2D Simultaneous Localization And Mapping for Micro Air Vehicles

Adrien Angeli, David Filliat, Stéphane Doncieux, Jean-Arcady Meyer

2D Simultaneous Localization And Mapping for Micro Air Vehicles

Adrien Angeli[1] David Filliat[2] Stéphane Doncieux[3] Jean-Arcady Meyer[3]

[1]DGA/CEP-Arcueil/GIP,
16 bis, avenue prieur de la côte,
Arcueil, 94110 France
[2]ENSTA,
32, boulevard Victor,
Paris, 75015 France
[3]Université Pierre et Marie Curie - Paris 6,
UMR 7606 LIP6, 8, rue du Capitaine Scott,
Paris, 75015 France
http://animatlab.lip6.fr
{Adrien.Angeli,Stephane.Doncieux,Jean-Arcady.Meyer}@lip6.fr
{david.filliat}@ensta.fr

Abstract

The purpose of the work reported here is to design a purely vision-based Simultaneous Localization And Map-building system for a MAV flying at a constant altitude. We demonstrate the effectiveness of our approach with videos taken from a heading-down camera mounted on different MAV. In particular, results on loop-closure detection and on map-precision improvement through an on-line estimate of the camera's radial distortion coefficient are presented.

1 Introduction

Over the last decade, MAV study has become a wide research area involving many different disciplines such as control, structural design, electronics, computer sciences or robotics. In particular, the last one also imported skills from the artificial intelligence community, offering opportunities to shift from human-controlled aircrafts to partially-autonomous flying agents. Today, one of the main goals of MAV navigation is to achieve totally autonomous flight, a challenge that implies the ability for the aircraft to self-localize with no *a priori* map of its environment. Currently, in the MAV community, localization is performed using GPS. However, because our research efforts come under the Robur project ([Doncieux et al., 2004], [Doncieux et al., 2006]) that aims at building an autonomous aircraft by drawing inspiration from biology, we prefer to rely on onboard sensors only, such as vision. Consequently, a map has to be built from scratch and simultaneously serve to estimate the MAV's position, leading to the so-called Simultaneous Localization and Map-building (SLAM) problem ([Filliat and Meyer, 2003], [Meyer and Filliat, 2003]). Since 1987 [Smith et al., 1987], several probabilistic frameworks have been developed to solve this problem. In particular, Kalman and particle filters, as well as the *Expectation Maximisation (EM)* algorithm [Thrun, 2002] are widely used. Basically, all these approaches provide different implementations of the same solution, relying on the Bayes rule as a common mathematical background. Based on those frameworks, SLAM has been partially achieved on ground mobile robots (e.g. [Dissanayake et al., 2001], [Thrun et al., 2004]), usually calling upon precise range sensors such as lasers, sonars or radars, and with the help of Kalman or particle filters to mix sensor information with robot odometry. However, the use of such sensors with a MAV is currently impossible because of their large size, heavy weight and high energy consumption. Instead, vision seems to be a good alternative: it is cheap, easy to manage, and offers good opportunities for characterizing the objects recorded in the map.

In this paper, we describe a SLAM method, using a Kalman filter in the case of 2D MAV navigation, which allows to simultaneously build a metric map of visual ground landmarks, and to perform accurate aircraft localization in this map. Indeed, in

the context of the Robur project, 2D-localization is a suitable first step in the perspective of implementing more cognitive behaviors like soaring for example, because the search of thermals or slope-winds can be done at a constant altitude. In our visual odometry approach, the 2D displacement of the aircraft between two adjacent instants in time is estimated using feature-matching between the corresponding images. To this end, we propose an image feature detector that associates the Harris corner detector [Harris and Stephens, 1988] with the SIFT descriptor [Lowe, 2004]. Additionally, in order to improve the map precision, we estimate the radial distortion coefficient of the camera on-line, as an additional parameter in the Kalman filter. Our main results concern the system's loop-closure capacities ([Newman et al., 2006]), i.e., the possibility of recognizing previously detected landmarks, and the reduction of the uncertainties in both the aircraft and the landmarks' positions.

This paper is organised as follows: section 2 summarizes related work in vision-based SLAM, section 3 describes our own approach, section 4 summarizes the results obtained, which are discussed in section 5.

2 Related work

SLAM techniques are often used with ground mobile robots, calling upon lasers, radars or sonars. Data issued from sensor measurements are mixed with wheel-encoded odometry using probabilistic estimators like Kalman or particle filters ([Dissanayake et al., 2001], [Thrun et al., 2004]). Recently, several authors (for example [Barfoot, 2005] or [Jeong and Mu Lee, 2005]) incorporated vision in that traditional scheme, replacing the range sensors by a camera. Also, the authors of [Kim and Sukkarieh, 2003] performed SLAM on an aircraft using a heading-down camera and control inputs sent to the aircraft instead of odometry. However, all these approaches are not totally vision-based since they still rely on "mechanical" odometry or control inputs. To achieve exclusively vision-based SLAM, two different approaches are possible: the first one, called *structure from motion (SfM)*, comes from the computer-vision community and does not need any kind of odometry, while the second one, let us call it *visual odometry SLAM (voSLAM)*, relies on the implementation of a visual odometry scheme.

