Device-Tagged Feature-based Localization and Mapping of Wide Areas with a PTZ Camera

Alberto Del Bimbo Giuseppe Lisanti Iacopo Masi Federico Pernici
{delbimbo, lisanti, masi, pernici}@dsi.unifi.it

Media Integration and Communication Center (MICC)
University of Florence, Viale Morgagni, 65 (FI) - Italy

Abstract

This paper proposes a new method for estimating and maintaining over time the pose of a single Pan-Tilt-Zoom camera (PTZ). This is achieved firstly by building offline a keypoints database of the scene; then, in the online step, a coarse localization is obtained from camera odometry and finally refined by visual landmarks matching. A maintenance step is also performed at runtime to keep updated the geometry and appearance of the map.

At the present state-of-the-art, there are no methods addressing the problem of being operative for a long period of time. Also, differently from our proposal these methods do not take into account for variations in focal length.

Experimental evaluation shows that the proposed approach makes it possible to deliver stable camera pose tracking over time with hundreds of thousand landmarks, which can be kept updated at runtime.

1. Introduction

In recent years, pan-tilt-zoom cameras are becoming increasingly common, especially for use as surveillance devices in large areas. Despite its widespread usage, there are still issues to be resolved regarding their effective exploitation for scene understanding at a distance. A typical operating scenario is that of abnormal behavior detection which requires both simultaneous target trajectories analysis on the 3D ground plane and the indispensable image resolution to perform target biometric recognition. This cannot generally be achieved with a single stationary camera mainly because of the limited field of view and poor resolution with respect to scene depth. PTZ sensors indeed assure a superior image quality (for example as zoom increases towards distant objects in the scene) and this will be crucial for the task of managing the sensor to detect and track several moving targets at a distance. To this end we are interested in the acquisition and maintenance of camera pose estimation, relative to some geometric 3D representation of its surroundings, as the sensor performs pan-tilt and zoom operations. This allows continuous dynamic-calibration of the sensor for a tracking system based on target scale inference and robust data-association [9].

1.1. Related Work

In its basic form this approach resembles the Simultaneous Localization And Mapping (SLAM) [10] made more challenging by the requirement of detection and tracking of moving objects. Nevertheless there are some important differences between the standard Visual monocular SLAM (monoSLAM) [8] and the PTZ-SLAM: monoSLAM assumes a static environment and known internal camera parameters, considering a freely moving observer (6DOF). On the contrary PTZ-SLAM must solve for 8DOF (three for the pose and five for the intrinsic parameters) and deals with dynamic environments. Civera *et al.* [6] present a sequential mosaicing algorithm for an internal calibrated rotating camera using an Extended Kalman Filter SLAM approach. Authors point out that their results are mainly occurred because a rotating camera is a constrained and well-understood linear problem. Here, we argue that tracking a zooming camera is much more difficult, because the structure of the problem is highly non-linear and near-ambiguities may arise when the perspective effects are small, due to large focal lengths [1]. Such difficulties make it unsuitable for most real-world PTZ surveillance scenarios and motivate the development of a system able to perform loop closure[1] over an extended area.

To avoid data association errors of EKF-SLAM based methods, Klein and Murray propose the Parallel Tracking and Mapping (PTAM) for small workspace [11] that employs standard Structure from Motion (SfM) bundle adjust-

[1]Loop closure refers to data association between two distinct parts of the map even when tracking is proceeding smoothly.

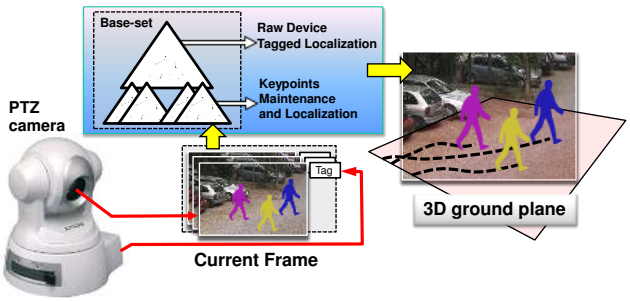


Figure 1. Components of the proposed framework. The current frame is associated with a tag provided by the device. The tag contains information about internal machine encoder values. (*Top*): Localization and Mapping component. (*Right*): 3D ground plane estimation for multiple target tracking system [9].

ment [13] to build a map from many views. PTAM separates in two threads the task of maintaining the camera track at frame-rate from the less stringent task of optimally updating the 3D structure from measurements (made in keyframes only). Despite the positive results, this system could not be applied to the PTZ-SLAM due to the cost of solving online the bundle adjustment considering a wide area.

