Real-Time Vehicle Global Localisation with a Single Camera in Dense Urban Areas: Exploitation of Coarse 3D City Models

Pierre Lothe[1] Steve Bourgeois[1] Eric Royer[2] Michel Dhome[2] Sylvie Naudet-Collette[1]

[1] CEA LIST, Vision and Content Engineering Lab, Point Courrier 94, Gif-sur-Yvette, F-91191 France
Name.Surname@cea.fr

[2] LASMEA-UMR 6602 Université Blaise Pascal/CNRS, 63177 Aubière Cedex, France
Name.Surname@lasmea.univ-bpclermont.fr

Abstract

In this system paper, we propose a real-time car localisation process in dense urban areas by using a single perspective camera and a priori on the environment. To tackle this problem, it is necessary to solve two well-known monocular SLAM limitations: scale factor drift and error accumulation. The proposed idea is to combine a monocular SLAM process based on bundle adjustment with simple knowledge, i.e. the position and orientation of the camera with regard to the road and a coarse 3D model of the environment, as those provided by GIS database. First, we show that, thanks to specific SLAM-based constraints, the road homography can be expressed only with respect to the scale factor parameter. This allows the scale factor to be robustly and frequently estimated. Then, we propose to use the global information brought by 3D city models in order to correct the monocular SLAM error accumulation. Even with coarse 3D models, turnings give enough geometrical constraints to allow fitting the reconstructed 3D point cloud with the 3D model. Experiments on large-scale sequences (several kilometres) show that the entire process permits the real-time localisation of a car in city centre, even in real traffic condition.

1. Introduction

In this system paper, we aim to tackle the problem of car positioning in dense urban cities. In fact, since they allow anybody to easily reach a precise place, car navigation systems are now widely used. Nevertheless, current navigation systems based on GPS still present limitations. Indeed, even if their precision (about 10 meters) is sufficient outside city, GPS systems incur precision and denial-of-service problems in dense urban areas due to the signal occlusion by buildings. Moreover, the ergonomy of those systems should be enhanced: at the moment, the information provided to the user (place you are, road to follow, *etc*.) are displayed on a virtual map. It could obviously be beneficial to display all those information directly on the image of the street. Nevertheless, even if many constructors have begun to introduce a camera in their navigation system for example for sign detection, the precision of GPS is too low for augmented reality display.

In this paper, we aim to show how the use of a single low-cost perspective camera and a coarse 3D model of the environment, as those provided in GIS database, can improve both the precision and the robustness of GPS navigation system in dense urban area.

1.1. Single Camera Localisation Overview

Different approaches can been distinguished by the fact that they use or not strong prior information about the environment.

In the first class of propositions, the idea is first to create offline a database (which is often a 3D point cloud associated to 2D descriptors) and then to use it online in order to obtain a global localisation of the camera [6, 2]. The remaining problem of those approaches is that the size of the created database is very high. Moreover, the robustness of the relocalisation is directly linked to the used descriptor so that it may be sensitive to illumination and point of view modification.

To avoid those limits, methods (called monocular SLAM process) have been proposed in order to localise a single camera in an unknown environment [10, 3]. Nevertheless, the computed camera localisation can be very inaccurate on large scale sequences. In fact, the localisation of the current pose of the camera is done with respect to the previous one. Thus, camera position errors are accumulated. Fur-

thermore, for monocular process, the scale factor of the obtained reconstruction is unknown. Moreover, if it is theoretically constant on the entire sequence, it appears that this scale factor can drift in practice, especially for perspective cameras because of their restricted field of view.

Thus, our goal is to show how additional simple prior knowledge about the scene can be exploited to tackle those two main monocular SLAM limitations, *i.e*. the scale factor drift and the error accumulation, in the case of a camera embedded on a car cruising in a dense city area.

