Using Planar Facets for Stereovision SLAM

Cyrille Berger, Simon Lacroix

Using Planar Facets for Stereovision SLAM

Cyrille Berger and Simon Lacroix

Abstract

In the context of stereovision SLAM, we propose a way to enrich the landmark models. Vision-based SLAM approaches usually rely on interest points associated to a point in the Cartesian space: by adjoining oriented planar patches (if they are present in the environment), we augment the landmark description with an oriented frame. Thanks to this additional information, the robot pose is fully observable with the perception of a single landmark, and the knowledge of the patches orientation helps the matching of landmarks perceived from different viewpoints. The paper depicts the chosen landmark model, the way to extract and match them, and presents some SLAM results obtained with such landmarks.

I. Introduction

Any solution to the problem of simultaneous localisation and mapping (SLAM) needs to develop the following functions:

- Landmarks detection. This mean the identification and extraction from the perceived data of elements in the environment on which the robot relies to estimate its position,
- Relatives measures estimation. Two process are needed:
  - Estimation of the position of the landmarks relatively to the current position of the sensor: this is the *observation* step,
  - Estimation of robot motions between two landmark perceptions: this is the *prediction* step,
- Data associations. Observing landmarks is only useful if they can be perceived from different positions: they need to be robustly *matched* when perceived from different viewpoints.
- Estimation. This the heart of the SLAM: using the various motion predictions and landmark observations, the estimation process computes the position of the robot and of landmarks

Most of the many existing contributions in the literature tackle the estimation process – an up to date state of the art can be read in [4], [1]. Various formalisms have been successfully introduced, and important contributions propose structures of the landmarks maps in order to both reduce the algorithmic complexity of the estimation process, and the difficulties related to the non-linearity of the problem.

Cyrille Berger is supported by Thales Optronics

Cyrille Berger and Simon Lacroix are with University of Toulouse, LAAS/CNRS, 7, Ave du Colonel Roche, F-31077 Toulouse Cedex 4, France, FirstName.Name@laas.fr

But most of the functions needed for SLAM are *perception process*. This is obvious for the detection of landmarks and the observation of their position, which comes from the processing of acquired data. And if the landmark matching problem can be solved by the mere knowledge of their estimated and observed positions, it is more robustly solved by the landmarks *identification and recognition*, because it is independent of the current position estimate.

For the perception processes, the choice of the landmarks model is a critical point. A good landmark must be salient in the data, and should be easy to detected and matched from different viewpoints. The model of a landmark can be split in two parts: one part dedicated to the estimation (geometric variables which defines its position), and one part dedicated to the matching process, which includes the information that identifies it. For instance, most of the solutions to the Vision SLAM problem are based on interest points (Harris points, or "SIFT" points, either in stereovision [8], [15] or monocular vision [3]). Interest points have all the required properties: they correspond to 3D points in the environment, and they carry visual information useful to match them.

But the environment model made of such landmarks is poor, and is only useful to solve the SLAM problem. There is a strong interest to rely on richer landmarks models: on the one hand it can help the matching process, and on the other hand it can yield environment models more representative of the environment structure, on the basis of which other functions than localization can be applied (*e.g.* computation of free space, computation of visibility...). The recent contributions to vision SLAM which use segments as landmarks are going in this direction [5], [17], [9].

In this paper, we propose a landmark model based on planar facets detected using stereovision. Relying on interest points, this model contains six geometric parameters and texture informations: this description gives a better observability of the robot position by the perception of a small number of landmark[1], and makes the matching process easier when detecting landmark from different view points. Section II presents this landmark model and the corresponding detection process in a pair of stereoscopic images. Section III describes tracking and matching algorithms, and SLAM results using those landmarks are shown in section IV.

II. Planar facets

Facets correspond to planar areas detected around interest points, by checking whether an homography between their two stereoscopic views can be fitted or not.