An alternative approach to the "hybrid tracking" of PTAM is repeated data-driven detection of pose, which does not require any motion model on pose estimation. Tracking by detection approach proposed in [14] follows this method and demonstrates that real-time repeated detection of pose via a forest of fast classifiers can be used to establish correspondences between image and target features.

In the case of PTZ-SLAM the matching system should be robust to changes in viewpoint/scale, blur/lighting, moving objects and other distractors; it must be fast enough to run online without any drift in the map geometry to ensure stability to multiple target tracker (MTT) [9] that uses camera pose estimation. Imagine a 24/7 monitoring application of parking: the whole appearance of the provided scene must be updated with the same wealth of details that exists in background maintenance [15, 2].

To overcome all above problems we build on the idea of textual tagged images [7]. Geo-located tags, are used to categorize images considering both spatial and visual information at respectively urban, regional or global scale. We adopt the same concept by tagging images taken with a PTZ camera at some raw camera pose given by the device. According to this, the device tag is hence used to retrieve keypoints[2] associated to a pre-build camera pose obtained in a offline learning stage. The final camera pose is computed by refining pre-build camera pose with respect to the current view through keypoint matching. Finally a novel step of keypoints tracking and updating is performed in order to add new/stable keypoints to the base-set and to prune unused ones. Our proposed approach differs from the current state-of-the-art in the following aspects: 1) precise, real-time camera pose estimation with very large focal length; 2) robust keypoints maintenance with respect to background changes that occur naturally in a scene; we demonstrate that long-time robust camera pose estimation is possible even if only few initial landmarks are present; 3) the retrieval accuracy (of the combined text and distinctive keypoints retrieval) and the total absence of any prior on motion models allow for robust data association and map estimation.

[2]The terms keypoint, feature and landmark are used interchangeably throughout the paper.

2. Offline Learning of the Scene

The wide area observed by the PTZ sensor is accurately described by a base set of M images taken at different values of pan, tilt and zoom. This base-set $\mathcal{B}$ of reference views is defined as:

$$\mathcal{B} = [\mathbf{y}_i, \{\mathbf{f}_j^i\}_{j=1}^n, \mathtt{H}_W, \mathtt{H}_{ih} : i = 1..M] \quad (1)$$

where:

- $\mathbf{y}_i \in \mathbb{R}^3$ is the *raw device-tagged PTZ value*, relative to the reference view $\mathbf{I}_i$ acquired from the sensor;
- $\mathbf{f}_j^i$ is the j^{th} keypoint of the i^{th} reference view;
- $\mathtt{H}_W : \Pi_{3D} \mapsto \Pi_h$ is the world-to-image homography that maps the 3D ground plane on the reference plane Π_h.
- $\mathtt{H}_{ih}$ is a family of homography that relates the i^{th} view onto a common reference plane Π_h, where h is a view selected from the set.

Each homography $\mathtt{H}_{ih}$ is parameterized by two rotation angles (pan and tilt) and by focal length [1]. These parameters are estimated offline with global bundle adjustment [13] as described in [5]. World-to-image homography $\mathtt{H}_W$ is precalculated offline as described in [9].