1.2. Proposition Positioning

In this work, the monocular SLAM process we use is based on Mouragnon *et al*. proposal [10]. The camera trajectory is initialized thanks to the essential matrix estimation and 2D observations are then triangulated to obtain the initial 3D point cloud. For each new frame, the current camera pose is computed from the reconstructed 3D point cloud. When necessary, *i.e*. when the number of observed 3D points is too low, the current frame becomes a keyframe: new observed 3D points are added to the 3D point cloud and a local bundle-adjustment is applied on the few last key cameras (*i.e*. the camera associated to keyframes) and the associated 3D structure. Since only keyframes are used to create and optimize the 3D structure, our method aims to correct those cameras.

The paper will be divided into two parts corresponding to the two main monocular SLAM limitations. First, we show that the scale factor between two cameras can be robustly computed by using information jointly brought by the known camera position in the car and the monocular SLAM process (section 2). Then, to obtain a global positioning of the camera, we propose to avoid residual error accumulation by matching the SLAM reconstruction with the coarse 3D models provided in dense city areas (section 3). Note that those two steps are inseparable. In fact, the scale factor correction is necessary to ensure the robustness of this 3D matching. The efficiency of those two steps are respectively validated (sections 2.6 and 3.3) on large scale sequences (more than 4 kilometres) in real traffic conditions.

2. Scale Factor from Ground Plane

In this part, we will present different methods to obtain the scale factor of a monocular SLAM reconstruction in the case of a vehicle displacement. Then, we will describe the approach we propose.

2.1. Related Works

Many researches have been done on large-scale monocular SLAM process embedded on a car. An additional sensor of the car could be used in order to obtain a precise norm of the camera displacement (odometer [13], *etc*.). However, such a deep integration of the system in the car would prevent its use with vehicles which are already in circulation. Considering this integration constraint, we will consider a system that relies on a simple camera. However, note that to obtain a global localisation with SLAM, the process must be initialized with coarse absolute information as GPS or georeferenced frame recognition. We will consider in the following that this step is done.

In order to obtain the scale factor of an inter-camera displacement in monocular SLAM, a prior knowledge about an absolute distance of the environment is necessary. A classical approach for cars is to obtain totally [15] or partially [14] the camera displacement by observing the homography described by the road plane. The *a priori* knowledge of camera/ground distance is then used to constrain the scale factor. Limitation of this approach is that the road does not ever contain useful 2D information and can be largely occluded (by cars, *etc*.) so that the camera motion estimation can fail. In a recent work, Scaramuzza *et al*. [13, 12] propose a novel approach for scale factor computation which is based on nonholonomic constraints and which does not affect the camera motion estimation. They demonstrate that the resulting constraints on car displacement and the knowledge of the offset between the camera and the vehicle's origin allow the computation of a global SLAM scale factor. Nevertheless, this method is only efficient in some turning where the nonholonomic constraint can be correctly observed. In their context, the scale factor drift is very limited thanks to the use of omnidirectional camera. Therefore, they can compute this scale factor only a few times for the entire sequence. However, when using a perspective camera, the scale factor must be estimated more frequently.

The approach we propose is linked to those two approaches. We have seen that in our case, the camera motion is computed with a bundle-adjustment-based monocular SLAM method [10]. This kind of algorithms is robust, fast and provides a good estimation of the camera motion. In this case, the homography undergone by the road plane can be expressed as a 1-DOF (Degree Of Freedom) problem, the only parameter being the scale factor. This new homography parameterisation can be used to estimate robustly and efficiently this scale factor.

In the following, we will introduce the equation that links plane motion and homography. Then, we will show how this problem can be reduced to a 1-DOF problem in our context. Finally, we will present an algorithm to solve this problem in a fast and robust way.

2.2. Homography and Planar Motion

Figure 1 shows the general case where the relation between camera poses and observed plane homography is well-known [14]. The two camera poses are $(\mathsf{R}_1, \mathbf{t}_1)$ and

$(\mathsf{R}_2, \mathbf{t}_2)$, that is to say that the transformation between a 3D point expressed in the world coordinate frame $(\mathcal{Q}_\mathcal{W})$ and in the camera one $(\mathcal{Q}_{\mathcal{C}_i})$ is $\mathcal{Q}_\mathcal{W} = \mathsf{R}_i \mathcal{Q}_{\mathcal{C}_i} + \mathbf{t}_i$.

