A Visual Compass based on SLAM

J. M. M. Montiel
Dpt. de Informatica e Ingenieria de Sistemas
Universidad de Zaragoza. Spain
Email: josemari@unizar.es

Andrew J. Davison
Department of Computing
Imperial College London. UK
Email: ajd@doc.ic.ac.uk

***Abstract*— Accurate full 3 axis orientation is computed using a low cost calibrated camera. We present a simultaneous sensor location and mapping method that uses a purely rotating camera as sensor and distant points, ideally at infinity, as features. A smooth constant angular velocity pure rotation motion model codifies the camera location. Because of the sequential EKF approach used, and the number of features in the map, about a hundred, the proposed method has been implemented in real time at standard video rates.**

Experimental results with real images show that the system is able to close loops with 360° pan and 360° cyclotorsion rotations. Sequences show good performance under challenging conditions: hand-held camera, varying natural outdoor illumination, low cost camera and lens and people moving in the scene.

I. Introduction

Despite the overwhelming biological evidence for the suitability of vision as a primary sensor for simultaneous sensor location and map building, it has proven difficult to match those capabilities using the state of the art digital hardware. Recently however, the breakthrough work by Davison [1], has proven the possibility of doing real time visual SLAM with standard low cost camera hardware.

Our goal is to build full 3 axis visual compass using standard low cost computer vision hardware, making the most of the prior information available, namely a calibrated wide angle camera gathering sequences at 30 fps. while undergoing smooth motion, and observing a scene where points are at distances much bigger than the camera translation. We closely observe the real time constraint using sequential algorithms able to run online; in fact, between submission and acceptance of the paper, we have a C++ version successfully running at 30 fps for maps up to 100 features.

Our work has strong links with [1]; it can be considered a specialization of this work for a rotating camera and features at infinity, in fact we use for the camera rotation part exactly the same motion model. In any case both works are quite complementary because [1] cannot cope with the rotating camera case nor with features at infinity. It is focused on close to camera features because it is difficult to represent depth uncertainty for distant ones. We consider this visual compass as the first step for real time outdoor visual SLAM, where we expect to have both close and distant points.

Some of the advantages of the visual compass system in this paper are:

- Instant start up from any camera orientation or velocity. No initialisation step or known scene objects are needed. In fact the tracking can be started from any frame of the sequence.
- Good performance rejecting non rigid motion objects.
- Routine loop closing. Our experimental results show that standard sequential processing can deal with loop closing without any special steps. This is a desirable feature for a real-time system.

Davison's work alloys basic vision point matching techniques, gaussian SLAM and particle filtering to detect the underlying geometry in a sequence observed by a moving hand-held camera. This geometry considers a selected reduced set of trackable 3D points (the map), and the camera motion jointly in the same random vector. This joint modeling allows the observation of the very same features in all the images and the ability to close loops. The loop closing prevents an explosion in the number of features when the sensor moves indefinitely in a confined area revisiting previously observed areas. Image sequences repeatedly observing the same environment have a underlying model that SLAM methods detect and exploit.

The standard approach to SLAM is based on the stochastic map proposed initially by Smith and Cheesman in [2] which offers a powerful tool to model the geometrical location errors. It has been widely used in mobile robotics for processing geometrical information coming for a range of different sensors, odometry, laser range finder, sonar, and vision among others [3]–[6].

In the computer vision literature the family of methods known as Structure From Motion, (SFM), focus on the redundancy that several observations of the same scene have because all the matched features shuould be consistent with a projective camera model. These methods can deal with small sets of shots and with sequences. During the last decade (see [7] for a review) understanding of this underlying geometry and its recover by means of robust statics has produced working methods for dealing with wide-baselines and sequences with variable and unknown camera parameters. SFM primarily intended for batch processing, has been successfully applied for real time robotics in [8]. SFM methods do not model the camera an the scene in a unique random vector, so the observed features along the sequence are not intended to be persistent, and this makes difficult loop closing. Not dealing with loops has two main drawbacks: the first is that the size of the map for a confined area increases indefinitely, and secondly the tight constraints that a loop close enforces in the reconstruction, sequential or batch are also missed.