2.1. Device-Tagged Coarse Localization

Real time Localization is achieved by first performing search over the device-tagged reference views. The current tag is compared with the set of tags in the base-set in order to retrieve the set of features (stored in a k-d tree) for the nearest reference view (see Fig. 2). The recovered k-d tree is further queried to retrieve the set of potential keypoints matching the current frame. From these correspondences the homography $\mathtt{H}_t : \mathbf{I}_t \mapsto \mathbf{I}_i$, mapping the current frame onto the retrieved view, is estimated with RANSAC. The time variant world-to-image homography $\mathtt{G}_\mathbf{t} : \Pi_{3D} \mapsto \mathbf{I}_\mathbf{t}$ is finally obtained as in [9]:

$$\mathtt{G}_t = \underbrace{\mathtt{H}_t^{-1}}_{\mathbf{I}_t \leftarrow \mathbf{I}_i} \cdot \underbrace{\mathtt{H}_{ih}^{-1}}_{\mathbf{I}_i \leftarrow \Pi_h} \cdot \underbrace{\mathtt{H}_W}_{\Pi_h \leftarrow \Pi_{3D}} \tag{2}$$

Localization performs real-time due to the fact that: 1) the keypoints detection and description is performed by exploiting Speed Up Robust Feature (SURF) [3]; 2) at each time only a region of the field of regard[3] is updated by rebuilding the local *k*-d tree.

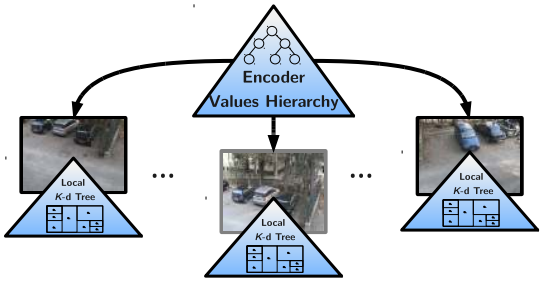


Figure 2. The structure used for real-time localization of the PTZ sensor. Raw camera encoder values are used to index the *k*-d tree of each reference view $\mathbf{I}_i$.

3. Real-Time Lifelong Mapping

Features update and maintenance is formalized as follows: given the model set of keypoints $\mathcal{F}_m \doteq \{\mathbf{f}_j\}_{j=1}^n$ in eq. (1) extracted from the scene at time t, choose online which ones keep in the set, remove or update, taking as observation the set of points of the current scene at time $\bar{t} \gg t$, defined as $\mathcal{F}_c \doteq \{\mathbf{f}_k\}_{k=1}^m$. The aim is to maintain consistency among the offline $\mathcal{F}_m$ features database with respect to current keypoints $\mathcal{F}_c$ so as to preserve a database of updated features that allows continuous dynamic-calibration.

To this aim the SURF keypoint $\mathbf{f}_j$ is extended as follows to create a time-invariant feature descriptor:

$$\mathbf{f}_j = \{ID\ , X_t\ ,\ T(\tau, t)\ ,\ \overline{\mathbf{d}}\} \tag{3}$$

where:

- ID is the identifier of a landmark.
- $X_t \sim \mathcal{N}(\mu_X, \Sigma_X)$ is a 2D Gaussian random variable where μ_X represents the averaged keypoint location and Σ_X is the relative covariance matrix.
- $T(\tau, t)$ is a random variable that describes the lifetime of each keypoint in the model.
- $\overline{\mathbf{d}} \in \mathbb{R}^{128}$ is the local descriptor updated over time.

Keypoints matching is performed in the local *k*-d tree according to approximate nearest neighbor search, as in [4].

[3]The camera field of regard (FOR) is defined as the union of all field of view over the entire range of pan and tilt rotation angles and zoom values.

3.1. Map Geometry Updating

To avoid drift errors and preserve the global optimization made offline, the geometry of each keypoint is estimated into the matched reference view $\mathbf{I}_i$. Furthermore, the noise introduced by the Fast-Hessian detector is limited by performing recursive time average of the keypoint geometry X_t , without any adaptation, as:

$$\mu_{X_t} = \frac{t-1}{t} \cdot \mu_{X_{t-1}} + \frac{1}{t} \cdot Z_t \tag{4}$$

$$\sigma_{X_t}{}^2 = \frac{t-1}{t} \cdot \sigma^2_{X_{t-1}} + \frac{1}{t-1} \cdot (\mu_X - Z_t)^2 \tag{5}$$

where $Z_t \in \mathbb{R}^2$ is current SURF keypoint detection back-projected on the reference view $\mathbf{I}_i$.

