Towards Geographical Referencing of Monocular SLAM Reconstruction Using 3D City Models: Application to Real-Time Accurate Vision-Based Localization

Pierre Lothe, Steve Bourgeois, Fabien Dekeyser
CEA, LIST, Embedded Vision Systems
Boîte Courrier 94, Gif-sur-Yvette, 91191 France
{name.Surname}@cea.fr

Eric Royer, Michel Dhome
LASMEA (CNRS / UBP)
24 Avenue des Landais, Aubière, 63177 France
{name.Surname}@lasmea.univ-bpclermont.fr

Abstract

In the past few years, lots of works were achieved on Simultaneous Localization and Mapping (SLAM). It is now possible to follow in real time the trajectory of a moving camera in an unknown environment. However, current SLAM methods are still prone to drift errors, which prevent their use in large-scale applications.

In this paper, we propose a solution to reduce those errors a posteriori. Our solution is based on a post-processing algorithm that exploits additional geometric constraints, relative to the environment, to correct both the reconstructed geometry and the camera trajectory. These geometric constraints are obtained through a coarse 3D modelisation of the environment, similar to those provided by GIS database.

First, we propose an original articulated transformation model in order to roughly align the SLAM reconstruction with this 3D model through a non-rigid ICP step. Then, to refine the reconstruction, we introduce a new bundle adjustment cost function that includes, in a single term, the usual 3D point/2D observation consistency constraint as well as the geometric constraints provided by the 3D model. Results on large-scale synthetic and real sequences show that our method successfully improves SLAM reconstructions. Besides, experiments prove that the resulting reconstruction is accurate enough to be directly used for global relocalization applications.

1. Introduction

Simultaneous Localization and Mapping (SLAM) and Structure From Motion (SFM) are major research topics in computer vision. Indeed, they make possible the reconstruction of both the environment and the trajectory of one or several moving cameras. Various approaches are proposed to tackle this problem in real-time.

Nister et al. [13] propose to compute the camera movements without any global optimization. It enables him to speed up the computation time and to process large-scale sequences but it is very sensitive to error accumulations since the 3D scene geometry is never questioned. Davison et al. [5] propose a Kalman-filter based solution which permits SLAM to reach real-time assuming that the number of landmarks is quite small. Another approach is to use a full non-linear optimization of the scene geometry: Royer et al. [14] use a hierarchical bundle adjustment in order to build large-scale scenes. Afterwards, Mouragnon et al. [12] propose an online incremental non-linear minimisation method, reducing the necessary computation time and resources by only optimizing the position of the geometry scene on the few last cameras.

Nevertheless, monocular SLAM and SFM methods still present limitations: the trajectory and 3D point cloud are known up to a similarity. Indeed, all the displacements and 3D positions are relative and it is not possible to obtain an absolute localization of each reconstructed element. Besides, in addition to being prone to numerical error accumulation [5, 12], monocular SLAM algorithms may present scale factor drift: their reconstructions are done up to a scale factor, theoretically constant on the whole sequence, but often fluctuating in practice. Even if these errors could be tolerated for guidance applications which only use local relative displacements, they become a burning issue for other applications using SLAM reconstruction results like trajectometry or global localization.

Therefore, we propose in this paper to adjust large-scale SLAM reconstructions by using a post-processing method, introducing a coarse 3D model of the environment.

2. Previous Works And Contribution

Several different approaches have been proposed to prevent or correct SLAM reconstructions drift.

A proposed approach is to use cameras which provide very large field of view. For example, Tardif et al. [17] show that using omnidirectional cameras drastically reduces the

978-1-4244-3991-1/09/$25.00 ©2009 IEEE

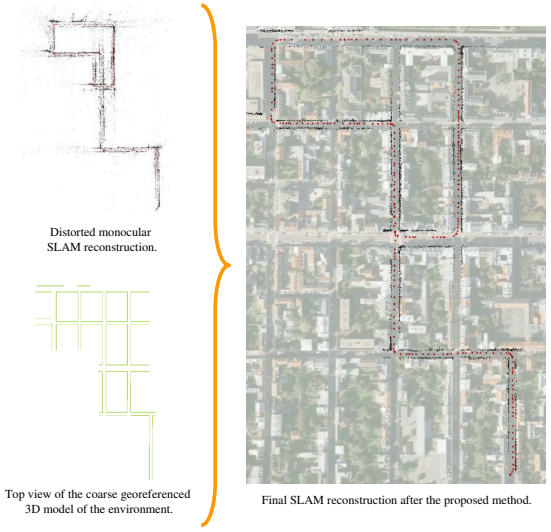


Figure 1. **Summary of the proposed method.** We proposed to correct an original distorted SLAM reconstruction thanks to a coarse 3D model of the environment. The result is superimposed on a satellite image.

trajectory estimation drift. Nevertheless, such a camera implies a rise of cost and integration complexity that seems incompatible with standard vehicle.

To improve SLAM reconstructions, another approach could be to still use a low-cost video sensor but to introduce additional constraints. For example, Clemente et al. [4] use trajectory loop information. However, this approach is only efficient if the same place is crossed several times. Besides, even if loop correction limits the drift effects, it does not correct the trajectory consistency: a square-like trajectory could be reconstructed as any rectangular path.