In the SfM approach ([Chiuso et al., 2002], [Davison, 2003]), the 3D-displacement of the camera is estimated using feature-tracking in images grabbed at consecutive instants in time. It calls upon the association of the *eight points algorithm* [Longuet-Higgins, 1981] with a Kalman filter used to simultaneously estimate the position of the camera and the structure of the scene. The pose of the camera is computed as a combination of 3D rotation and translation relative to the original position. The structure of the scene is a map containing points (or features) extracted from the images, whose 3D-coordinates are triangulated once the camera's position has been estimated. The method used for feature detection is generally a simple corner detector, and matching is usually done by the minimization of a *Sum of Squared Difference (SSD)* over patches taken around the points in the image. The KLT algorithm [Shi and Tomasi, 1994] is an efficient implementation of such a feature detector. The SfM approach does not need any kind of odometry, which makes it independent from the dynamic model of the camera motion: the frame rate is considered high enough (25 to 30Hz) to only concern very small displacements between two adjacent instants in time. Then, when the camera has moved, since its new pose is considered very close to the previously computed one, some uncertainty is simply added to this last one, without modifying it. After that, matching between the actually observed points and those recorded in the map correct the predicted pose, as well as the previously computed feature coordinates.

In the voSLAM approach ([Jung, 2004], [Lemaire et al., 2006] and [Saeedi et al., 2006]), a model of the MAV motion is needed and point matching between two adjacent images is used to compute the corresponding visual odometry. Several optimization algorithms, like the *Least Square Estimation (LSE)*, can be used to estimate the combination of rotation and translation that corresponds to the motion separating the two consecutive images. The depth-estimate of the points recorded in the map can be done separately or simply abandoned. In the former case, a stereo-vision system can be used to compute the 3D local coordinates of the features. In the latter case, localization and mapping are performed in 2D, with a single camera moving on a plane perpendicular to its optical axis. In both cases, when the camera has moved, visual odometry is used to predict its new pose, using the same correction procedure as the SfM one. As the visual odometry may be difficult to estimate reliably for some particular displacements of the camera, as stated in [Saeedi et al., 2006], it may be useful to allow only certain types of movements for the aircraft, so as to obtain a precise estimation the motion, assuming for example that rotations are small in comparison with translations.

Finally, the main advantage of the SfM approach is its ability to handle almost any type of 3D camera

motion (independently of any precise dynamic model) and to incorporate the depth-estimate of the features, provided that the differences between consecutive images are small, which implies a high frame rate. Instead, voSLAM does not need high frame rates and visual odometry can be a good alternative to traditional wheel-encoded odometry in the case of MAV navigation, providing a good estimate of the predicted position. Moreover, when the dynamic model is well known and finely tuned, the associated uncertainty can be managed more precisely, in order to fit the real kinematic model.

Nevertheless, for SfM or voSLAM methods to perform well, the *data association problem* ([Neira and Tardós, 2001]), i.e., the problem of correctly matching individual visual features, needs to be solved. In the SfM scheme, matching is done by simple keypoint tracking over several consecutive images, without the addition of any extra-information to the landmarks of the map. Loop-closure detection is thereby very difficult in this case, implying very precise position estimates, with near-zero drift over time, to make it possible. On the contrary, the use of the SIFT descriptor (detailed in [Lowe, 2004]), as in [Barfoot, 2005] and [Saeedi et al., 2006] permits efficient large baseline matching, i.e. the match of visual features when differences between images are large, and enhances loop-closure detection.

In our case, we have a precise knowledge of the MAV's dynamic model and we want to be able to detect loop-closure, so as to decrease the aircraft uncertainty and to obtain a reliable trajectory estimate. For these reasons, the voSLAM solution to the SLAM problem seems to be a better alternative than the SfM approach and has been applied here.

3 System description

The experimental setup leading to the results described herein consists in a tiny wireless grayscale camera, mounted downward on a MAV flying at a constant altitude and over a flat terrain. Both a blimp (see figure 1) or the Twinstar MAV of the Paparazzi team[1] (see figure 2) were used, the corresponding images being sent to a ground receptor connected to a PC that performed the required computations.