[1] As opposed to [14], in which facets are only used to ease the matching process

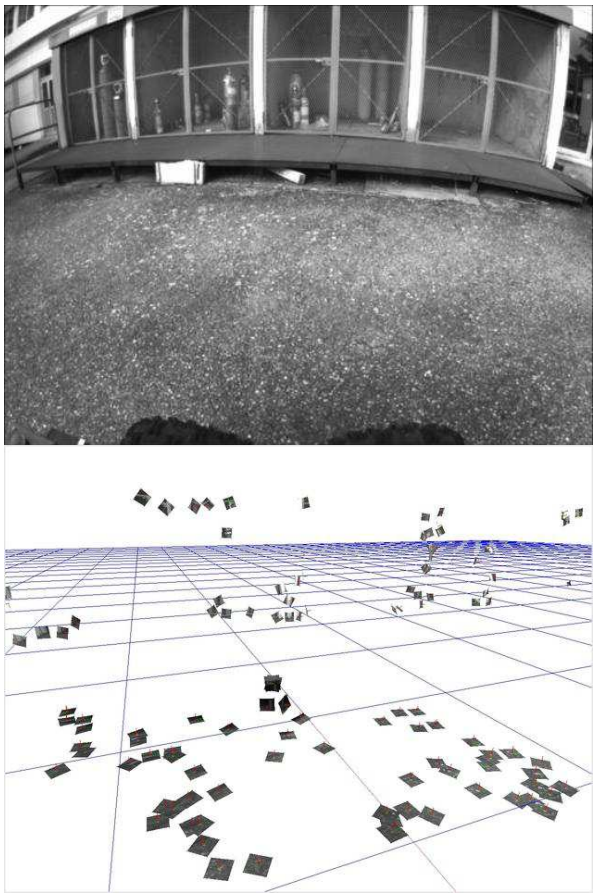

Fig. 1. Top: left image of a stereoscopic image pair. Bottom: extracted facets.

A. Facet model

a) a basis: A facet is a set of geometric properties that represent its position and orientation, completed by signal information. Figure 1 shows an example of facets extracted from a pair of stereoscopic images.

Two equivalent geometric models are defined:

- A matrix representation of the position and orientation of the facet (12 parameters: the facet center, plus the 3 vectors of the associated frame)
- A minimal representation (six Euler parameters)

The matrix representation is used to compute comparisons and transformations during detection and matching, whereas the Euler angles are used for the SLAM estimation.

To simplify the matching process, facets correspond to a constant size of planar patches in the environment (we typically use a size of 10×10 centimeters), and the associated texture is stored in a fixed size image (25×25 pixels in our implementation).

B. Facets extraction

b) Interest point detection: Interest points are image pixels to which are associated numeric properties that are stable with respect to viewpoint changes. A facet can be associated to a Harris point [6], or to scale invariants points [13], [11], [7] – the later offer a better repeatability, at the expense of a much higher computation time.

c) Homography estimation: Dense pixel stereovision could be used to estimate the normal vector of the surface corresponding to an interest point, with a least square plane fitting algorithm applied to the neighbouring 3D points. But fast stereovision algorithms yields noisy coordinates of the 3D points, which make the estimation of the normal very unstable.

An approach based on the homography estimation is more robust and reliable. The two image projections I_1 and I_2 of a plane P corresponding to different viewpoints are linked by a homography $s * H$, where H is a 3x3 matrix, and s is an unknown scale factor (often defined such that $(s*H)(3,3) = 1.0$). So two images I_1^p and I_2^p extracted from I_1 and I_2 correspond to a plane in the environment if there is a matrix H that satisfies:

$$H * I_2^p = I_1^p \tag{1}$$

Alignment algorithms which compute the value of H are optimization procedures whose goal is to minimize:

$$E = H * I_2^p - I_1^p - (\mu(H * I_2^p) - \mu(I_1^p)) \tag{2}$$

Where $\mu(H * I_2^p)$ and $\mu(I_1^p)$ are the mean of the pixels of $H * I_2^p$ and I_1^p, which reduce the influence of lightning change between two images.