On the other side, the additional constraints could come from another sensor or from external information. Levin et al. [10] propose to correct large-scale reconstructions thanks to a method introducing a hand-drawn map of the trajectory. A segment-based transformation is applied to the resulting trajectory to fit this hand-drawn map. Then, the result is used as initialisation in a classical bundle adjustment to correct the residual local errors of camera positions. Nevertheless, the additional information brought by the map are not used during the bundle adjustment so that bad deformations could appear again. On the contrary, Sourimant et al. [16] develop a SFM algorithm that exploits the knowledge of a coarse 3D model of the environment. Indeed, those models are now widely spread, for example by GIS database, and tend to be standardised[1]. Their method leans on projection of feature points detected in the image onto the 3D model to compute the 3D points and then on a classical KLT [18] tracker. A limitation is that the 3D model is never questioned. The precision of the reconstructed cameras is then directly linked to the precision of this 3D model. In the same way, Fruh et al. [7] blend two 3D models, one from aerial images and one from ground views, without calling the last one into questions because it is supposed correct.

Like Sourimant and Fruh, our solution consists in exploiting a 3D model of the environment (see figure 1) in order to tackle the challenging problem of correcting large-scale SLAM reconstructions. Nevertheless, we consider that exploiting an errorless 3D environment model is not a realistic hypothesis. Therefore, we introduce a two steps post-processing algorithm that fuses SLAM reconstruction with 3D model by taking into account the uncertainty of both of them. Our first contribution relates to a non-rigid ICP algorithm: we use an articulated-like transformation model to align the reconstructed point cloud with the model (section 3). Then, an original cost function is proposed to correct the residual local trajectory position errors, with respect to the model, through a bundle adjustment step (section 4). Finally, we evaluate the complete refinement process on both synthetic and real sequences and present an example of global localization application in section 5.

3. Coarse SLAM Reconstruction Correction

The method we describe in the following is summarised in figure 2. The inputs to our system is the SLAM reconstruction (using Mouragnon algorithm [12]), i.e. both the reconstructed trajectory and the 3D points cloud of the environment (see figure 2(b)). Moreover, we have a very simple 3D model of the environment, composed of vertical planes representing the fronts of buildings (figure 2(d)).

3.1. Overview

To correct the reconstruction geometry distorted by drift, we propose to fit it with this model of the environment. Such a kind of registration has been widely studied and mainly used methods to solve this problem are Iterative Closest Point ICP [15] and Levenberg-Marquardt [6]. Nevertheless, these methods were initially developed for Euclidean transformations or similarities, which is not adapted to our problem due to the inherent deformation complexity of SLAM reconstructions. To overcome with this limit, non-rigid fitting methods have been proposed. Because of their high number of degrees of freedom, this category of algorithm needs to be constrained (see [3] for an example of regularized non-rigid fitting).

We choose to restrict our problem to a specific class of non-rigid transformations that will approximate as well as possible the deformations induced by SLAM process. We observe that, for a classical vehicle path, the scale factor is nearly constant on straight lines trajectory and changes