3.2. Map Landmark Updating

During tracking each keypoint undergoes some changes in appearance due to structural variations provided ,for example, by a car that enters or leaves the scene, or by gradual changes according to sunlight.

3.2.1 Birth and Death Process of Visual Landmarks

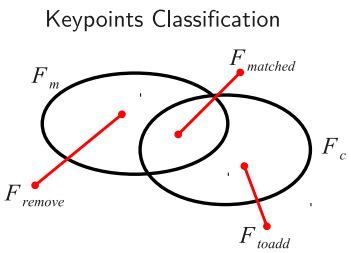


Figure 3. The set of points in the model $\mathcal{F}_m$ is bi-partitioned between points to keep and remove. The current set of points $\mathcal{F}_c$ instead is bi-partitioned between the matched points and those candidate to be inserted.

Insertion and deletion of landmarks are obtained by the intersection of the current keypoints set $\mathcal{F}_c$ with respect to the model-set $\mathcal{F}_m$, as shown in Fig. 3. Keypoints are classified as follows :

- $\mathcal{F}_{matched} \doteq \{\mathcal{F}_m\} \bigcap \{\mathcal{F}_c\}$ represents keypoints which remain in the model.
- $\mathcal{F}_{remove} \doteq \{\mathcal{F}_m\} \setminus \{\mathcal{F}_{matched}\}$ represents the set of keypoints candidate to be removed.
- $\mathcal{F}_{to\ add} \doteq \{\mathcal{F}_c\} \setminus \{\mathcal{F}_{matched}\}$ represents the subset of keypoints from which new features will be selected.

To speed up feature matching we perform a random sampling step on $\mathcal{F}_c$; this allows also for some feature randomization which effectively makes RANSAC less likely to fail

in subsequent frames. According to this, a landmark removal probability is defined as follows: a keypoint that has not been seen for a long time will gain a high probability to exit out from the set. This is motivated by the fact that it has been part of $\mathcal{F}_{remove}$ for many times. On the contrary a keypoint that was just observed will extend its lifetime since that it has been ranked in the $\mathcal{F}_{matched}$. In this way, we achieve a dynamic keypoint modeling with an effective birth and death process of landmarks.

The lifetime of each discriminative keypoint is modeled as a negative exponential random variable $T(\tau, t) \sim \tau e^{-\tau t}$ of eq. (3) that varies with time in order to simulate a memory decay process (i.e. the loss of memory over time). Under such assumption at each instant the lifetime is given by a p.d.f. parameterized by $\tau \geq 0$ where τ models the tendency for the keypoint of being removed. Each time there is no observation the parameter τ is incremented, otherwise it is set to zero. The death probability of a keypoint at time t is then given by:

$$p_{death}(T < t|\tau) = \int_0^t \tau \cdot e^{-\tau x} dx = 1 - e^{-\tau \cdot t} \quad (6)$$

On the contrary the birth process, tackling structural changes, is based on the set given by the classification refinement step, previously defined as $\mathcal{F}_{to\ add} \doteq \{\mathcal{F}_c\} \setminus \{\mathcal{F}_{matched}\}$, representing new landmarks in the scene. Since this set at time t contains some temporary outliers of the model (i.e. features that do not belong to the scene), we track them for a limited period in order to select the keypoints that will be able to enter the model.

The basic idea is to compare two sets of candidates so as to finally decide which points to include as background: 1) at time t the set of keypoints candidate to enter the model $\mathcal{F}^t_{to\ add}$ is taken into account; 2) after a temporal window δt, at time $t + \delta t$ a new candidate set $\mathcal{F}^{t+\delta t}_{to\ add}$ is obtained; 3) the keypoints correspondences between these two sets make a new set $\mathcal{F}^{t+\delta t}_{add}$ that will become part of the map. Otherwise, items without correspondences will be labeled as foreground $\mathcal{F}^{t+\delta t}_{foreground}$ and will be discarded.