Our system is based on visual odometry associated with an *Extended Kalman Filter (EKF)* [Smith et al., 1987], and accordingly pertains to the voSLAM category mentioned earlier. Let $X_t = \left[[X_{aircraft}]^T, [X_{map}]^T\right]^T$ denote the *state vector* of the EKF, recording the positions of the aircraft and visual landmarks' at time t, where:

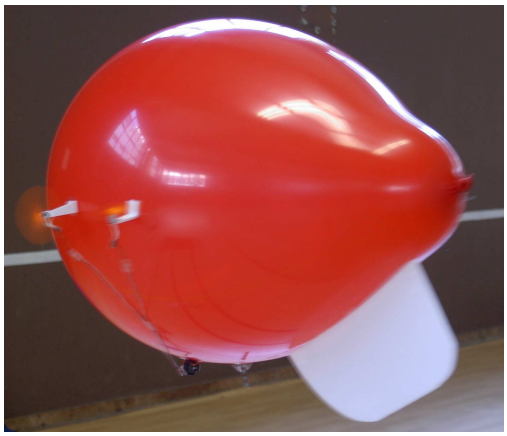

Figure 1: The blimp.

Figure 2: Paparazzi Twinstar.

- $X_{aircraft} = [x_a, y_a, \phi_a]^T$ contains the 2D coordinates and orientation of the aircraft
- $X_{map} = [x_1, y_1, \ldots, x_i, y_i, \ldots, x_n, y_n]^T$ contains the 2D coordinates of n punctual landmarks in the map

All the coordinates contained in the state vector are given in pixels and correspond to positions in the space of the images. Furthermore, since the camera is heading-down, its position (which is considered to be the MAV's position) corresponds to the centre of the image. Then, the SLAM problem consists in recovering the state X_t from observations Z^t and visual odometry U^t up to time t, in a way that maximizes the posterior distribution over the map and the aircraft's pose $p(X_t|Z^t, U^t)$. The evolution of X_t over time is modelled by a function f of the *prediction model*:

[1] www.nongnu.org/paparazzi

$$X_t = f(X_{t-1}, U_t) + q \quad (1)$$

Under a static world assumption, the function f does not affect X_{map}. Only $X_{aircraft}$ is modified according to the displacement estimated by the visual odometry U_t. q represents the time-independent zero-mean gaussian noise associated with the displacement of the MAV. Also, a function h of the *observation model* transforms the global coordinates of the landmarks stored in the map into local coordinates corresponding to the current image:

$$Z_t = h(X_t) + r \quad (2)$$

r represents the time-independent zero-mean gaussian noise associated with the feature extraction in the images.

A covariance matrix P_t associated with the state vector contains the uncertainties corresponding to the quantities stored in X_t. In the EKF scheme, X_t and P_t are estimated in a recursive two-step *prediction-update* procedure:

1. Prediction step

$$\begin{aligned} X_t^* &= f(X_{t-1}, U_t) \\ P_t^* &= F_t P_{t-1} F_t^T + Q \\ Z_t^* &= h(X_t^*) \end{aligned}$$

2. Update step

$$\begin{aligned} K_t &= P_t^* H_t^T (H_t P_t^* H_t^T + R)^{-1} \\ X_t &= X_t^* + K_t (Z_t - Z_t^*) \\ P_t &= P_t^* - K_t H_t P_t^* \end{aligned}$$

F_t (respectively H_t) is the Jacobian matrix of f (respectively h). Q (respectively R) is the noise covariance matrix associated with q (respectively r). Intuitively, during the update step, the gain K_t is designed so as to grant more confidence to the information with the lowest uncertainty. For further details on the implementation of the EKF, the interested reader may refer to [Dissanayake et al., 2001].

3.1 The prediction model

The prediction model of the EKF tries to "guess" the actual position of the MAV, taking advantage of the displacement estimated by the visual odometry since the last position estimate. The state prediction calls upon the function f, while P is updated by F according to the dynamic model of the MAV. Visual odometry is obtained by feature-tracking between consecutive images using the KLT algorithm [Shi and Tomasi, 1994]. Applying a LSE over all the tracked features in two adjacent frames, we can compute the 2D combination of rotation and translation representing the MAV's motion. However, for the KLT to perform well, differences between adjacent images have to be small, which implies a sufficient frame rate: 5 images per second were used in our implementation to obtain reliable estimates of 2D displacements.

3.2 The observation model

As previously mentioned, when flying over a previously hovered area, a MAV must be able to match currently seen features with earlier stored landmarks. This "matching event" is managed by the observation model of the EKF and a robust large baseline matching procedure is needed. Therefore, the simple keypoint-tracking proposed by the KLT algorithm cannot be trusted in this case, because it is adapted to small baseline procedures and because the time-lag separating the actual fly-over from the previous one can be very long. This is why we associated a SIFT descriptor to all the features extracted using the KLT algorithm.

The SIFT descriptor [Lowe, 2004] is a vector of size 128 more or less invariant with changes in scale, rotation, translation and illumination, which contains relevant and stable information for large baseline matching. It is very powerful when the differences between two images are important, as it is insensitive to small affine changes in the 3D viewpoint of the scene for example. But, although the vector computation procedure of the SIFT algorithm is fast, the feature detection is time consuming: local minima and maxima are searched in differences of gaussians taken from several sub-sampled versions of the original image. Thus, the association proposed here is faster than the SIFT algorithm alone, and provides more accurate "long-time matchings" than the KLT algorithm alone.