[1] http://www.citygml.org/

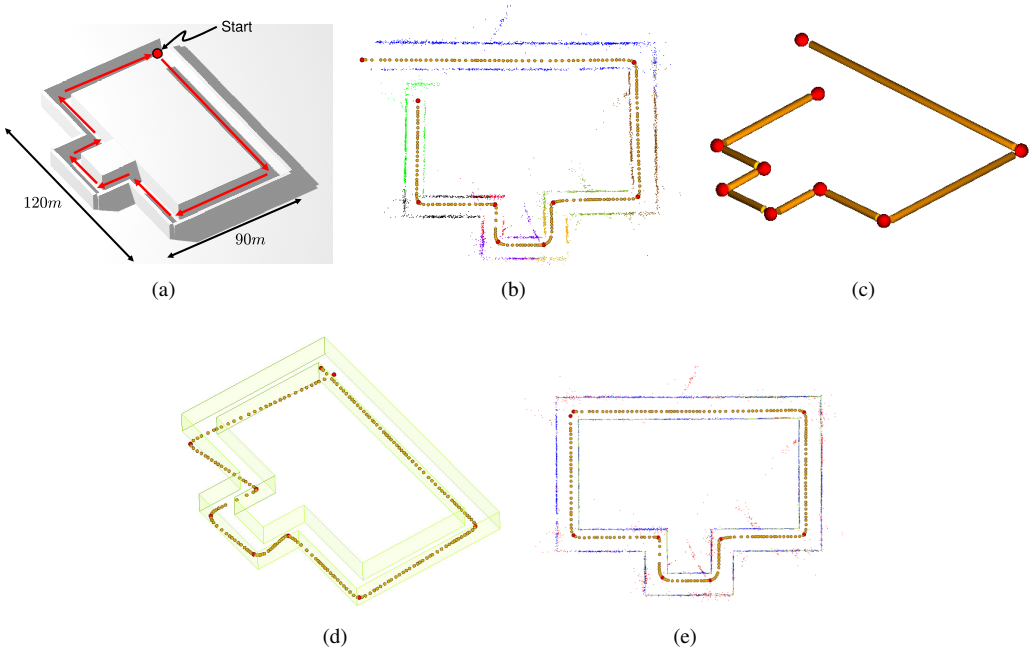


Figure 2. **Non-rigid ICP method summary.** (a) 3D model of the environment and followed path. (b) SLAM initial reconstruction with Mouragnon [12] algorithm. The automatic trajectory and point-fragment segmentation is also represented: orange spheres are cameras while red ones are the fragment extremities. Reconstructed 3D points are coloured by fragment belonging. (c) The proposed piecewise similarity model. (d) Trajectory initialisation around the 3D model (in green). (e) Final reconstruction after our non-rigid ICP step: blue 3D points are inliers and red one are outliers.

during turnings (see figure 2(b)). So we have decided to use a segment-based model like Levin [10] did: trajectory straight lines are considered as rigid bodies while articulations are put in place in each turning (figure 2(c)). Thus the selected transformations are piecewise similarities with joint extremity constraints.

We will first present the 3D SLAM reconstruction fragmentation method. Afterwards, we will propose a non-rigid ICP step to find the piecewise similarities that best-fits the reconstructed point cloud and the 3D model. At last, we will obtain a coarsely corrected SLAM reconstruction in a well-known coordinate frame (see figure 2(e)).

3.2. SLAM Reconstruction Fragmentation

The fragmentation step aims to segment the trajectory and then associate each 3D reconstructed point to one of those trajectory segments.

To segment the camera trajectory, we use the idea suggested by Lowe in [11] where he proposes to recursively cut a set of points into segments with respect to both their length and deviation. So we split the reconstructed trajectory (represented as a set of temporally ordered key cameras) into m different segments $(\mathcal{T}_i)_{1\leq i\leq m}$ whose extremities are camera positions denoted $(\mathbf{e}_i, \mathbf{e}_{i+1})$.

Then to associate each 3D point to a trajectory segment, we define this rule: we say that a segment "sees" a 3D point if at least one camera of the segment observes this point. Two cases are then possible. The simplest one is when only one segment sees this point: the point is then linked to this segment. The second appears when a point is seen by two or more different segments. In this case, we have tested different policies which give similar results and thus we arbitrarily decided to associate the point to the last segment which sees it.

In the following, we call $\mathcal{B}_i$ a fragment composed both of the cameras of $\mathcal{T}_i$ (i.e. those included between $\mathbf{e}_i$ and $\mathbf{e}_{i+1}$) and of the associated reconstructed 3D points. Obviously, $\forall i \in [2, m-1]$, the fragment $\mathcal{B}_i$ shares its extremities with its neighbours $\mathcal{B}_{i-1}$ and $\mathcal{B}_{i+1}$.

3.3. Non-Rigid ICP

Once the different fragments are computed, the piecewise similarity (with joint extremities constraints) that best fits the reconstructed 3D point cloud with the 3D model can be estimated. Practically, those transformations are parameterized with the 3D translation of the extremities $(\mathbf{e}_i)_{1\leq i\leq m+1}$. From these translations, we deduce the similarities to apply to each fragment (i.e. its cameras and 3D points). Since the camera is embedded on a land vehicle, we chose to disregard the roll angle. Then, each extremity $\mathbf{e}_i$ has 3 degrees of freedom and so each fragment has 6 as expected.

The problem we want to solve is then to find the 3D positions of the extremities which minimize the distance between the reconstructed 3D points $\mathcal{Q}_i$ and the 3D model $\mathcal{M}$ that is to say:

$$\min_{\mathbf{e}_1,...,\mathbf{e}_{m+1}} \sum_i d(\mathcal{Q}_i(\mathbf{e}_1,...,\mathbf{e}_{m+1}),\mathcal{M})^2 \quad (1)$$

where d is the normal distance between the 3D point $\mathcal{Q}_i(\mathbf{e}_1,...,\mathbf{e}_{m+1})$ (simply denoted $\mathcal{Q}_i$ in the following) and the 3D model $\mathcal{M}$.

Point-Plane Association. The distance d should be the distance between the 3D point $\mathcal{Q}_i$ and the plane of the 3D model it belongs to in reality. Nevertheless, this plane is unknown in our case. So, we decide to compute the distance d as the distance between $\mathcal{Q}_i$ and its nearest plane $\mathcal{P}_{h_i}$. As done in classical ICP algorithms, we suppose that the nearest plane $\mathcal{P}_{h_i}$ does not change during the minimization. Thus, the selection between the 3D point and the corresponding plane can be done outside the minimization:

$$\forall \mathcal{Q}_i, \mathcal{P}_{h_i} = \underset{\mathcal{P}\in\mathcal{M}}{argmin}\ d(\mathcal{Q}_i,\mathcal{P}) \quad (2)$$

and the problem to solve becomes:

$$\min_{\mathbf{e}_1,...,\mathbf{e}_{m+1}} \sum_i d(\mathcal{Q}_i,\mathcal{P}_{h_i})^2. \quad (3)$$

It is important to notice that the distance d takes into account that the planes are finite: to be associated to a plane $\mathcal{P}$, a 3D point $\mathcal{Q}$ must have its normal projection inside $\mathcal{P}$ bounds.