An analysis of various alignment algorithms is available in [2], in which is also proposed a new method for homography estimation called "Inverse Compositional Estimation" (ICE). [12] introduce the "Efficient Second-order Minimization" (ESM) used for tracking large planar areas using an homography estimation.

For small image areas, both methods are able to estimate an homography which either precisely corresponds to the associated plane or is totally erroneous. Experimental trials show that when an erroneous homography is estimated, the resulting normal is completely random: those cases can therefore be identified by analysing successive observations (see III-D). We noticed that the ICE approach yields less erroneous results with small image patches (around 20 pixels by 20 pixels): in such cases, ICE is slightly better than ESM, unlike what is observed for larger image patches in [2] and [12].

d) Normal estimation: Once the homography is computed, the normal of the facet is computed using the geometric parameters of the stereovision bench – *e.g.* by computing the coordinates of three points of the plane using the homography.

e) Completing the facet orientation information: The facet orientation is defined by three vectors: it is only necessary to compute two of them, the third one being the result of their cross product. The first vector is the normal vector, and the second vector is computed on the basis of the texture of the facet, so as to represent its orientation: the gradient is computed on each pixel P of a square window W

around the interest point IP, using Sobel masks. The facet orientation is then defined as the following weighted sum:

$$Orientation = \frac{\sum_{P \in W} w(d(P, IP) * atan2(Gy(P), Gx(P))}{\sum_{P \in W} w(d(P, IP)} \quad (3)$$

Where $d(P, IP)$ is the distance between the pixel P and the interest point IP and $w(x)$ is a Gaussian weighting function.

Unfortunately, despite the decrease of sensitivity to noise and to viewpoint changes brought by the weighted sum, the orientation is either very stable (in most cases) or very random. As for the computation of homography, facets whose orientation is not stable can be eliminated by analysing successive observations (see III-D). In our convention, this orientation is the third Euler angle of the facet ("roll", denoted w).

C. Texture

The texture of a facet F is interpolated from the image of the camera, using the geometric properties of the facet. Each point p_t of the texture correspond to a 3D point $P \in F$, this point P is then projected on a pixel p_c of the camera.

Let $\mathcal{P}_{Camera}$ the projection matrix of a point in the environment on the focal plane of the camera, OF the vector from the origin of the world to the center of the facet F, and v and w, the orientation vectors parallel to the facet plane. Assuming the texture pixels are indexed from the facet center by i and j, and given r the resolution of the texture, the following equation gives the value for each pixel of texture as shown figure 2 :

$$p_t(i, j) = p_c(\mathcal{P}_{Camera}(OF + i * v * r + j * w * r)) \quad (4)$$

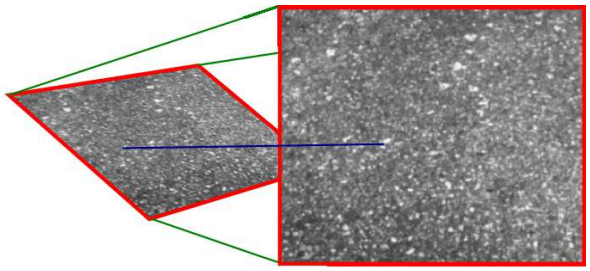

Fig. 2. Interpolation of the texture of a facet. The blue line shows how a pixel of the image is associated to a pixel of the texture.

By applying this interpolation to memorize the facet texture, the texture of the facet is represented the way it would have been perceived with the camera "aligned" to the facet, *i.e.* with the optical axis parallel to the facet normal, and the horizontal axis aligned to the facet orientation w. Thanks to this representation, during matching, a pixel by pixel comparison of the texture allows to get a similarity score between the observed texture and the memorized texture.