In the observation model, the function h is responsible for the state correction, while P is updated by H according to the sensor model. Since the SIFT descriptors make our observation model robust for large baseline matching, the frequency of the prediction-update process can be lower than the visual odometry rate: the loop of the EKF is then performed 5 times slower than that of visual odometry, thus performing SLAM at 1Hz.

3.3 Radial distortion

The radial distortion coefficient of the camera is responsible for deformations proportional to the dis-

tance to the image centre: the further from the centre, the higher the distortion. This produces rounded images that need to be unwrapped in order to enhance the precision in the position of the landmarks stored in the map. This effect is quite important in the case of the small camera we were using. Then, to retrieve the original "flat" aspect of each image, the value of this coefficient has to be determined. To this end, we add to the Kalman filter a new unknown, C, which corresponds to the radial distortion coefficient. Since feature-matching based on the wrapped image induces a measurement error due to the distortion, we can update the value of C proportionally to the induced error. This implies knowledge of the relation between a pixel's position in the image and the amount of error due to distortion. This relation is given by:

$$\hat{x}_i = \frac{x_i}{\sqrt{1+2Cr^2}} \quad \hat{y}_i = \frac{y_i}{\sqrt{1+2Cr^2}}$$

where $r = \sqrt{x_i^2 + y_i^2}$. $\hat{x}_i$ and $\hat{y}_i$ are the wrapped coordinates of the point i, while x_i and y_i are the unwrapped coordinates that are stored in the map. Then, we can define a function g that unwraps the pixel's position based on this relation. The amount of correction imputable to distortion can be computed as the Jacobian matrix G of g. The previous observation function h must be updated to take advantage of g. We then define a new observation function $w = g \circ h$ and the corresponding Jacobian $W = G \times H$. Thus, C can be initialised to 0 before being incrementally estimated, until it converges to the correct value.

3.4 Map management

The $o(N^2)$ complexity of the Kalman filter (N being the dimension of the state vector) does not allow large environments to be mapped, limiting the total number of landmarks that can be stored in the map. Beyond this upper limit, real-time processing is no longer possible. To prevent the state vector from a too rapid growth that would dramatically limit the mapping capacity of our SLAM system, we introduced a landmark notation that makes the substitution of some elements of the state vector possible: each time a feature is recognized, its "recognition-score" is increased. Then, when including new landmarks to the map, we do not add all of them at the end of the current state vector. Some new features will replace older ones whose recognition-score are small, implying that they are not suitable for robust matching. Thereby, we empirically set the size increase of the map to one landmark every second, while the number of feature-replacements has been set to 4 every second. Such tuning affords a good compromise: in case of too many substitutions, matching may fail (in particular when closing the loop), whereas, in case of too scarce updates, real-time is only possible during a short time-span.

4 Experimental results

In this section we will describe the results obtained using our 2D vision-based SLAM system. A first subsection will present results obtained with the Paparazzi MAV, while a second section will concentrate on loop-closure detection using a blimp. Finally, a third subsection will comment on the effects of some tuning parameters affecting the visual odometry.

4.1 Mapping results

An example of a mosaic automatically generated using the MAV trajectory estimated by the EKF is given in the figure 3. The corresponding map and EKF trajectory, as well as the visual odometry, are shown in the figure 4.

As we can see, the visual odometry is precise in this case and the EKF-filtered trajectory is close to it. Additionally, it appears that the corrections made on the point positions are relatively small, although they increase as time elapses. This is due to the increasing drift in the MAV's position over time, which leads to noisy predictions of landmarks' positions.

4.2 Loop closing

Figure 5 corresponds to data generated by a blimp following a simple loop trajectory. It shows the corresponding map and the estimates of the blimp's position.

As demonstrated by the red trajectory of figure 5, simply integrating visual odometry over time diverges, and loop-closure is not detected: this is caused by error accumulation over the consecutive displacement estimations, giving a more and more noisy approximation of the real position. Instead, the EKF filter (whose trajectory is shown in green) can cope with the noise associated with visual odometry and with the perception of the landmarks so as to decrease the positioning error. Additional evidence of this phenomenon is provided by the evolution of the aircraft's position uncertainty depicted on figure 6.

This evolution may be decomposed in two steps: a first constant growth in a saw-toothed manner, fol-