Robust Estimation. There are two cases where the association $(\mathcal{Q}_i,\mathcal{P}_{h_i})$ can be wrong: if the initial position of $\mathcal{Q}_i$ is too far from its real position or if it is not (in the real scene) on the surfaces of the model. In those two cases, $d(\mathcal{Q}_i,\mathcal{P}_{h_i})$ could make the minimization fail. To limit this effect, we insert a robust M-estimator ρ in equation (3):

$$\min_{\mathbf{e}_1,...,\mathbf{e}_m} \sum_i \rho(d(\mathcal{Q}_i,\mathcal{P}_{h_i})) \quad (4)$$

We chose to use the Tukey M-estimator [8]. The M-estimator threshold can be automatically set thanks to the Median Absolute Deviation (MAD). The MAD works with the hypothesis that the studied data almost follow a Gaussian distribution around the model. This assumption could be done for each individual fragment but not for the whole reconstruction. So we decided to use a different M-estimator threshold ξ_j per fragment. This implies that we have to normalize the Tukey values on each fragment:

$$\rho'_{l_i}(d(\mathcal{Q}_i,\mathcal{P}_{h_i})) = \frac{\rho_{l_i}(d(\mathcal{Q}_i,\mathcal{P}_{h_i}))}{\max\limits_{\mathcal{Q}_j\in\mathcal{B}_{l_i}} \rho_{l_i}(d(\mathcal{Q}_j,\mathcal{P}_{h_j}))} \quad (5)$$

where l_i is the index of the fragment owning $\mathcal{Q}_i$ and ρ_{l_i} the Tukey M-estimator used with the threshold ξ_{l_i}.

Fragment Weighting. With the cost function (5), each fragment will have a weight in the minimization proportional to the number of 3D points it contains. Then, fragments with few points could be not optimized in favour of the others. To give the same weight to each fragment, we must unify all the Tukey values of their 3D points with respect to their cardinal:

$$\rho^*_{l_i}(d(\mathcal{Q}_i,\mathcal{P}_{h_i})) = \frac{\rho'_{l_i}(d(\mathcal{Q}_i,\mathcal{P}_{h_i}))}{card(\mathcal{B}_{l_i})} \quad (6)$$

and the final minimization problem is:

$$\min_{\mathbf{e}_1,...,\mathbf{e}_{m+1}} \sum_i \rho^*_{l_i}(d(\mathcal{Q}_i,\mathcal{P}_{h_i})) \quad (7)$$

that we solve using the Levenberg-Marquardt algorithm [9].

Iterative Optimisation. Practically, as performed in classical ICP method, several non-linear minimisations are done successively. The point-plane association is computed before each one of them. It enables 3D points to change their associated plane with limited impact on computation time.

Initialisation. Non-linear algorithms require a correct initialisation: the 3D reconstruction should be placed in the same frame than the model. To realize this stage, estimating an initial rigid transformation is sufficient when the 3D reconstruction is accurate. However, the drift frequently observed with SLAM reconstruction may induce important geometrical deformations. Therefore, to ensure convergence of the algorithm, we chose to roughly place each extremity $\mathbf{e}_i$ around the model. It could be done automatically if the sequence is synchronised with GPS data for example. Otherwise, it can be realized manually through graphic interface.

4. Bundle Adjustment Introducing 3D model

Nevertheless, the deformation model used in the previous step was overconstrained to reach a fine correction: in our hypothesis, each trajectory segment is a rigid body and consequently cameras of a segment do not move relatively to each other. Thus, errors along the trajectory direction could be still significant on each fragment. Experiment results emphasize this behaviour (figure 4(a)).

To reduce those residual errors, we have then to optimize all the reconstruction geometry (i.e. the camera poses and 3D point position). This problem is often tackled with a bundle adjustment. However, since classical bundle adjustment only minimizes the reprojection error in the images, the geometric consistency between reconstructed 3D point cloud and the 3D model would be suppress. We have observed that it can then converge back towards the original

SLAM reconstruction. That is why we will now propose a new bundle adjustment cost function which includes the 3D model information.

4.1. Proposed Bundle Adjustment Cost Function

Two kinds of information have to be included in the cost function: the relation between the cameras and the observed 3D points and the geometric consistency between these reconstructed 3D points and the model. A possible approach is to sum those two residual errors (one for each kind of information), which must have the same unit, up to a factor λ. In this kind of solution, the factor λ is often hard to determine and specific to the processed sequence. Thus, we propose in the following a cost function including the two relations in a single residue.