The parameter δt is used to refine the keypoints selection step that will decide about which landmarks may be included. Low values for δt may let in landmarks that are not required (e.g. foreground points generated from shadows), while high values may void the research between the two sets. In order to maintain persistence in the scene appearance we choose $\delta t = 20$.

It is important to remark that not every keypoints will be added to $\mathcal{F}_m$ with the same probability, since they are mapped to the reference view $\mathbf{I}_i$ through $\mathtt{H}_t$. Indeed to stabilize the estimated camera pose we keep updated the map by firstly adding points with less uncertainty. Only after that, keypoints with a larger uncertainty are considered.

This "exploration vs exploitation" trade-off governs the process of adding new keypoints, while exploiting less uncertainty. We proceed similarly as in [12] following the idea that adding points far from the $\mathcal{F}_{matched}$ may provide worse performance. Therefore we define the probability of a point to enter the model as the ratio between the minimum bounding box area that contains the convex hull of the $\mathcal{F}_{matched}$, and the same bounding box temporarily expanded with the new feature to add. This process alone however is not enough to deal with gradual changes in landmark appearance. Therefore each landmark descriptor is updated by running average ($\alpha = 0.25$ is used):

$$\overline{\mathbf{d}}_r = (1 - \alpha) \cdot \overline{\mathbf{d}}_r + \alpha \cdot \mathbf{d}_r \ \ \forall r = 1..128 \quad (7)$$

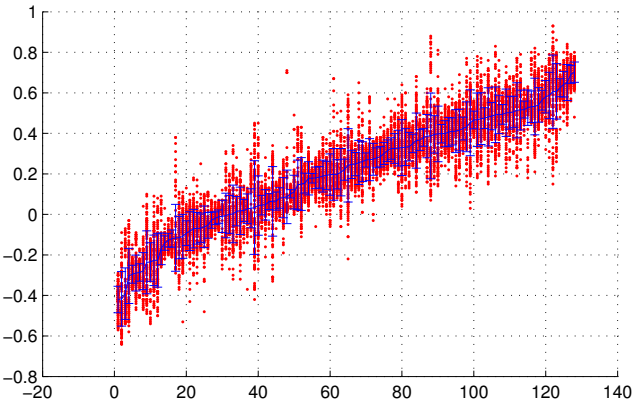


Figure 4. Representation of changes occurred in a keypoint descriptor over time. Error bars (blue) show the standard deviation across realizations (red). Descriptor components are plotted in ascending order.

4. Experimental Results

We use a real dataset taken from a wide parking area with an off-the shelf SONY PTZ camera placed at about seven meters in height. The scene is learned, as described in Sec. 2, from about $n = 300$ reference views (acquired at different levels of pan, tilt and zoom) which results in about $150,000$ keypoints for the whole initial database. The video stream and the reference views have a resolution of 320×240 pixels. Our system implementation runs in real-time at 30 fps on a Intel Xeon Quad-Core at 2.93 GHz. We performed qualitative and quantitative experiments showing the effectiveness of the system to handle with structural changes of the scene.

Regarding qualitative experiments, Fig. 5 shows a detailed example of structural change caused by a car that has left the scene. Fig. 5*top-right* shows the reference view $\mathbf{I}_i$ chosen by the raw camera localization system. Fig. 5*bottom-right* is the current frame $\mathbf{I}_t$. The maintenance process automatically increases the death probability $p_{death}(\mathbf{f}_i)$ for each keypoint of the car (in yellow), as shown

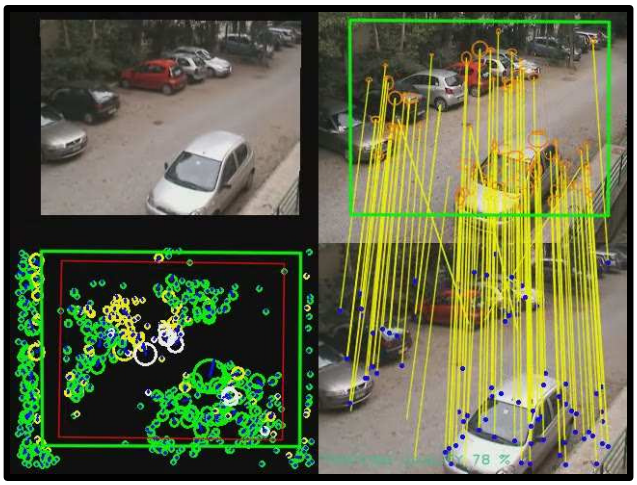


Figure 5. Keypoints birth and death process, tackling structural changes.

in Fig. 5*bottom-left*. This procedure ensures good stability for the estimation of $\mathtt{H}_t$ as shown in Fig. 5*top-left*.