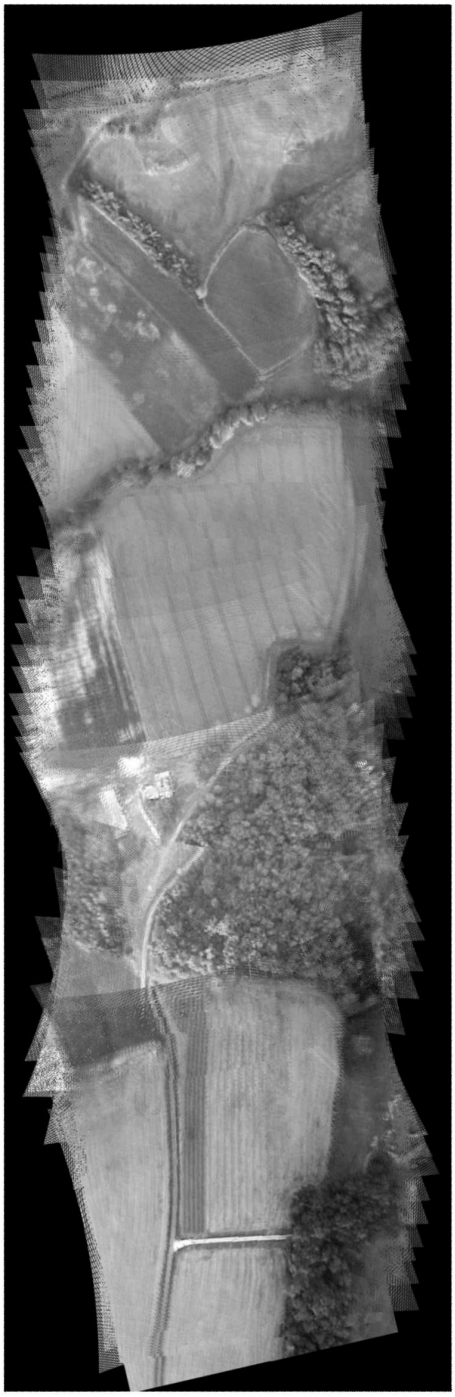

Figure 3: Mosaic obtained using images sent by the Paparazzi MAV.

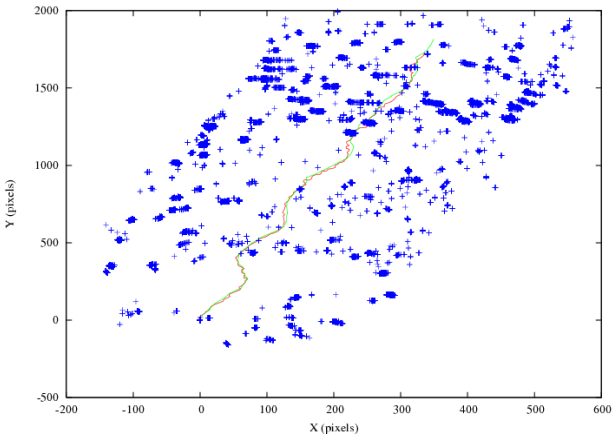

Figure 4: Paparazzi's EKF trajectory (green), visual odometry (red) and map (blue crosses) obtained with a 5Hz visual odometry and a 1Hz SLAM.

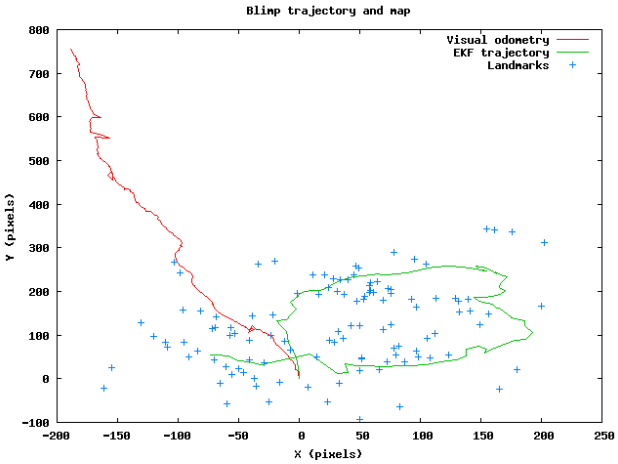

Figure 5: A blimp's loop trajectory generated by a 5Hz visual odometry and a 1Hz SLAM. The trajectory resulting of the integration of visual odometry over time is show in red while the EKF-filtered trajectory is shown in green. The landmarks are represented by blue crosses.

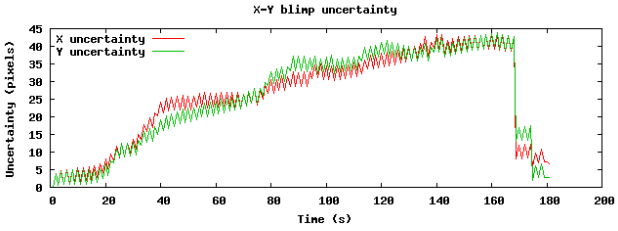

Figure 6: Evolution of the blimp's position uncertainty over time.

lowed by a dramatic drop around $t \simeq 170s$, i.e., when loop closure is detected. The saw-toothed shape of the first phase is due to the predict-update process of the Kalman filter. During the prediction step, a new state is estimated from the previously computed

one by taking the displacements computed from the visual odometry into account. The uncertainty on the position of the aircraft is thus increased according to the error associated with the prediction model. Then, during the update step, matching local features with map landmarks refines the predicted estimate, making it more reliable and decreasing its uncertainty. After the loop closure detection, a series of smaller drops occurs that decrease the uncertainty value down to a level of 5 to 10 pixels.