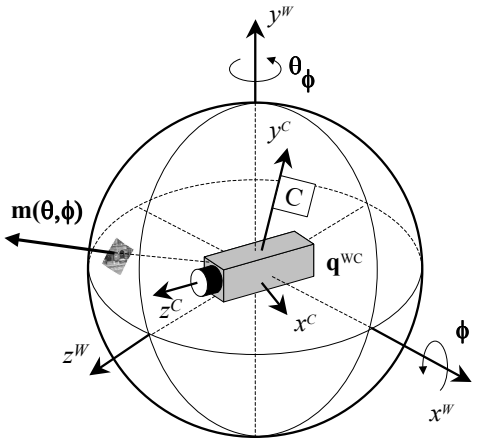

Fig. 1. The infinite point $\mathbf{y}_i$ coded by its θ (azimut, around y^W), and ϕ (elevation, around x^W;) and a texture patch gathered when imaged for the first time. The camera is located by the rotating reference C, defined by the rotation matrix R^{WC} or the quaternion $\mathbf{q}^{WC}$

A purely rotating camera offers a particular geometry in which depth cannot be recovered, but that provides one to one homography (if the distortion is compensated) mapping between every image pair. In a real scene this geometry appears when the distance of the observed features is much bigger (ideally at infinity), than the camera translation. This geometry has been used for producing mosaics, with robust statistics batch approach, e.g. [9] for a rotating and zooming camera. More recently, Brown and Lowe, [10] use this geometry and for producing mosaics from discrete images under severe wide-baseline constraints by using SIFT invariant feature detectors [11].

Sets of parallel lines, when imaged under a projective camera intersect at a vanishing point, which actually is the the image of the point at infinity where the parallel lines meet. The detection of vanishing points corresponding to dominant directions in indoors or outdoors man made "Manhattan like" environments allows to compute camera orientation from single frames. This approach has been followed in [12], and [13]. Compared to ours, they focus on the detection of predefined infinite direction only available in man-made environments; their main advantage is that camera translation is not constrained to be small compared to the distance to the scene.

II. Feature Models

Our system deals with directions, not considering at all the depth of the observed features nor the camera translation, so points can be regarded as points at infinity. Each gazing direction is modeled registering both geometrical and photometrical information (see Fig 1). The geometry for direction is coded as a angle azimut-elevation pair to avoid overparametrization:

$$\mathbf{y}_i = \begin{pmatrix} \theta_i & \phi_i \end{pmatrix}^\top$$

Alternatively the direction can be coded by means of a unit directional vector $\mathbf{m}_i$:

$$\mathbf{m}_i^W = \begin{pmatrix} \cos\phi_i \sin\theta_i & -\sin\phi_i & \cos\phi_i \cos\theta_i \end{pmatrix}^\top \quad (1)$$

The photometrical part stores a texture patch around the image of the point, extracted when the point is imaged for the first time. It used for correlation based recognition and accurate location of matching points inside the predicted uncertainty ellipse.

III. Camera motion model

The camera motion is modelled as a smoothly rotating camera. Camera orientation is defined by means of a quaternion $\mathbf{q}^{WC} = \begin{pmatrix} q_0 & q_x & q_y & q_z \end{pmatrix}^\top$.

A constant velocity model is used for the smooth camera motion. The state vector for the camera, $\mathbf{x}_v$:

$$\mathbf{x}_v = \begin{pmatrix} \mathbf{q}^{WC} \\ \omega^W \end{pmatrix}.$$

where ω^W is the camera angular velocity.

It is assumed an unknown angular acceleration input $\alpha^W = \begin{pmatrix} \alpha_x & \alpha_y & \alpha_z \end{pmatrix}^\top$ that is a gaussian processes of zero mean and known covariance matrix $\mathbf{P}_\alpha$. The effect of the unknown input during every processing step Δt produces an impulse of angular velocity: $\mathbf{n} = \Omega^W = \alpha^W \Delta t$. So the camera state prediction equation is:

$$\mathbf{f}_v(\mathbf{x}_{v,k}, \mathbf{n}) = \begin{pmatrix} \mathbf{q}_{k+1}^{WC} \\ \omega_{k+1}^W \end{pmatrix} = \begin{pmatrix} \mathbf{q}_k^{WC} \times \mathbf{q}\left(\left(\omega_k^W + \Omega^W\right)\Delta t\right) \\ \omega_k^W + \Omega^W \end{pmatrix}$$

IV. Measurement Model

One of the advantages of a projective camera is its ability to observe points at infinity. First we will consider the projection of infinite points in an ideal projective camera. Next subsection is devoted to the lens distortion compensation.

The camera orientation is defined by $\mathbf{q}^{WC}$, from there it can be computed R^{CW}, which transform the coordinates, between references W and C. So the coordinates of $\mathbf{m}$ expressed in C are:

$$\mathbf{m}^C = R_{CW}(\mathbf{q}^{WC})\mathbf{m}^W \quad (2)$$

Hence the undistorted projective image of the infinite point $\mathbf{m}$ is defined by:

$$\mathbf{h}_u = \begin{pmatrix} u_u \\ v_u \end{pmatrix} = \begin{pmatrix} u_0 - \frac{f}{d_x}\frac{\mathbf{m}_x^C}{\mathbf{m}_z^C} \\ v_0 - \frac{f}{d_y}\frac{\mathbf{m}_y^C}{\mathbf{m}_z^C} \end{pmatrix} \quad (3)$$

Where, u_0,v_0 are the camera center in pixels, f is the focal length and, d_x and d_y the pixel size.