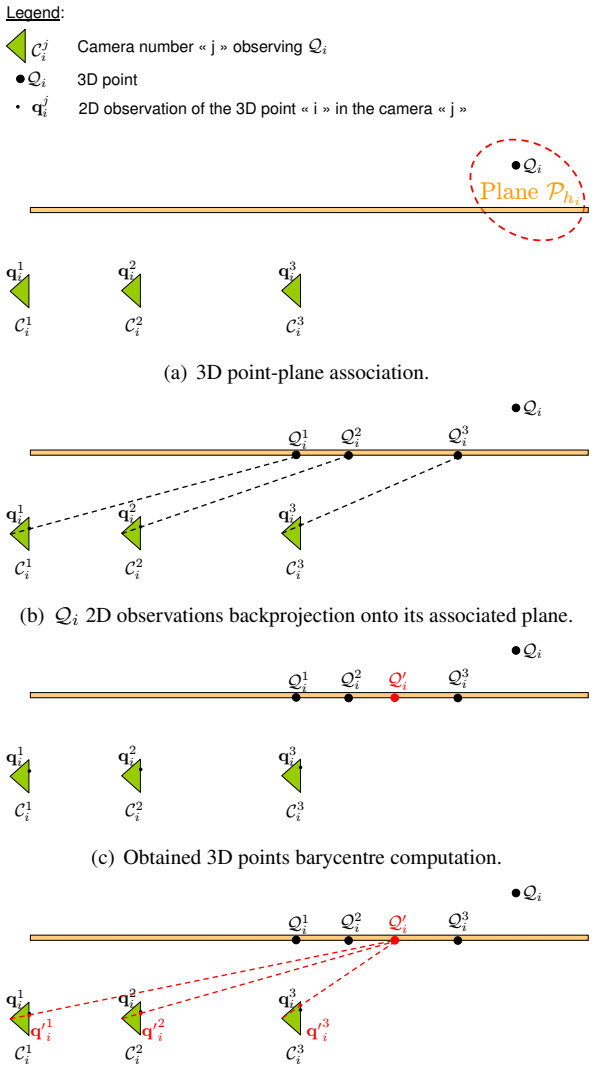
This cost function is resumed in figure 3. The main idea is to encourage each 3D reconstructed point to belong to a plane of the model. For this, let us consider a reconstructed 3D point $\mathcal{Q}_i$. First of all, as performed in the non-rigid ICP, we associate it with its nearest plane $\mathcal{P}_{h_i}$ in the model. Furthermore, we consider the set of cameras $(\mathcal{C}_i^j)_j$ observing $\mathcal{Q}_i$ and we note $(\mathbf{q}_i^j)_j$ those 2D observations (figure 3(a)). Then, for each camera $\mathcal{C}_i^j$, we compute the 3D point $\mathcal{Q}_i^j$ as the intersection between the bundle issued from $\mathbf{q}_i^j$ backprojection and the plane $\mathcal{P}_{h_i}$ (figure 3(b)). The $(\mathcal{Q}_i^j)_j$ barycentre is denoted $\mathcal{Q}'_i$ (figure 3(c)). $\mathcal{Q}'_i$ is then the 3D point on the plane $\mathcal{P}_{h_i}$ associated to the 3D point $\mathcal{Q}_i$. Advantages to use the barycentre of backprojections, for example compared to directly projecting $\mathcal{Q}_i$ onto $\mathcal{P}_{h_i}$, are that the movement of $\mathcal{Q}'_i$ is then directly linked to the displacement of the cameras. Besides, since $\mathcal{Q}_i$ position is not used in the cost function, we do not have to optimise its parameters. Thus, the problem complexity is considerably reduced.

Then, to measure the $\mathcal{Q}'_i$ reprojection error, it is projected into each camera $\mathcal{C}_i^j$ and the resulted 2D points are named $\mathbf{q'}_i^j$ (figure 3(d)). The function we minimize is then the squared sum of the residual 2D distances between the points $\mathbf{q}_i^j$ and $\mathbf{q'}_i^j$ with respect to the pose of each camera parameterised by 3 translation parameters and 3 rotation parameters.

It is important to note that the final optimised position of $\mathcal{Q}_i$ is obtained by triangulating its observations $(\mathbf{q}_i^j)_j$ with the newly optimised camera parameters. Thus, the proposed cost-function does not force $\mathcal{Q}_i$ to be on a plane: it only encourages it to be near a plane while its position must still be consistent with its observations in the images.

4.2. Robust Optimisation

To be robust to 3D points outliers, we use the Geman-McClure [2] M-estimator in the minimised cost function with an automatic threshold computation based on MAD.



(a) 3D point-plane association.

(b) $\mathcal{Q}_i$ 2D observations backprojection onto its associated plane.

(c) Obtained 3D points barycentre computation.

(d) Barycentre projection into cameras images.

Figure 3. **Proposed cost function.** Example of a 3D point $\mathcal{Q}_i$ observed by 3 cameras. The successive steps are described in subfigures (a) to (d). The residues are the 2D distances between $(\mathbf{q}_i^j)_j$, the observations of $\mathcal{Q}_i$, and $(\mathbf{q'}_i^j)_j$, the projections of its associated 3D point $\mathcal{Q}'_i$.