Fig. 6 shows in the top row a grid on the 3D ground plane superimposed using $\mathtt{G}_t$ at different hours of the day. Despite the presence of moving objects or strong scene variations the imaged grid does not drift over time. In each frame of Fig. 6 in the middle row is shown the adaptive features-set with current camera pose estimation (in purple). The approach is robust to illumination changes produced by the camera automatic night-mode switch (last frame of Fig. 6).

In Fig. 7 is shown the performance obtained by the features maintenance procedure in relation to dynamic landmarks modeling. In particular, the green curve indicates the insertion process, the blue curve indicates the removal process and the red curve denotes the number of maintained keypoints over time. In Fig. 8 is also shown a comparison of our method with respect to a non-adaptive one [9] and it is possible to observe that our approach (Fig. 8 *red curve*) provides a suitable numbers of inliers to fit $\mathtt{H}_t$ over time while the robustness of the other approach (Fig. 8 *blue curve*) decreases over time.

Fig. 9 shows geometry localization error for the estimated position of a keypoint that enters the model at frame

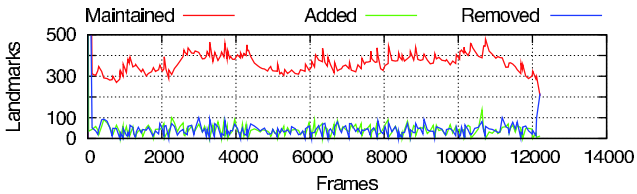


Figure 7. Maintenance procedure performance: the green curve represents the insertion process, the blue curve is the trend of the removal process and the red curve denotes the number of maintained keypoints.

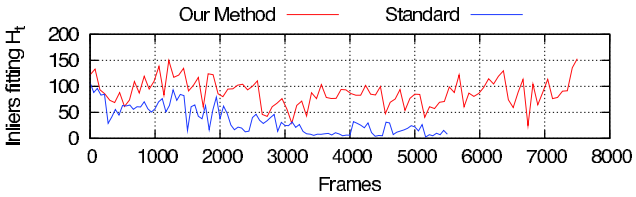


Figure 8. These two curves illustrate the number of inliers available over time. The red curve represents the proposed method while the blue curve represents the technique proposed in [9].

1700 and is updated until its end, keeping a sub-pixel error accuracy.

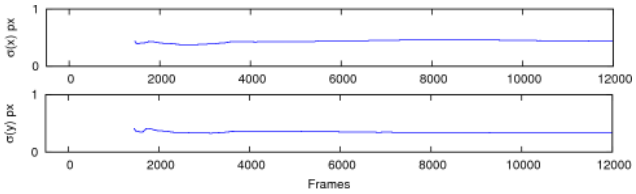


Figure 9. Subpixel accuracy in the estimated position for keypoint with ID 3201.