A. Distortion model

We are using a wide angle lens and the system is showing sensibility with respect to camera calibration, because of that we use the two parameters radial distortion model commonly used in photogrammetry (see [14]). Compared with the distorsion model proposed in [15], ours is more expensive but more accurate as well. Our main cost is that the model has

not closed form for the distorsion of coordinates nor for its jacobian.

To recover the ideal projective undistorted coordinates $\mathbf{h}_u = (u_u, v_u)^\top$, from the actually distorted ones gathered by the camera, $\mathbf{h}_d = (u_d, v_d)^\top$, the next formulas are applied:

$$\begin{aligned} \mathbf{h}_u \begin{pmatrix} u_d \\ v_d \end{pmatrix} &= \begin{pmatrix} u_0 + (u_d - u_0)\left(1 + \kappa_1 r_d^2 + \kappa_2 r_d^4\right) \\ v_0 + (v_d - v_0)\left(1 + \kappa_1 r_d^2 + \kappa_2 r_d^4\right) \end{pmatrix} \\ r_d &= \sqrt{\left(d_x\left(u_d - u_0\right)\right)^2 + \left(d_y\left(v_d - v_0\right)\right)^2} \end{aligned} \quad (4)$$

To compute the distorted coordinates from the undistorted:

$$\mathbf{h}_d \begin{pmatrix} u_u \\ v_u \end{pmatrix} = \begin{pmatrix} u_0 + \frac{(u_u - u_0)}{\left(1+\kappa_1 r_d^2+\kappa_2 r_d^4\right)} \\ v_0 + \frac{(v_u - v_0)}{\left(1+\kappa_1 r_d^2+\kappa_2 r_d^4\right)} \end{pmatrix} \quad (5)$$

$$r_u = r_d\left(1 + \kappa_1 r_d^2 + \kappa_2 r_d^4\right) \quad (6)$$

$$r_u = \sqrt{\left(d_x\left(u_u - u_0\right)\right)^2 + \left(d_y\left(v_u - v_0\right)\right)^2} \quad (7)$$

Being the problem that formula (5) depends on r_d that has to be solved from (6), e.g using Newton-Raphson, once r_u is computed from (7).

In summary:

$$\mathbf{h}\left(\mathbf{y}_i, \mathbf{q}^{WC}\right) = \mathbf{h}_d\left(\mathbf{h}_u\left(\mathbf{m}_C(\mathbf{y}_i, R_{CW}(\mathbf{q}^{WC}))\right)\right) \quad (8)$$

given an infinite point $\mathbf{y}_i$ and a camera position $\mathbf{q}^{WC}$ it is computed its image. It the composition of expressions (1,2,3,5).

Equation (8), can be inverted giving an analytic expression for $\mathbf{y}_i$

$$\mathbf{y}_i = \mathbf{y}\left(\mathbf{z}_i, \mathbf{q}^{WC}\right) \quad (9)$$

from a single infinite point observation $\mathbf{z}_i$ and the camera orientation $\mathbf{q}^{WC}$.

Despite the accurate calibration, because of the radial distortion, points close to the center are more accurate than points close to the image border. Because of that we have applied a heuristic covariance assignment for image points that increases the standard deviation linearly with the radius with respect to the image center:

$$\sigma_h = \sigma_{h_0}\left(1 + \beta\frac{r_d}{r_{d_{\max}}}\right) \quad r_{d_{\max}} = \sqrt{u_0^2 + v_0^2} \quad (10)$$

r_d is computed using (4). In our experiments, $\sigma_{h_0} = 2$pixel and $\beta = 1$. And for every point, the corresponding measurement noise matrix:

$$\mathbf{R}_i = \begin{pmatrix} \sigma_h^2 & 0 \\ 0 & \sigma_h^2 \end{pmatrix}$$

Undistortion jacobian, $\frac{\partial \mathbf{h}_u}{\partial \mathbf{h}_d}$ has analytical expression:

$$\left(\begin{array}{c|c} \begin{array}{c}\left(1 + \kappa_1 r_d^2 + \kappa_2 r_d^4\right) + \\ 2\left(\left(u_d - u_0\right) d_x\right)^2 \times \\ \left(\kappa_i + 2\kappa_2 r_d^2\right)\end{array} & \begin{array}{c}2 d_y^2\left(u_d - u_0\right)\left(v_d - v_0\right) \times \\ \left(\kappa_1 + 2\kappa_2 r_d^2\right)\end{array} \\ \hline \begin{array}{c}2 d_x^2\left(v_d - v_0\right)\left(u_d - u_0\right) \times \\ \left(\kappa_1 + 2\kappa_2 r_d^2\right)\end{array} & \begin{array}{c}\left(1 + \kappa_1 r_d^2 + \kappa_2 r_d^4\right) + \\ 2\left(\left(v_d - v_0\right) d_y\right)^2 \times \\ \left(\kappa_i + 2\kappa_2 r_d^2\right)\end{array} \end{array}\right) \quad (11)$$