Nevertheless, the cost function proposed above optimizes the camera poses from the previously estimated positions of 3D points $(\mathcal{Q}_i)_i$. Therefore, an inaccurate position of $(\mathcal{Q}_i)_i$ can lead to a wrong point-plane association and prevent the optimal convergence of the optimisation process (as in section 3.3). To outperform this point-plane association problem, we have to update the position of 3D points $(\mathcal{Q}_i)_i$ (which is obviously not necessarily the one of $(\mathcal{Q}'_i)_i$) with respect to the new camera poses. Thus, to solve the global problem, we iteratively compute positions of 3D points and camera poses. The global method iterates the following steps:

- Computation of the camera poses by minimizing the cost function proposed in section 4.1 through the Levenberg-Marquardt algorithm [9].

- Triangulation of 3D points $(\mathcal{Q}_i)_i$ from observations and new camera poses.

- Association of those 3D points with their nearest plane.

The following section presents results provided both by ICP and bundle adjustment.

5. Experimental Results

In this section, we will present results on both synthetic and real sequences. The SLAM algorithm we use for our reconstruction is the one proposed by Mouragnon [12]. Then we will present possibilities provided by the SLAM reconstruction we have corrected through an example of global localization application.

5.1. Synthetic Sequence

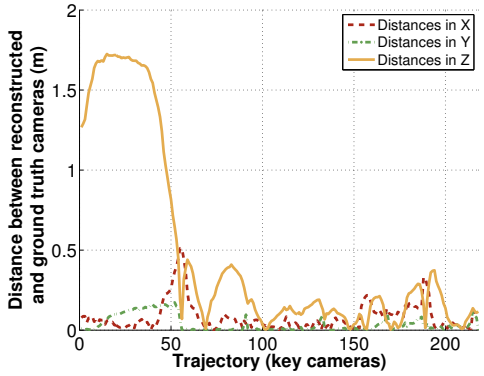
Figure 2 presents the different steps of our experiment. The synthetic sequence (based on 3D model in figure 2(a)) have been made in order to use the SLAM algorithm in a textured 3D world generated with a 3D computer graphics software. The followed trajectory is represented by the red arrows in figure 2(a). Besides, the fact that the camera trajectory does not loop in SLAM reconstruction (figure 2(b)) underlines the original SLAM method drift.

The first step of our method is the non-rigid ICP. For its initialisation, we have manually simulated low-cost GPS results: each segment extremity has been moved with respect to the error of this kind of material (figure 2(d)). Then we observe that after the ICP step (figure 2(e)) the loop is restored while no loop constraint is directly included into our transformation model. Table 5.1 confirms those enhancements: the average distance between the reconstructed cameras and the ground truth is reduced from more than 4 meters to about 50 centimetres. It is to note that statistics in this table have been computed on the 5591 reconstructed 3D points (among the 6848 proposed by SLAM) kept as inliers by the ICP step. Furthermore, we can notice in figure 4(a) that only errors along the direction of the trajectory remain significant. This is due to the fact that our proposed transformations model supposed the drift and scale factor error strictly constant on each segment. Although, this hypothesis reveals to be only a rough approximation.

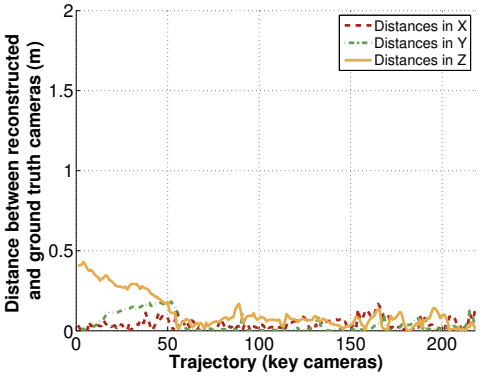
Figure 4(b) shows that the bundle adjustment step permits to correct those residual errors. The mean camera position error reaches about 14 centimetres, that is to say about three times less than after the ICP (table 5.1).

	Before ICP	After ICP	After bundle adjustment
Camera mean distance to ground truth (m)	4.61	0.51	**0.14**
Standard Deviation (m)	2.25	0.59	**0.10**
3D points-Model mean distance (m)	3.37	0.11	**0.08**
Standard Deviation (m)	3.9	0.08	**0.08**
Tukey threshold	×	0.38	×

Table 1. **Numerical results on the synthetic sequence.** Each value is a mean over all the reconstruction.



(a)



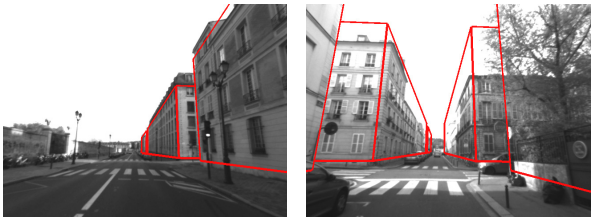
(b)

Figure 4. **Residual values on distances between reconstructed cameras and ground truth.** The (X, Y, Z) coordinates frame is relative to each camera: Z is the optical axis, X is the latitude axis and Y the altitude. (a) represents those residual values after the non-rigid ICP step and (b) after the bundle adjustment step.