D. Error model

The error model for the minimal geometric representation of facets is made of covariances of its center coordinates and of its Euler angles. The center coordinates and the orientation angles being computed by independent processes, the center/orientation covariances are equal to 0. Similarly, the facet normal estimate is provided by the homography estimate, and its orientation by a analysis of the texture: these parameters variances are therefore independent. This yield a covariance matrix with the following form:

$$\begin{bmatrix} M^{stereo}_{[3\times3]} & & 0_{[3\times3]} & \\ & \sigma_u & \sigma_{u/v} & 0 \\ 0_{[3\times3]} & \sigma_{v/u} & \sigma_v & 0 \\ & 0 & 0 & \sigma_w \end{bmatrix} \quad (5)$$

Where $M^{stereo}_{[3\times3]}$ is the stereovision usual error model [18]. The variance and covariances value for the angles are empirically set as follows: $\sigma_u = \sigma_v = \sigma_w = 0.1$ and $\sigma_{u/v} = 0.01$.

III. Facets matching

A. General Algorithm

The method used for facets matching is an extension to the third dimension of an interest point matching algorithm described in [8]: the idea is to mix signal information with geometric relations between neighbouring facets to assess robust matches.

Let $\mathcal{F}_1$ and $\mathcal{F}_2$ two sets of facets within which we are looking for matches. The algorithm is a hypothesize-and-test procedure: it starts by establishing a first match between a facet from $\mathcal{F}_1$ and one from $\mathcal{F}_2$ using only signal information. This first match hypothesis gives a geometric transformation $\mathcal{T}_{1\to2}(f)$, which is used to focus the search of additional matches, the discovery of additional matches reinforcing the initial hypothesis.

1) Given $f_1 \in \mathcal{F}_1$, let $f_2 \in \mathcal{F}_2$ the facet whose texture is the closest to the one of f_1 – in other word, the facet which maximizes $CompareTexture(f_1, f)$ $\forall f \in \mathcal{F}_2$ where $CompareTexture$ is a texture comparison function (for instance the ZNCC score)
2) This first match allows to compute the geometric transformation $\mathcal{T}_{1\to2}(f)$ such that:

$$\mathcal{T}_{1\to2}(f_1) = f_2 \quad (6)$$

3) $\forall f'_1 \in \mathcal{F}_1$, if there is $f'_2 \in \mathcal{F}_2$ which satisfies the following two conditions:

$$\mathcal{T}_{1\to2}(f'_1) \approx f'_2 \quad (7)$$

$$CompareTexture(f'_1, f'_2) > \mathcal{T}_{texture} \quad (8)$$

Then the couple (f'_1, f'_2) is a match.

Figure 3 shows two example of facet matching results.

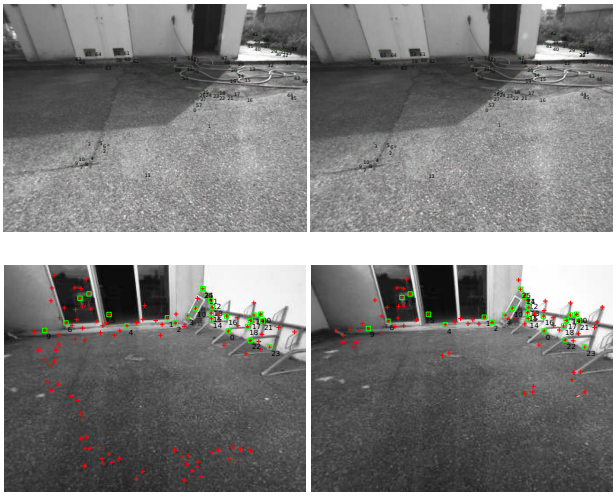

Fig. 3. Two results of facets matching. Red "+" denote the detected facets, and green numbered squares show the ones that have been matched.