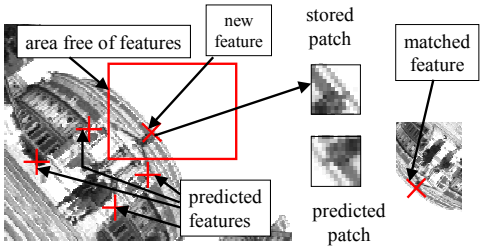

Fig. 2. New features initialization and patch prediction. Right image shows acceptance region and the matched measurement. The acceptance region has been magnified for displaying

The jacobian for the distortion is computed by inverting expression (11).

V. Kalman Filter Sequential Processing

The classical prediction-match-update EKF is applied [16] to deal with geometrical information producing an elliptical acceptance region where the measurement should be. The usage of the photometrical data plays key role to determine where inside the region the actual matching measurement.

All the estimated variables: the camera state $\mathbf{x}_v$ and all the estimated features $\mathbf{y}_i$ $i = 1 \ldots n$ are considered in a joint gaussian state: $\mathbf{x} = \begin{pmatrix} \mathbf{x}_v & \mathbf{y}_1 & \ldots & \mathbf{y}_n \end{pmatrix}^\top$ vector with its corresponding covariance $\mathbf{P}$.

Next the EKF steps are detailed:

a) Prediction:

$$\hat{\mathbf{x}}_{k+1|k} = \begin{pmatrix} \mathbf{f}_v\left(\hat{\mathbf{x}}_{v,k|k}, 0\right) \\ \hat{\mathbf{y}}_{1,k|k} \\ \ldots \end{pmatrix}$$

$$\hat{\mathbf{P}}_{k+1|k} = \mathbf{F}\hat{\mathbf{P}}_{k+1|k}\mathbf{F}^\top + \mathbf{G}\begin{pmatrix} (\Delta t)^2 \mathbf{P}_\alpha & 0 \\ 0 & 0 \end{pmatrix}\mathbf{G}^\top$$

$$\mathbf{F} = \operatorname{diag}\left(\frac{\partial \mathbf{f}_v}{\partial \mathbf{x}_v}, \mathbf{I}\right) \quad \mathbf{G} = \operatorname{diag}\left(\frac{\partial \mathbf{f}_v}{\partial \mathbf{n}}, 0\right)$$

And hence the predicted features in the image:

$$\hat{\mathbf{h}}_{i,k+1|k} = \mathbf{h}\left(\hat{\mathbf{y}}_{i,k+1|k}, \hat{\mathbf{x}}_{v,k+1|k}\right)$$

each predicted feature, has associated a two linearized measurement equations, defined by the two rows matrix, composed of a $0_{2\times 2}$ block except for the considered feature:

$$\mathbf{H}_i = \left(\frac{\partial \mathbf{h}}{\partial \mathbf{x}_v}, 0^1_{2\times 2}, \ldots, 0^{i-1}_{2\times 2}, \frac{\partial \mathbf{h}}{\partial \mathbf{y}_i}, 0^{i+1}_{2\times 2}, \ldots, 0^n_{2\times 2}\right)$$

All the $\mathbf{H}_i$ are piled up vertically to produce the full linearized measurement equation $\mathbf{H}$, and hence the innovation covariance is derived:

$$\mathbf{S}_{k+1|k} = \mathbf{H}\mathbf{P}_{k+1|k}\mathbf{H}^\top + \operatorname{diag}\left(\mathbf{R}_1, \ldots, \mathbf{R}_n\right)$$

b) Match: Every prediction $\hat{\mathbf{h}}_{i,k+1,k}$ and its corresponding $\mathbf{S}_{i,k+1|k}$ 2×2 diagonal submatrix matrix extracted from $\mathbf{S}_{k+1|k}$ define an elliptical acceptance region in the image where the image of the predicted feature should be:

$$(u, v)\,\mathbf{S}^{-1}_{i,k+1|k}\begin{pmatrix} u \\ v \end{pmatrix} \leq \chi^2_{0.05,2}$$

For every map feature, when it was imaged for the first time, we have stored both a texture patch and the camera orientation, then at the prediction stage, the image of the predicted patch is synthesized being quite effective for dealing with distortion, and cyclotorsion. Figure2 shows an example of the stored and predicted patches.