5.2. Real Sequence

The real sequence is a 640x480 video of a 1500 meters long tour in Versailles, France (see figure 5(a)). Fig-

ure 5(b) shows that the 3D model is only composed by vertical planes roughly representing the building fronts. The precision of this 3D model is about 2 meters (see figure 6(b)). The initial SLAM reconstruction is a good example of SLAM method drift: by manually putting it in the same coordinates frame than the model, we observe that the trajectory moves away from its real place from the third bend onwards. After our method, figure 5(f) shows that the drift is corrected along all the tour. Indeed, the camera trajectory follows the road between the buildings and besides, the reconstructed point cloud regains its consistency. This result if confirmed by superimposing the SLAM reconstruction on a satellite image (figure 1). Furthermore, projecting the 3D model into the SLAM key frames (figure 6) permits the quality of the camera positioning to be appreciated.



(a) (b)

Figure 6. **3D model projection into SLAM key frames.** The used coarse 3D model can be close (a) or far (b) from the real scene geometry.

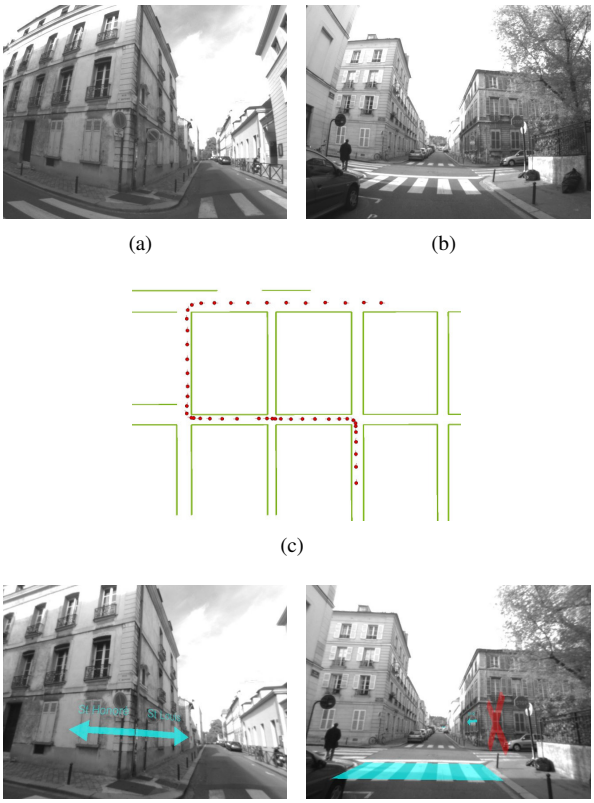
5.3. Application: Global Localization And Augmented Reality

In this part, we propose a possible application which uses our method results: the vision-based global localization. The inputs to this application are both the corrected (and thus georeferenced) SLAM reconstruction which is the knowledge database and a new set of images taken in some of the previously treated streets. The outputs are the camera poses for each one of those new images.

It seems that the proposed database is well designed to be used in vision-based localization. First, since the 3D points are reconstructed from vision-based SLAM, they are still associated to one or more feature descriptors. Moreover, our 3D point database is sparse and thus fast to explore because it is only composed of visually relevant primitives.

Figure 7 presents an example of global localization. This experiment is based on a new video of a car going through four streets of our database. We pick up many frames of this video and try to localize them with respect to our database. To realize this step, for each new current image, we first find its nearest key image in our database. Then, thanks to the distance between SURF [1] descriptors, we associate each interest point of the current image with one of reconstructed 3D points present in the chosen key frame in order to compute the camera pose. Results can be seen in figure 7(c).

The precision of the resulting pose permits for example the addition of augmented reality data in the current image (figures 7(d) and 7(e)), those additional information having simply to be added in the 3D model.



(a) (b)

(c)

(d) (e)

Figure 7. **Augmented reality on relocalised frames.** The first line presents two examples of new video frames. (c) is the result of frame global localization. Frames (d) and (e) present the result of augmented reality on frames (a) and (b).

6. Conclusion

In this paper, we have proposed a new approach to correct large-scale SLAM reconstructions when a coarse 3D model of the environment is available. Our post-processing method relies on two original steps. First, we use a fragment-based transformation, obtained thanks to a non-rigid ICP, to correct roughly the SLAM reconstruction. A specific bundle adjustment, directly introducing the 3D model, is then used to refine the scene geometry. Experiments prove that our approach successfully deals with both synthetic and real sequences. Furthermore, the proposed augmented reality application shows that the obtained reconstruction precision is sufficient to be used in global localization problem.

