SNC-RZ30P PTZ camera and Intel Xeon Dual Quad-Core at 2.8GHz and 4GB of memory, with no GPU processing. From Tab. 5 we can see that we perform real-time calibration and tracking (in 3D world coordinates) at 12 fps. The current implementation of the method exploits multiple cores and was developed in C/C++. Frame grabbing, camera calibration and scene map updating are performed in one thread, detection and tracking are performed in a separate thread.

Component	Time	fps
Camera Pose Estimation	88 ms	11
Scene Map Update	5 ms	200
Detection	43 ms	23
Tracking	35 ms	28
Total (Sequential)	171 ms	5
Total (Parallel)	83 ms	**(x2.4) 12**

Table 5: Computational requirements per processing module on a Intel Xeon Dual Quad-Core at 2.8GHz.

6 Conclusions

In this paper, we have presented an effective solution for online PTZ camera calibration that supports real-time multiple

target tracking with high and stable degree of accuracy. Calibration is performed by exploiting the information in the current frame and has proven to be robust to camera motion, changes of the environment due to illumination or moving objects and scales beyond thousands of landmarks. The method directly derives the relationship between the position of a target in the 3D world plane and the corresponding scale and position in the 2D image. This allows real-time tracking of multiple targets with high and stable degree of accuracy even at far distances and any zooming level.

References

1. R. Hartley, in *Proc. of the European Conference on Computer Vision* (1994)
2. L. de Agapito, E. Hayman, I.D. Reid., International Journal of Computer Vision **45**(2), 107 (2001)
3. S. Sinha, M. Pollefeys, in *Proc. of ECCV Workshops, Omnidirectional Vision and Camera Networks* (2004)
4. S. Sinha, M. Pollefeys, Computer Vision and Image Understanding **103**(3), 170 (2006)
5. G. Klein, D. Murray, in *Proc. of the IEEE and ACM International Symposium on Mixed and Augmented Reality* (2007)
6. B. Williams, G. Klein, I. Reid, in *Proc. of the IEEE International Conference on Computer Vision* (2007)
7. J. Civera, A.J. Davison, J.A. Magallon, J.M.M. Montiel, International Journal of Computer Vision **81**(2), 128 (2009)
8. S. Lovegrove, A.J. Davison, in *Proc. of European Conference on Computer Vision* (2010)
9. Z. Wu, R. Radke, IEEE Transactions on Pattern Analysis and Machine Intelligence **99**, 1994 (2012)
10. D. Song, K. Goldberg, in *IEEE International Conference on Robotics and Automation* (2006)
11. E. Hayman, T. Thorhallsson, D.W. Murray, in *Proc. of the International Conference on Computer Vision* (1999)
12. B. Tordoff, D. Murray, IEEE Transactions on Pattern Analysis and Machine Intelligence **26**(1), 98 (2004)
13. H. Tao, H.S. Sawhney, R. Kumar, IEEE Transactions on Pattern Analysis and Machine Intelligence **24**(1), 75 (2002)
14. P. Varcheie, G.A. Bilodeau, in *IEEE International Workshop on Robotic and Sensors Environments* (2009)
15. S. Kang, J.K. Paik, A. Koschan, B.R. Abidi, M.A. Abidi, in *International Conference on Quality Control by Artificial Vision* (2003)
16. K. Okuma, A. Taleghani, N. de Freitas, J.J. Little, D.G. Lowe, in *Proc. of the European Conference on Computer Vision* (2004)
17. N.d.F. Yizheng Cai, J. Little, in *Proc. of the European Conference on Computer Vision* (2006)

18. Y. Seo, S. Choi, H. Kim, K.S. Hong, in *Proc. of the International Conference on Image Analysis and Processing* (1997)
19. L. Barceló, X. Binefa, J.R. Kender, in *Proc. of the International Conference on Image and Video Retrieval* (2005)
20. A. Del Bimbo, G. Lisanti, F. Pernici, in *Proc. of ICCV Workshops, International Workshop on Visual Surveillance* (2009)
21. A. Del Bimbo, G. Lisanti, I. Masi, F. Pernici, in *Proc. of CVPR Workshops, Socially Intelligent Surveillance and Monitoring* (2010)
22. H. Bay, T. Tuytelaars, L.V. Gool, in *Proc. of the European Conference on Computer Vision* (2006)
23. D.G. Lowe, International Journal on Computer Vision **60**(2), 91 (2004)
24. A. Criminisi, I. Reid, A. Zisserman, Image and Vision Computing pp. 625–634 (1999)
25. F. Pernici, A.D. Bimbo, IEEE Transactions on Pattern Analysis and Machine Intelligence **99**, 1 (2013)
26. D. Liebowitz, A. Zisserman, in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition* (1998)
27. R.J. Fitzgerald, IEEE Transactions on Aerospace and Electronic Systems, **AES-21** (1985)
28. N. Dalal, B. Triggs, in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition* (2005)
29. K. Bernardin, R. Stiefelhagen, Journal on Image and Video Processing, **2008,**, 1
30. B. Yang, C. Huang, R. Nevatia, in *Proc. of IEEE Conference on Computer Vision and Patter Recognition* (2011)
31. B. Leibe, A. Leonardis, B. Schiele, International Journal of Computer Vision **77**(1-3), 259 (2008)
32. M. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, L. Van Gool, IEEE Transactions on Pattern Analysis and Machine Intelligence **99**, 1820 (2010)
33. W. Brendel, M. Amer, S. Todorovic, in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition* (2011)