B. Facets tracking

One of the advantages of using planar facets is the possibility to re-project them and to predict how a camera will observe them from a different viewpoint. Especially, if the transformation is precisely known, it is very easy to compare the observation with the texture in memory. This is of a limited interest for SLAM when the change of view point is not very well known – typically when closing a loop. But between t and $t+1$, the estimation of the viewpoint change $T_{t\rightarrow(t+1)}$ provided by the prediction step is precise enough to predict the position and orientation of the facets observed at time t to *track* them.

Let $\mathcal{I}p(I^l_{t+1})$ and $\mathcal{I}p(I^r_{t+1})$ the list of interest points detected at time $t+1$ in the left and right images I^l_{t+1} and I^r_{t+1}, and $\mathcal{F}(t)$ the set of facets detected at time t.

1) $\forall f \in \mathcal{F}(t)$, the projection P^l_f of f on the image I^l_{t+1} is computed
2) Let $\mathcal{C}$ the list of interest points located close to the predicted position of the facet on the image:

$$\mathcal{C} = Ip^l \in \mathcal{I}p(I^l_2)/|Ip - P_F| < \epsilon \qquad (9)$$

   Using the motion estimate $T_{t\rightarrow(t+1)}(base)$, it is possible to predict the facet parameters, and especially to use its predicted normal to compute the texture for each points of $\mathcal{C}$ as in section II-C. Let $I^l_p(F) \in \mathcal{C}$ the interest point whose texture is the closet to the one of the facet.
3) The same method is used to find $Ip^r(F)$ in the right image, with the added constraint that the two interest points must satisfy the epipolar constraint
4) using the couple (Ip^l, Ip^r), the parameters of the facet f_{suivi} are computed as in section II, this allow to check that $f_{suivi} = T_{t\rightarrow(t+1)}(f)$

With respect to other tracking methods (such as [16] or [12]), this approach offers the interest to get a direct control on the facets parameters, the possibility to update their models and to filter out the ones for which an erroneous homography has been estimated, as shown in the following sections. A tracking step requires 300 ms of computation (including all processings: image rectification, interest point detection and faces tracking), whereas an initial facet detection requires 500ms (including all processing), and the matching without any prior motion estimate requires a second [2].

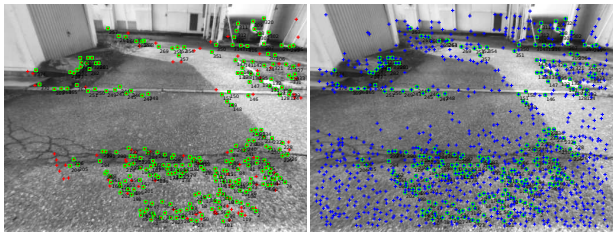

Fig. 4. Tracked facets in two consecutive images. The red "+" denote detected facets, the blue points are Harris points, and green squares shows tracked facets.

C. Facets update

Everytime a facet is tracked or matched, its model is updated: geometric information are updated by the SLAM estimation process, and the texture information is updated if the facet has been perceived from a "better position" than previously – for instance, if the new observation is closer to the previous one, and made with a smaller angle between the facet normal and the camera axis.

D. Unstable facets elimination

After the application of the matching or tracking algorithms, some facets remain unmatched, or their observation is not consistent with the matched facets observation. Such facets correspond either to an interest point with a too small repeatability, or to an erroneous normal or rotation estimate (see section II-B). This can be due to various causes: for instance, if the neighbourhood of an interest point has a weak texture, this can lead to a wrong homography (a black point on a white wall is a strong interest point, but the resulting homography is very likely to be erroneous).

Unmatchable, untrackable and inconsistent facets are considered to be weak facets, and are simply discarded.

IV. Application to SLAM

A. Facets grouping

To constitute a landmark for SLAM, facets are grouped in clusters: one the one hand this reduces the size of the SLAM filter state used, and on the other hand it increases the chance of detecting and matching a landmark when the robot comes back to a previous location (loop closing). Indeed, using facet clusters as landmarks, it doesn't matter if one of the facet is hidden or if its interest point is not detected, as the position

[2]Time measured on a Intel core Duo @ 2GHz using only one thread, on 512×392 images.

of the landmark can be observed from the observation any other facet in the cluster.