A normalized correlation score is computed for every pixel in that region, the highest score, over a threshold, is considered to be matching measurement $\mathbf{z}_{i,k}$. See figures fig-prediction 3 and 5 for examples of this regions. It has to be noted that we only use a feature detector at the initialization stage (see Sec. VII), at the matching stage no feature detection is performed in the acceptance region and the predicted patch correlation score is computed with respect to all the pixels in the region.

c) Update: Using only the submatrices corresponding with actually matched features, $\mathbf{H}_m$,$\mathbf{R}_m$, $\hat{\mathbf{h}}_m$ and $\mathbf{z}_m$ are built and hence the state estimate is updated:

$$\begin{aligned}\hat{\mathbf{x}}_{k+1|k+1} &= \hat{\mathbf{x}}_{k+1|k} + \mathbf{K}\left(\mathbf{z}_m - \hat{\mathbf{h}}_m\right)\\ \mathbf{K} &= \hat{\mathbf{P}}_{k+1|k}\mathbf{H}_m^\top\left(\mathbf{H}_m\hat{\mathbf{P}}_{k+1|k}\mathbf{H}_m^\top + \mathbf{R}_m\right)^{-1}\\ \hat{\mathbf{P}}_{k+1|k+1} &= (I - \mathbf{K}\mathbf{H}_m)\,\hat{\mathbf{P}}_{k+1|k}\end{aligned}$$

Finally the camera orientation quaternion is normalized. The $\hat{\mathbf{P}}_{k+1|k+1}$, has to be affected by the corresponding normalization jacobian.

VI. State Initialization

As an infinite point can be initialized from a single image (see eq. 9), the camera initial state is null rotation and null angular velocity:

$$\mathbf{x}_{v,0|0} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}^\top \quad (12)$$

$$\mathbf{P}_{0|0} = \text{diag}\begin{pmatrix} 0 & 0 & 0 & 0 & \sigma_\Omega^2 & \sigma_\Omega^2 & \sigma_\Omega^2 \end{pmatrix} \quad (13)$$

The camera orientation for the first frame is considered as the reference frame, hence the null uncertainty in orientation. For the angular velocity, a high value is assigned to σ_Ω, in our case $\sqrt{(2)}\ \frac{\text{rad}}{\text{sec}}$, in order to deal with an initial unknown velocity. This is a remarkable system characeristic that allows to initialize the map from any frame, at any initial angular velocity. In fact in the experiments, the initial angular velocity is not null.

Once the initial value for the camera vector has been defined, about 10 well spread over the image Harris [17] points are detected and the corresponding scene features initialized, expanding the state vector as defined in the next section.

VII. Feature Initialization and Deletion

Once a new image is processed, if the number of features predicted to be visible inside the image goes below a threshold, in our case 14, a new feature is initialized. An area without features is searched randomly in the image, if found the most salient interest point in that area, $\mathbf{z}_j$, is located using Harris point detector. Figure 2 shows an initialization example.

For every new measurement the corresponding feature estimate is computed using equation (9) and the state vector is expanded with the new feature estimate $\hat{\mathbf{y}_j}$. The covariance matrix has to be expanded as well:

$$\mathbf{P}_{k|k}^{new} = \mathbf{J}\begin{pmatrix}\mathbf{P}_{k|k} & 0\\ 0 & \mathbf{R}_j\end{pmatrix}\mathbf{J}^\top, \mathbf{J} = \begin{pmatrix} I & 0\\ \mathbf{J}_1 & \end{pmatrix}, \mathbf{J}_1 = \begin{pmatrix}\frac{\partial \mathbf{h}^{-1}}{\partial \mathbf{x}_v}, 0, \ldots, \frac{\partial \mathbf{h}^{-1}}{\partial \mathbf{z}_i}\end{pmatrix}$$

Features that are predicted to be in the image but not matched are deleted based on the accumulated ratio between times visible and times effectively matched. This simple mechanism allows us to delete non trackable features for example those over non rigid objects (see fig 4 (a)). The deletion method allows us to remove non persistent static scene elements if a scene is continuously revisisited.

VIII. Experimental Results

The experiments, currently programmed in Matlab, are directed to test the feasibility of the system for real time performance. In particular we are focusing the 360° loop closing because this will allow us to completely map a scene. In fact, between submission and acceptance of the paper, we have a C++ version successfully running at 30 fps for maps up to 100 features

Two sequences of an outdoor scene have been acquired with a rotating camera. The processed images were 90° field of view, 320×240 B&W, acquired at 30 fps. with a low cost Unibrain IEEE1394 camera.