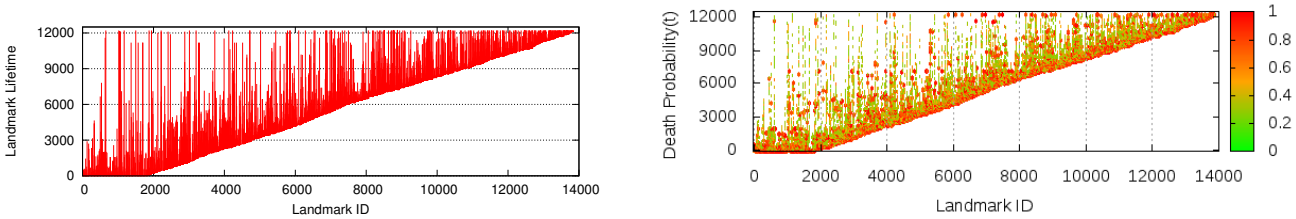
From an accurate analysis of the birth and death process (Fig. 10) it is possible to see that only few keypoints extracted offline (those with ID $\in$ [0..2000]) are still alive at the end of the experiments and this number is not suitable for good calibration. Landmarks added by the online procedure (those with ID $\geq$ 2000) ensure a very good stability for the pose estimation as shown in Fig. 10 (a). This histogram shows on the x-axis the landmark ID and on the y-axis the history of a each keypoint from its birth to its death. Fig. 10 (b) shows instead the death probability of a keypoint for each frame, depending on the parameter τ. The probability is plotted according to a color gradient from green (stable landmark with low death probability) to red (landmark with high death probability, ready to exit the model).

5. Conclusion

In this paper we have presented a method for the challenging task of updating large set of visual landmarks in real-time while performing PTZ camera pose estimation. The result is a robust mapping system that produces unbiased detailed map of a dynamic environment and provides stable dynamic-camera calibration for 3D ground plane estimation. Future research may include a recovery procedure that avoids losing camera tracking whenever a part of the wide area is not updated for a long time.

Acknowledgment. This work is partially supported by Thales Italia, Florence, Italy.

Figure 6. *(Top row)*: Superimposed grid of the 3D ground plane. *(Middle row)*: The adaptive feature-set over time. *(Bottom row)*: The reference view $\mathbf{I}_i$ selected by the localization system.



(a) (b)

Figure 10. (a) The history of lifetime for each keypoint observed in a region of the FOR. (b) The histogram of death probability for each visual landmark.

References

[1] L. D. Agapito, E.Hayman, and I.Reid. Self-calibration of rotating and zooming cameras. *International Journal of Computer Vision (IJCV)*, 45:2, 2001.

[2] G. Baugh and A. Kokaram. Feature-based object modelling for visual surveillance. *International Conference on Image Processing*, 2008.

[3] H. Bay, T. Tuytelaars, V. Gool, and L. Surf: Speeded up robust features. In *Internation Conference on Computer Vision*, 2006.

[4] J. Beis and D. Lowe. Indexing without invariants in 3d object recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(10):1000–1015, October 1999.

[5] M. Brown and D. Lowe. Recognising panoramas. In *Proceedings of the 9th International Conference on Computer Vision*, 2003.

[6] J. Civera, A. J. Davison, J. A. Magallón, and J. M. Montiel. Drift-free real-time sequential mosaicing. *International Journal of Computer Vision*, 81(2):128–137, 2009.

[7] M. Cristani, A. Perina, U. Castellani, and V. Murino. Geo-located image analysis using latent representations. In *International Conference on Computer Vision and Pattern Recognition*, 2008.

[8] A. J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *International Conference on Computer Vision*, 2003.

[9] A. Del Bimbo, G. Lisanti, and F. Pernici. Scale invariant 3d multi-person tracking using a base set of bundle adjusted visual landmarks. In *Proc. of ICCV International Workshop on Visual Surveillance*, 2009.

[10] H. Durrant-Whyte. Uncertain geometry in robotics. In *Proceedings of IEEE International Conference on Robotics and Automation.*, 1987.

[11] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *International Symposium on Mixed and Augmented Reality*, 2007.

[12] S. Negahdaripour, R. Prados, and R. Garcia. Planar homography: accuracy analysis and applications. *International Conference on Image Processing*, 2005.

[13] B. Triggs, P. Mclauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment - a modern synthesis. *Proceedings of the International Workshop on Vision Algorithms*, 2000.

[14] M. O. Vincent, V. Lepetit, F. Fleuret, and P. Fua. Feature harvesting for tracking-by-detection. In *European Conference on Computer Vision*, 2006.

[15] Q. Zhu, S. Avidan, and K.-T. Cheng. Learning a sparse, corner-based representation for time-varying background modelling. *International Conference on Computer Vision*, 2005.