Facets are grouped by geometric proximity, and so that the density of the group is higher close to the center of the landmark. The reason is that facets closer to the center of the landmark gives a better estimation of its position. Indeed, an error on the observation of the facet angles basis yields an higher error on the position of the landmark the farther away the facet is (the error is $\delta_{position} = \delta_{angle} * distance$, assuming δ_{angle} is small so that $\delta_{angle} \simeq tan(\delta_{angle})$).

After the detection step, we have a set $\mathcal{F}$ of facets.

1) Given $f^i \in \mathcal{F}$, given G^i the set of facets close of f^i :

$$G^i = f \in \mathcal{F} / d(f, f^i) < r \tag{10}$$

where $d(f_1, f_2)$ is the distance between two facets f_1 and f_2 and r is the radius of a landmark

2) Using this first group of facets, the center C of the landmark is computed as the barycenter:

$$OC = \frac{\sum_{f \in = G^i} w_f * f}{\sum_{f \in = G^i} w_f} \tag{11}$$

The weighting w_f is used to favor facets which are considered to be better observed, *i.e.* whose normal is parallel to the camera. Thus, the weighting function is:

$$w_i = < axe_{camera} | n_f > \tag{12}$$

3) The group of facets that define the landmark is the set:

$$f \in \mathcal{F} / d(f / OC) < r \tag{13}$$

Steps 2 and 3 could be repeated in a loop until the group of facets remain stable. But during experiments show that the group of facets does not change much during the following iterations. Figure 5 shows the result of grouping facets.

B. Integration in SLAM

Let $\mathcal{A}$ the set of landmarks in the environment, $\mathcal{F}_{tr}$ the set of facets tracked at a given time t (that is to say the set of facets which have been tracked and the facets which couldn't be tracked but were possibly in the field of view of the camera), and M_{robot} the prediction of the robot displacement (provided by *e.g.* odometry).

1) The set of tracked facets $\mathcal{F}_{tr}$ is determined using the algorithm described in section III-B, the motion estimation M_{robot}, and $\mathcal{F}_{t-1}$, which allows to deduce a set of landmark observations $\mathcal{O}$
2) if the ratio of tracked facets is bellow a given threshold:

$$\frac{|\mathcal{F}_{tr}|}{|\mathcal{F}_{t-1}|} < th_{TrackeFacets} \tag{14}$$

it is necessary to start a new detection step:

a) the facet detection process described section II-B returns the set $\mathcal{F}_{detect}$ of detected facets
b) the matching algorithm of section III-A is used to compute whether one of the landmark of $\mathcal{A}$ has reappeared in the field of view. Considering a landmark $A \in \mathcal{A}$ and if the set $\mathcal{F}_{matched}$ of matched facets between the facets of A and $\mathcal{F}_{detect}$ is not empty, $\mathcal{F}_{matched} \neq \oslash$. Then the set of observations $\mathcal{O}$ is completed with a new observation of the landmark, and the facets which are part of this landmark are removed from $\mathcal{F}_{detect}$
c) the grouping algorithm of the facets in section IV-A allows to create a new landmark $newlandmarks$

3) the sets $\mathcal{O}$ and $newlandmarks$ are used to update the Kalman filter and its state vector
4) the set $\mathcal{F}_t$ is computed by removing facets that can not be tracked anymore (because they left the field of view), and by adding the newly detected facets:

$$\mathcal{F}_t = (\mathcal{F}_t \cup \mathcal{F}_{detect}) \setminus \mathcal{F}_{untrackable} \tag{15}$$

where $\mathcal{F}_{untrackable}$ is the subset of facets of $\mathcal{F}_{t-1}$ which are not in the field of view of the camera