The steep fall corresponding to loop detection is due to the very small uncertainty associated with old landmarks that are recognised when closing the loop. In the video sequence, when new features are added to the map, they are generally "seen" again in several consecutive images just after having been added. Then, the corresponding uncertainties have often been decreased, converging to a lower limit around 0.7 pixels. This can be seen in figure 7, which represents the evolution of the uncertainty associated with a specific feature over time, this feature being chosen as representative of the issue at odds. Turns out that the corresponding uncertainty rapidly converges to a value between 0.6 and 0.8 pixels at first. Then, since in the video sequence this feature is not seen again until loop-closure around time $t \simeq 170s$, its uncertainty remains constant, before decreasing to a near-zero level when it is detected again during loop-closure.

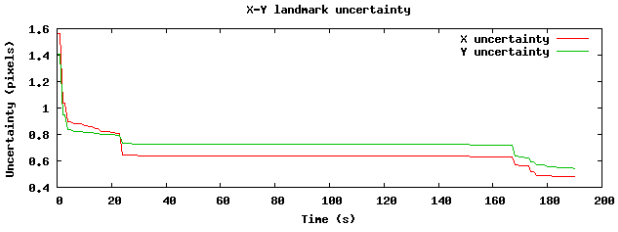

Figure 7: Time evolution of the position uncertainty attached to a specific landmark.

Finally, the figure 8 shows both the evolution of the radial distortion coefficient of the camera and the associated uncertainty. The coefficient converges to a near-zero value, and its initial strong corrections are imputable to its high uncertainty. As time elapses, the estimated value is becoming more and more precise, as shown by the decrease on the uncertainty graph, thus producing smaller corrections.

4.3 Effects of visual odometry

The main parameters we need to adjust for the Kalman filter are the uncertainties associated with

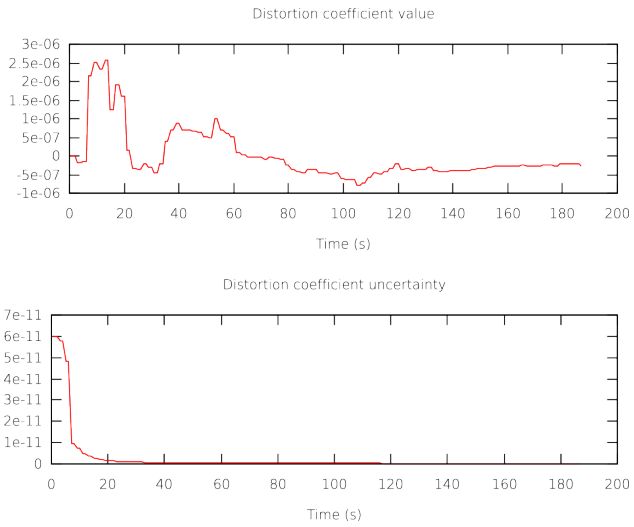

Figure 8: Evolution of the distortion coefficient of the camera together with the associated uncertainty.

each of the models, as well as the frame rates used for the visual odometry and the SLAM predict-update process. As explained in sections 3.1 and 3.2, these frame-rates may be different. For the experiences just described, they were set to 5 images per second for odometry, and to 1Hz for the SLAM loop.

However, we can see from the EKF trajectory of figure 5 that the corrections made by the Kalman filter are sometimes huge, producing an irregular trajectory: this is due to the large differences between images processed by the EKF, causing important updates. This can be dangerous if the noises associated with the two models of the EKF are not precisely evaluated. To obtain smoother trajectory estimates and to avoid drastic corrections, we can increase the frame rates, so as to limit the differences between consecutive images, but with the unfortunate consequence of making real-time computation impossible. For instance, figure 9 shows a blimp's reconstructed trajectory using a 25Hz visual odometry and a 5Hz SLAM.

As we can see, the estimated trajectory is more precise and reliable, and loop-closure is correctly detected. Interestingly, in spite of the higher frequency, we can see that visual odometry still performs poorly. In particular, it is very noisy when rotations are not small in comparison with translations, as already mentioned in [Saeedi et al., 2006]. Such observation can be exploited to simplify the prediction model of the EKF by taking only rotations into account, and simply adding noise to the $X - Y$ position without modifying it. Since the frame rate is high, the observation model should be able to correctly update

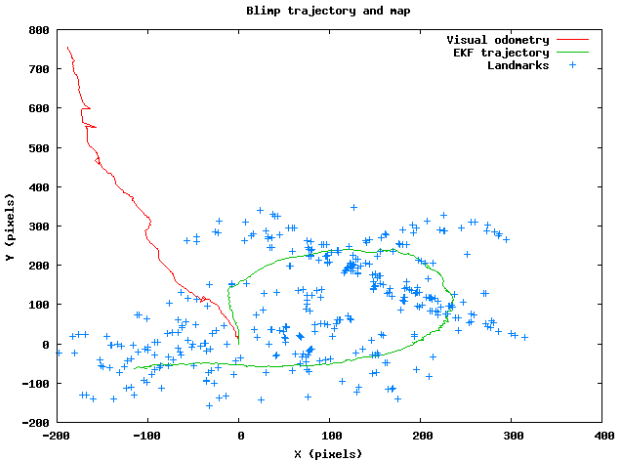

Figure 9: A blimp's loop trajectory obtained with a 25Hz visual odometry and a 5Hz SLAM. The trajectory resulting from the integration of visual odometry over time is shown in red, while the EKF-filtered trajectory is shown in green. The landmarks are represented by blue crosses.