The sequences are challenging because only the camera translation were tightly controlled; there were pedestrians walking around and cars moving along a road; the camera automatic control exposition introduces a great deal of change in the image contrast and brightness in response to the outdoors natural illumination conditions.

The fist sequence is on a tripod 360° pan sequence. Being the camera on at tripod the translation is almost assured to be about 1-2cm. and the camera motion is approximately repetitive. Figure 3 shows frames with the acceptance region for every predicted feature and the matched observation. The selected frames are the beginning (frame 14) and the loop closing (frame 286). It is seen how among the first 3 revisited features, 2 were not matched and the third was detected very close to the limit; however in next frame and all the following (frame 319) the reobserved features are regarded as such. It should be noticed that the loop is closed with the normal sequential prediction-match-update, without being necessary any additional computation.

Figure 4 (a) displays a magnified view of two intermediate steps where it is seen how the system rejects the matches for features being initialized on a moving pedestrian The predicted texture patch for the feature cannot be matched inside the acceptance region because the pedestrian motion is not coherent with the rotating camera observing a rigid scene model.

The camera trajectory and all the map features have been plotted over the unit sphere (see fig. 4 (b).) Uncertainties ($\alpha = 95\%$) for each scene point has been plotted as well.

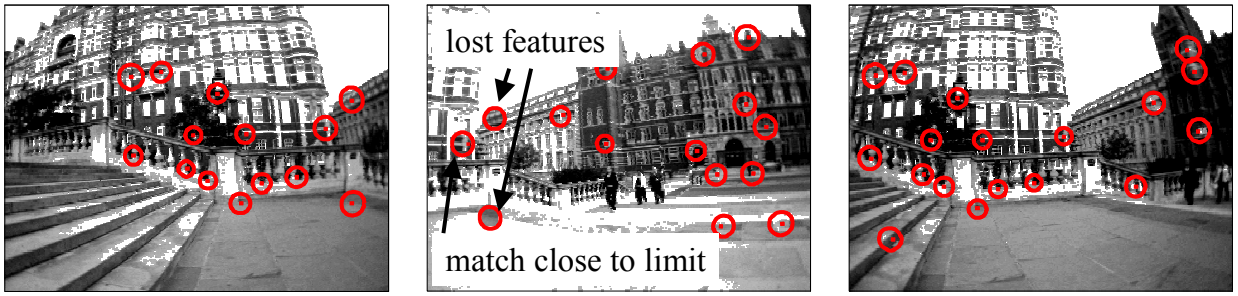

Fig. 3. Loop closing. From left to right frames 14, 286 and 319. It is shown how two features are lost, and a feature has been matched close to the limit of the acceptance region. The rest of the map features has been correctly matched, can be verified comparing frames 14 and 319.

The sphere surface has been magnified around the estimated camera trajectory to illustrate the rotation accuracy; the camera was on a tripod so the first and the second lap should be close.

The second sequence (see Fig. 5) is more challenging because the camera is hand-held, and the trajectory includes both a 360° pan and 360° cyclortorsion; the outdoor natural illumination was challenging as well. The systems also performed successfully. From top left to right bottom, several sequence frames, with the predicted acceptance regions are shown. Third image in the second shows that loop is closed without loosing any of the reobserved features. The camera trajectory over the unit sphere is displayed as well.

IX. Conclusion

A SLAM visual compass has been build and tested off line on outdoor sequences gathered with a hand-held and on a tripod camera. The sequences were challenging because of the natural illumination and the moving pedestrians. The ability to close loops reidentifying most of the persistent map features *inside* of the acceptance 95% region is a clear signal of the correctness of the gaussian stochastic model for coding the geometry of the scene.

Because of the number of features, about a hundred, and the sequential processing this system can, and has been, implemented in real time hence being effectively a real time visual compass.

Our proposal has strong links with the existent real time system proposed in [1] that only deals with features close to the camera. Besides our proposal shows that distant points can provide a robust and accurate information about orientation, specially outdoors. This extend the applicability of SLAM real time computer vision combining both points close to the camera and points at infinity.

Beside the direct application in mobile robotics, this work shows the feasibility of SLAM to process image sequences and its ability to detect a map composed of a reduced set of trackable features, selected for being detectable in the frames where they should visible; this turns out to be a reduced set of key features detected also at loop closing. This loop closing matches provide a rich information not only for real time processing but also to for batch non-linear optimization methods. In particular this system could be well used for build image mosaics for sequences both real time or off line.

Acknowledgment

Project founded by the Spanish CICYT DPI2003-07986, Spanish Ministerio de Educacion y Ciencia PR2005-0332, EPSRC grant GR/T24685 and an EPSRC Advanced Research Fellowship to AJD.