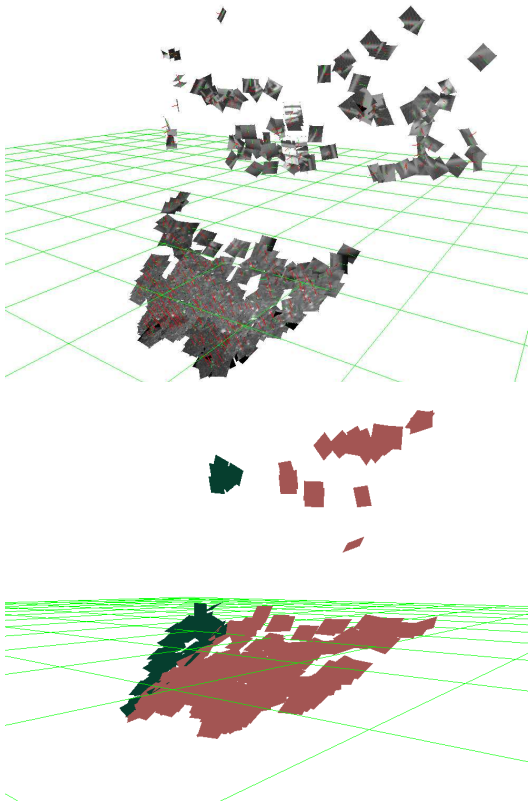

Fig. 5. The top image shows the facets which have been extracted in the environment, and the bottom one shows the two groups of facets which will be used as landmarks for SLAM.

This process is summarized by figure 6, and figure 7 shows two trajectories, one with the loop detection and one where the matching algorithm has been disabled. Naturally, applying the loop detection algorithm yields a final position estimate that is closer to the ground truth.

V. Future work

The work made until now have shown the interest of modeling the environment using facets for the SLAM. There are however some limitations that should be overcome:

- while facets are observable from different view points, as they are centered on interest points, their detection is still very sensible to changes of viewing angle
- without an heuristic to reduce the space of research, the facets matching process is a costly one. The heuristic we used in this paper is based on the estimation of the position: it will be necessary to develop other methods, especially when the position is unknown or too imprecise.

Furthermore, this representation of the environment, while richer than models using until now in vision SLAM is far to use all the available information that can be extracted from a stereovision bench. To limit this loss of information, we have decided to suppose that the transformation of two facets observed at a given time was certain (see section IV-B), and that the two facets could be inserted in a single landmark without any problems. But it would be also interesting to re-estimate the relative positions of the facets that are grouped with respect the local frame associated to the group landmark. This could be achieved by associating a Kalman filter to each group landmark, using a "Divide and Conquer" method as in [10].

References

[1] T. Bailey and H. Durrant-Whyte. Simultaneous Localisation and Mapping (SLAM): Part II - State of the Art. *Robotics and Automation Magazine*, September 2006.

[2] Simon Baker and Iain Matthews. Equivalence and efficiency of image alignment algorithms. In *Proceedings of the 2001 IEEE Conference on Computer Vision and Pattern Recognition*, December 2001.

[3] A. Davison. Simultaneous localisation and mapping with a single camera. In *9th International Conference on Computer Vision, Nice (France)*, October 2003.

[4] H. Durrant-Whyte and T. Bailey. Simultaneous Localisation and Mapping (SLAM): Part I - The Essential Algorithms. *Robotics and Automation Magazine*, June 2006.