In future work, we would like to investigate the integra-

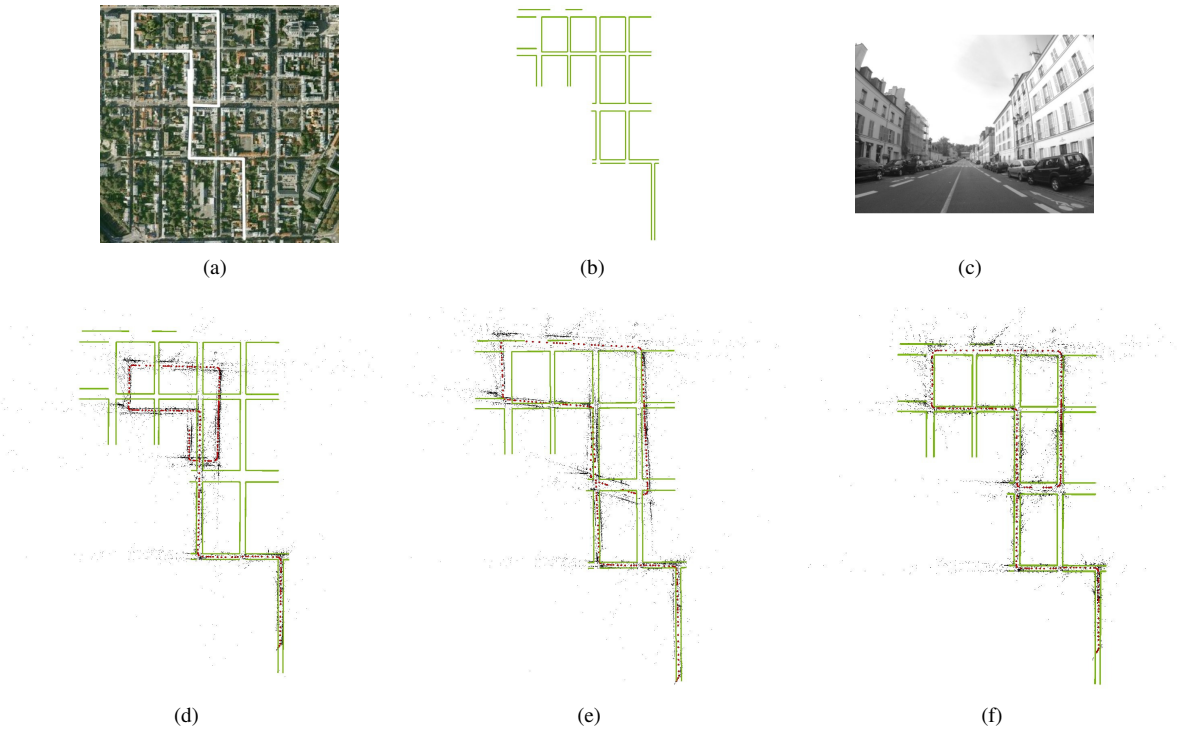


Figure 5. **Versailles sequence.** The first line presents the real sequence information: the followed trajectory (a), the coarse 3D model (b) and a frame of the recorded video (c). The second line presents the different configurations of SLAM reconstruction compared to the 3D model: the initial SLAM reconstruction with Mouragnon [12] algorithm (d), the non-rigid ICP initialisation (e) and the result of the proposed method (f).

tion of our method directly in the online SLAM treatment in order to correct progressively the reconstruction and therefore to realize real-time global localization without having to create a prior database.

References

[1] H. Bay, T. Tuytelaars, and L. V. Gool. Surf: Speeded up robust features. In *ECCV*, pages 346–359, 2006.

[2] M. J. Black and A. Rangarajan. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *IJCV*, 19(1):57–91, 1996.

[3] U. Castellani, V. Gay-Bellile, and A. Bartoli. Joint reconstruction and registration of a deformable planar surface observed by a 3d sensor. In *3DIM*, pages 201–208, 2007.

[4] L. Clemente, A. Davison, I. Reid, J. Neira, and J. Tardos. Mapping Large Loops with a Single Hand-Held Camera. In *RSS*, 2007.

[5] A. Davison, I. Reid, N. Molton, and O. Stasse. MonoSLAM: Real-time single camera SLAM. *PAMI*, 26(6):1052–1067, 2007.

[6] A. Fitzgibbon. Robust registration of 2d and 3d point sets. In *BMVC*, pages 411–420, 2001.

[7] C. Fruh and A. Zakhor. Constructing 3d city models by merging aerial and ground views. *IEEE CGA*, 23(6):52–61, 2003.

[8] P. Huber. *Robust Statistics*. Wiley, New-York, 1981.

[9] K. Levenberg. A method for the solution of certain non-linear problems in least squares. *Quart. Appl. Math.*, 2:164–168, 1944.

[10] A. Levin and R. Szeliski. Visual odometry and map correlation. In *CVPR*, pages 611–618, 2004.

[11] D. G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31(3):355–395, 1987.

[12] E. Mouragnon, F. Dekeyser, P. Sayd, M. Lhuillier, and M. Dhome. Real time localization and 3d reconstruction. In *CVPR*, pages 363–370, 2006.

[13] D. Nister, O. Naroditsky, and J. Bergen. Visual odometry. In *CVPR*, pages 652–659, 2004.

[14] E. Royer, M. Lhuillier, M. Dhome, and T. Chateau. Localization in urban environments: Monocular vision compared to a differential gps sensor. In *CVPR*, pages 114–121, 2005.

[15] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *3DIM*, pages 145–152, 2001.

[16] G. Sourimant, L. Morin, and K. Bouatouch. Gps, gis and video fusion for urban modeling. In *CGI*, may 2007.

[17] J.-P. Tardif, Y. Pavlidis, and K. Daniilidis. Monocular visual odometry in urban environments using an omnidirectional camera. In *IROS*, pages 2531–2538, 2008.

[18] C. Tomasi and T. Kanade. Detection and tracking of point features. Technical report, Carnegie Mellon University, 1991.