We are very grateful to David Murray and other members of Oxford's Active Vision Laboratory for discussions and software collaboration.

References

[1] A. Davison, "Real-time simultaneous localization and mapping with a single camera," in *ICCV*, 2003.

[2] R. C. Smith and P. Cheeseman, "On the representation and estimation of spatial uncertainty," *Int. J. Robotics Research*, vol. 5, no. 4, pp. 56–68, 1986.

[3] J. Castellanos and J. Tardós, *Mobile Robot Localization and Map Building: A Multisensor Fusion Approach*. Boston. USA: Kluwer Academic Publishers, 1999.

[4] J. Castellanos, J. Montiel, J. Neira, and J. Tardós, "Sensor influence in the performance of simultaneous mobile robot localization and map building," in *Experimental Robotics VI. Lecture Notes in Control and Information Sciences. Vol 250*, P. Corke and J. Trevelyan, Eds. Springer-Verlag, 1994, pp. 287 – 296.

[5] H. Feder, J. Leonard, and C. Smith, "Adaptive mobile robot navigation and mapping," *Int. Journal of Robotics Research*, vol. 18, no. 7, pp. 650–668, 1999.

[6] D. Ortín, J. M. M. Montiel, and A. Zisserman, "Automated multisensor polyhedral model acquisition," in *ICRA*, 2003.

[7] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, ISBN: 0521540518, 2004.

[8] D. Nister, O. Naroditsky, and J. Bergen, "Visual odometry," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2004, pp. 652–659.

[9] L. de Agapito, E. Hayman, and I. A. Reid, "Self-calibration of rotating and zooming cameras," *Int. J. of Computer Vision*, vol. 45, no. 2, pp. 107–127, Nov 2001.

[10] D. Brown, M.and Lowe, "Recognising panoramas," in *International Conference on Computer Vision*, Nice, 2003, pp. 1218–1225.

[11] D. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[12] J. Kosecka and W. Zhang, "Video compass," in *European Conference on Computer Vision*, June 2002, pp. 476–491.

[13] J. M. M. Montiel and A. Zisserman, "Automated architectural acquisition from a camera undergoing planar motion," in *International Symposium on Virtual and Augmented Architecture*, 2001.

[14] E. Mikhail, J. Bethel, and M. J.C., *Introduction to Modern Photogrammetry*. John Wiley & Sons, 2001.

[15] A. Davison, Y. G. Cid, and N. Kita, "Real-time 3D SLAM with wide-angle vision," in *Proc. IFAC Symposium on Intelligent Autonomous Vehicles, Lisbon*, Jul 2004.

[16] Y. Bar-Shalom and T. E. Fortmann, *Tracking and Data Association*, ser. Mathematics in Science and Engineering. San Diego: Academic Press, INC., 1988, vol. 179.

[17] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proceedings of the 4th Alvey Vision Conference*, 1988, pp. 147–151.

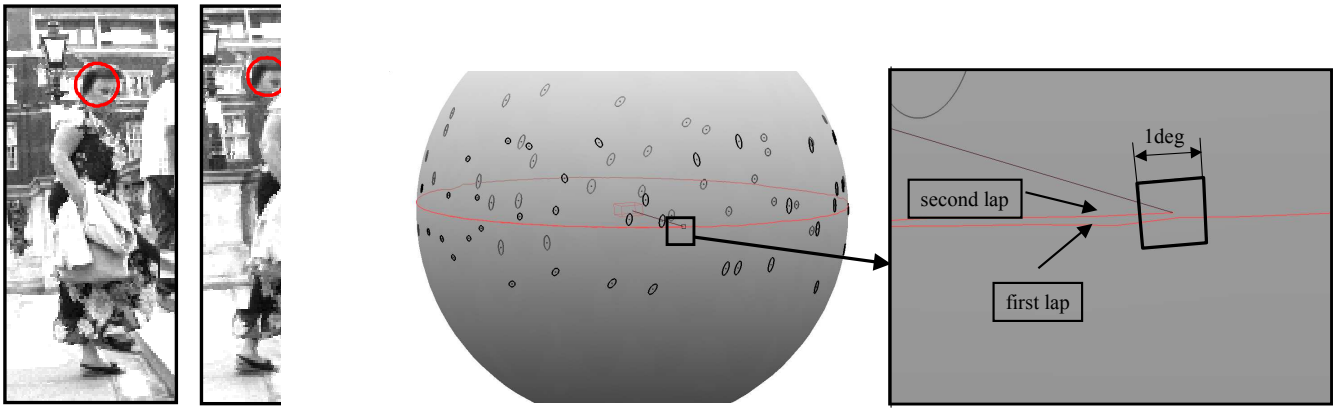

Fig. 4. (a) Shows how features initialized over mobile objects are rejected. Right image shows the predicted feature, that clearly cannot be matched in the acceptance region. (b) Shows a general view of the unit sphere with the extracted features and its corresponding covariances represented; the full trajectory is plotted over the sphere. It is shown as well a magnified view of the final part of the camera trajectory over the sphere; given the sequence was taken with tripod, the first and the second lap should be very close.