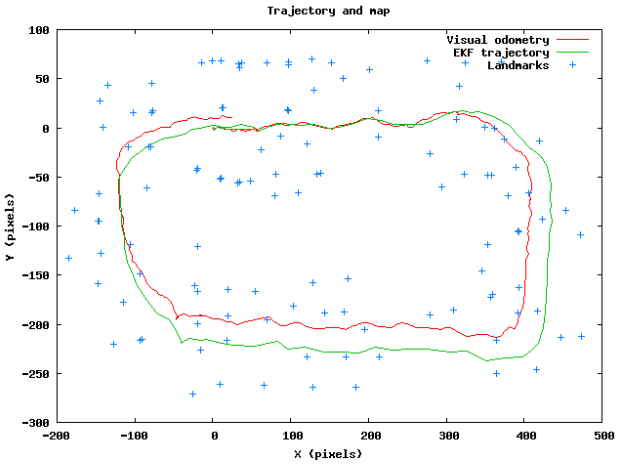

Figure 10: A constant orientation loop trajectory obtained with a 5Hz visual odometry and a 1Hz SLAM. The trajectory resulting of the integration of visual odometry over time is show in red, while the EKF-filtered trajectory is shown in green. The landmarks are represented by blue crosses.

the predicted position. Accordingly, the noise associated with the prediction model can be set to a lower value, thus slowing the saw-toothed growth over time in the position uncertainty. When tuned this way, our SLAM system becomes closer to the SfM approach described in the state of the art than to the voSLAM approach originally chosen.

Nevertheless, when rotations have a low amplitude and when as textured images as possible are used, the visual odometry is very precise. In figure 10 a loop trajectory is shown that was obtained with translations only (the camera orientation is kept nearly constant) and with low frame rates (5Hz for visual odometry and 1Hz for SLAM).

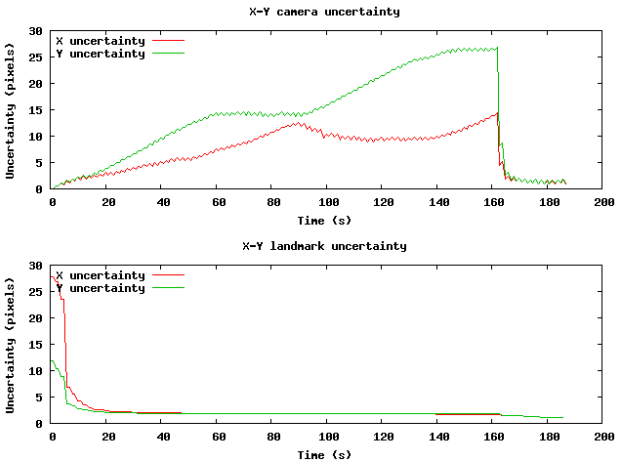

Figure 11: Uncertainty evolution over time for constant orientation loop trajectory obtained with 5Hz for visual odometry and 1Hz for SLAM.

5 Discussion

The results presented in the previous section showed successful loop-closure detection, with an associated uncertainty decrease in both the aircraft and the landmarks' parameters. At the end of a given run, the positions estimated are particularly reliable: the MAV's uncertainties are between 5 and 10 pixels, while the landmarks' uncertainties are around 1 pixel. Such small position error is crucial for further use, like precise navigation or trajectory-planning. The online estimate of the camera's radial distortion turns out to be of particular interest to improve the map's precision, specially in the case of a wide-angle camera, where distortion is very strong near the edges of the image.

However, several features of the proposed system need to be investigated more deeply. First, we did not completely solve the data-association problem mentioned in section 2. We added a precise and robust SIFT descriptor to reinforce the matching conditions, but *false-positive matching* may still trigger the divergence of the Kalman filter. In order to deal with such outliers, we could try to match groups of interest points like in [Jung, 2004] using the *Joint Compati-*

bility Test defined in [Neira and Tardós, 2001]. Instead, we used the Mahalannobis distance to discard dangerous features. Furthermore, we are limited by the $o(N^2)$ complexity of the Kalman filter that does not allow large environments to be mapped. In spite of our map management method using landmark notation, it turns out that our system cannot manage more than 150 features if real-time processing is a mandatory constraint. Possibly, this size limitation issue could be addressed by combining global topological localization and precise metrical mapping, as done in [Bosse et al., 2003] or [Kouzoubov and Austin, 2004]. Finally, in order for the Robur MAV to exhibit other behaviors entailing a 6-degree of freedom navigation, the vision-based SLAM system presented must be extended to cope with a 3D-space.