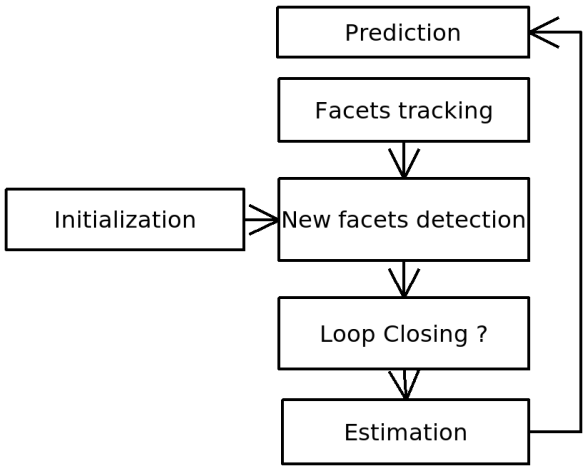

Fig. 6. The various steps of the SLAM algorithm

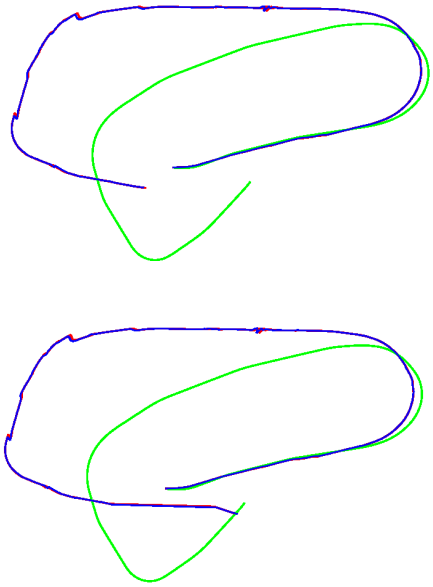

Fig. 7. On the two figures, the green trajectory is the result of odometry, the blue trajectory is the position of the robot given after the prediction step, and the red trajectory (which is nearly confused with the blue one) is the trajectory after using observations. The top figure shows the trajectory without the matching algorithm, in other word without the loop closing detection, while the bottom figure shows the result of the loop closing.

[5] E. Eade and T. Drummond. Edge landmarks in monocular slam. In *British Machine Vision Conference, Edinburgh (UK)*, Sep. 2006.

[6] C. Harris and M. Stephens. A combined corner and edge detector. In *4th Alvey Vision Conference*, pages 147–151, 1988.

[7] Luc Van Gool Herbert Bay, Tinne Tuytelaars. SURF: Speeded up robust features. In *9th European Conference on Computer Vision*, 2006.

[8] I-K. Jung and S. Lacroix. A robust interest point matching algorithm. In *8th International Conference on Computer Vision, Vancouver (Canada)*, July 2001.

[9] T. Lemaire and S. Lacroix. Monocular-vision based SLAM using line segments. In *IEEE International Conference on Robotics and Automation, Roma (Italy)*, April 2007.

[10] J.D. Tards L.M. Paz, P. Jensfelt and J. Neira. EKF SLAM updates in O(n) with divide and conquer SLAM. In *IEEE Int. Conf. Robotics and Automation*, Rome, Italy, april 2007.

[11] D. Lowe. Distinctive features from scale-invariant keypoints. *International Journal on Computer Vision*, 60(2):91–110, 2004.

[12] Ezio Malis. Improving vision-based control using efficient second-order minimization techniques. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, April 2004.

[13] Krystian Mikolajczyk and Cordelia Schmid. Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004.

[14] N. Molton, A.J. Davison, and I. Reid. Locally planar patch features for real-time structure from motion. In *British Machine Vision Conference, London (UK)*, Sept. 2004.

[15] S. Se, D. Lowe, and J. Little. Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. *International Journal of Robotics Research*, 21(8):735–758, 2002.

[16] J. Shi and C. Tomasi. Good features to track. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'94)*, Seattle (USA), June 1994.

[17] P. Smith, I. Reid, and A. Davison. Real-time monocular slam with straight lines. In *British Machine Vision Conference, Edinburgh (UK)*, Sep. 2006.

[18] Y. Xiong and L. Matthies. Error analysis of a real time stereo system. In *IEEE Conference on Computer Vision and Pattern Recognition, Puerto Rico*, pages 1087–1093, June 1997.