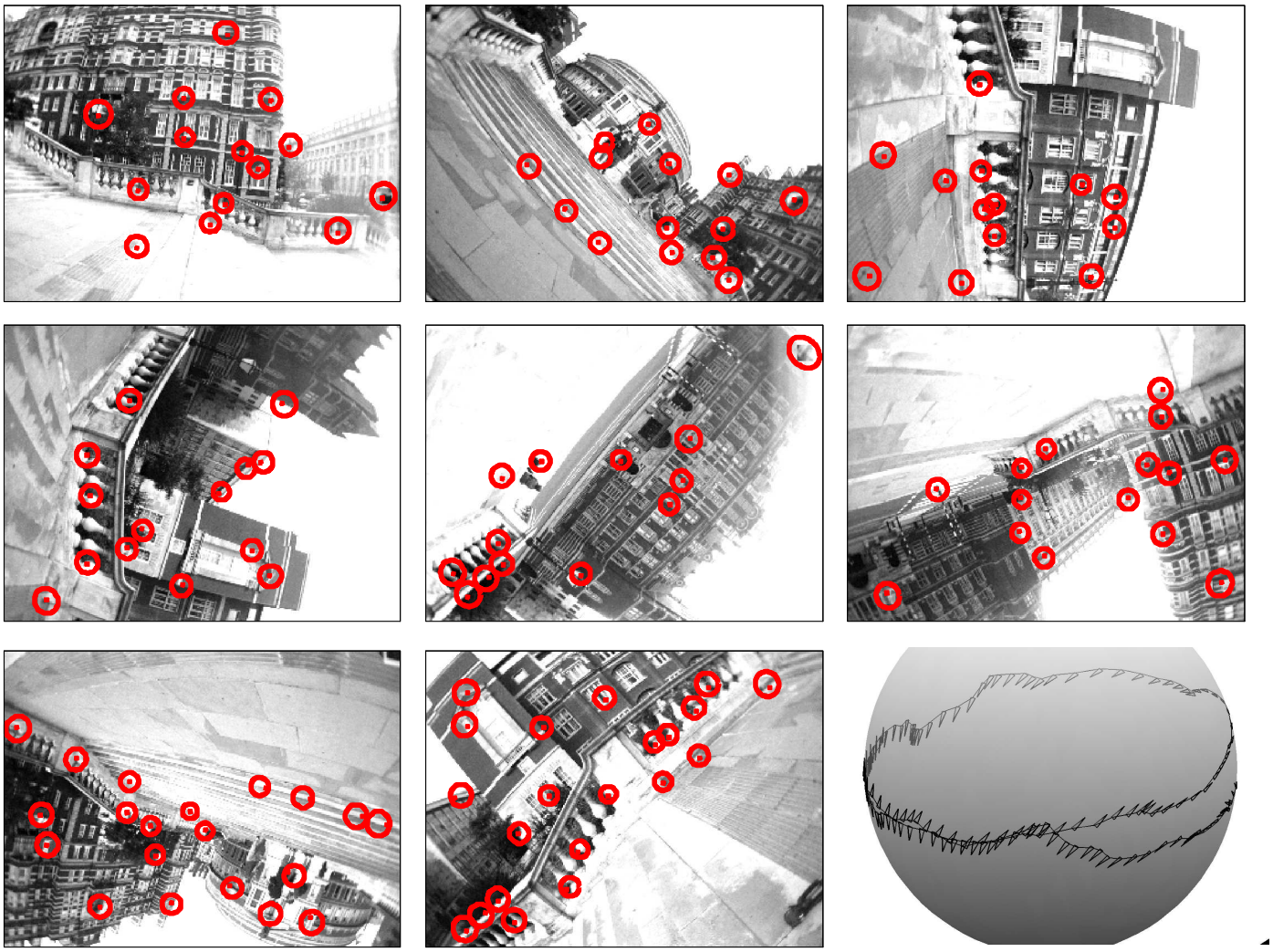

Fig. 5. Hand-held camera sequence with 360 pan and cyclotorsion. Images ordered from top left to bottom right. First image is at the beginning of the sequence, second row last image shows the loop closing; all the features over the walls and the banister are successfully reobserved. The camera trajectory has been plot over the unit sphere; trajectory points are triangles to display the camera cyclotorsion.