6 Conclusion

In this paper, we presented a vision-centred 2D-SLAM system that affords reliable estimates of the aircraft position and orientation, as well as a map calling upon landmarks whose coordinates are precisely estimated. We demonstrated loop-closure detection capabilities, entailing a decrease in the uncertainties on the aircraft and landmarks' positions when closing the loop. In the future, contextual information about the landmarks could be recorded in the map, making it possible to encode dangerous, energy sparing or recharging areas. Such information would be of particular interest for trajectory planning.

Acknowledgement

A financial support (BQR grant) from University Pierre and Marie Curie - Paris 6 is gratefully acknowledged.

References

T.D. Barfoot. Online visual motion estimation using fastslam with sift features. In *Proceedings of Intelligent Robots and Systems (IROS)*, August 2005.

M. Bosse, P. Newman, J. Leonard, M. Soika, and W. Feiten. An atlas framework for scalable mapping. In *International Conference on Robotics and Automation*, 2003.

A. Chiuso, P. Favaro, H. Jin, and S. Soatto. Structure from motion causally integrated over time. *IEEE transactions on pattern analysis and machine intelligence*, 24(4):523–535, 2002.

A.J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *Ninth IEEE International Conference on Computer Vision (ICCV'03)*, 2003.

M. W. M. Dissanayake, P. Newman, S. Clark, H. F. Durrant-White, and M. Csorba. A solution to the simultaneous localization and map building (slam) problem. *IEEE Transactions On Robotics and Automation*, 17(3):229–241, 2001.

S. Doncieux, J.B. Mouret, L. Muratet, and J.-A. Meyer. The ROBUR project: towards an autonomous flapping-wing animat. In *Proceedings of the Journes MicroDrones*, Toulouse, 2004.

Stephane Doncieux, Jean-Baptiste Mouret, Adrien Angeli, Renaud Barate, Jean-Arcady Meyer, and Emmanuel Margerie. Building an artificial bird: Goals and accomplishments of the robur project. In *European Micro Aerial Vehicles (EMAV)*, 2006.

D. Filliat and J.-A. Meyer. Map-based navigation in mobile robots - I. a review of localisation strategies. *Journal of Cognitive Systems Research*, 4(4):243–282, 2003.

C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of 4th Alvey Vision Conference*, page 147151, 1988.

W.Y. Jeong and K. Mu Lee. Cv-slam: A new ceiling vision-based slam technique. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3070–3075, 2005.

I.K. Jung. *Simultaneous Localization and Mapping in 3D Environnements with Stereovision*. PhD thesis, Institut National Polytechnique de Toulouse, 2004.

Jong-Hyuk Kim and Salah Sukkarieh. Airborne simultaneous localisation and map building. In *ICRA*, pages 406–411, 2003.

Kirill Kouzoubov and David Austin. Hybrid topologicavmetric approach to slam. In *In proceedings of the 2004 IEEE lnternatlonal Conference on Robotics and Automation*, April 2004.

T. Lemaire, I.K. Jung, and S. Lacroix. Vision-based slam: Stereo and monocular approaches. Technical report, LAAS-CNRS, 2006. submitted to IJCV/IJRR special joint issue.

H.C Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, 1981.

D.G. Lowe. Distinctive image feature from scale-invariant keypoint. *International Journal of Computer Vision*, 60(2):91–110, 2004.

J.-A. Meyer and D. Filliat. Map-based navigation in mobile robots - II. a review of map-learning and path-planing strategies. *Journal of Cognitive Systems Research*, 4(4):283–317, 2003.

J. Neira and J.D. Tardós. Data association in stochastic mapping using the joint compatibility test. *IEEE Transactions on Robotics and Automation*, 17(6):890–897, 2001.

Paul Newman, Dave Cole, and Kin Ho. Outdoor slam using visual appearance and laser ranging. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2006.

P. Saeedi, P.D. Lawrence, and D.G. Lowe. Vision-based 3-d trajectory tracking for unknown environments. *IEEE transactions on robotics*, 22(1): 119–136, February 2006.

J. Shi and C. Tomasi. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR94)*, 1994.

R. Smith, M. Self, and P. Cheeseman. A stochastic map for uncertain spatial relationships. In *Workshop on Spatial Reasoning and Multisensor Fusion*, 1987.

S. Thrun. Robotic mapping: A survey. In , editor, *Exploring Artificial Intelligence in the New Millenium*, number CMU-CS-02-11, chapter Robotic mapping: A Survey, pages –. Morgan Kauffman, February 2002.

S. Thrun, M. Montemerlo, D. Koller, B. Wegbreit, J. Nieto, and E. Nebot. Fastslam: An efficient solution to the simultaneous localization and mapping problem with unknown data association. *Journal of Machine Learning Research*, -:–, 2004.