If we note $(\mathsf{R}_{1\to2}, \mathbf{t}_{1\to2})$ the transformation between the two cameras, $\mathbf{n}$ the plane normal expressed in $\mathcal{C}_1$ coordinate system and d the distance between $\mathcal{C}_1$ and the plane, the coordinate frame transformation of a 3D point $\mathcal{Q}$ lying on the plane is

$$\mathcal{Q}_{\mathcal{C}_2} = \left(\mathsf{R}_{1\to2} - \frac{\mathbf{t}_{1\to2}\mathbf{n}^t}{d}\right)\mathcal{Q}_{\mathcal{C}_1} \tag{1}$$

and finally the 2D observations are linked by the homography $\mathcal{H}$, associated to the 3×3 matrix H:

$$\begin{aligned} \mathbf{q}_{\mathcal{C}_2} &= \mathsf{K}\left(\mathsf{R}_{1\to2} - \frac{\mathbf{t}_{1\to2}\mathbf{n}^t}{d}\right)\mathsf{K}^{-1}\mathbf{q}_{\mathcal{C}_1} \\ &= \mathsf{H}\mathbf{q}_{\mathcal{C}_1} \end{aligned} \tag{2}$$

where K is the camera calibration matrix.

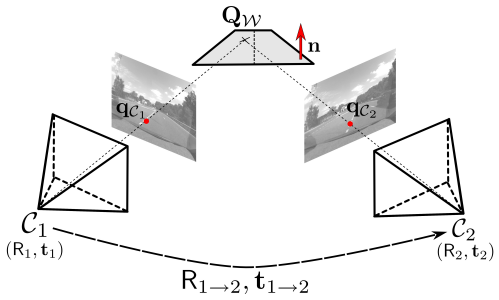


Figure 1. **Homography and Planar Motion.** The two observations of a 3D point lying on a plane are linked by homography defined thanks to the camera displacement and the plane parameters.

2.3. Scale factor estimation as an 1-parameter homography problem

In our case, the relative displacement of the camera is estimated up to a scale factor (*i.e.* $\mathsf{R}_{1\to2}$ and $\frac{\mathbf{t}_{1\to2}}{||\mathbf{t}_{1\to2}||}$ are known) thanks to the monocular SLAM method proposed in [10]. Besides, the normal $\mathbf{n}$ and distance d can reasonably be considered as known constant values. Thus, points lying on the ground plane are linked by the homography $\mathcal{H}(\lambda)$ where λ is the only unknown parameter:

$$\mathsf{H}(\lambda) = \mathsf{R}_2^t\mathsf{R}_1 + \lambda\frac{\mathsf{R}_2^t(\mathbf{t}_2 - \mathbf{t}_1)\mathbf{n}^t}{d} \tag{3}$$

So, the scale factor estimation can be expressed as the search of the λ value which minimizes the transfer error linked to $\mathcal{H}(\lambda)$. Optimising only the scale factor λ is not optimal in that $(\mathbf{n}, d, \mathsf{R}_{1\to2}, \mathbf{t}_{1\to2})$ values may be not perfect. However, it appears experimentally that refining all those parameters causes important convergence problem in our context, in consequence of bad interest points distribution, bad detection, dense traffic, *etc*. That is why we have decided to optimise only the λ parameter. We will now show how to robustly estimate this parameter.

2.4. Scale Factor Estimation

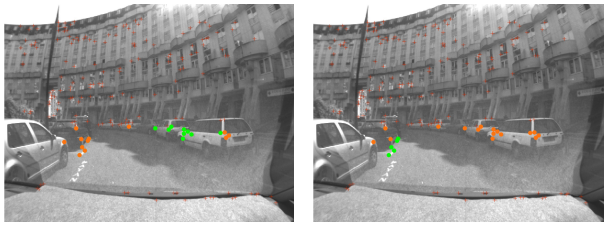
Three main steps are necessary in order to estimate precisely and robustly λ: find the couples of 2D interest points lying on the ground plane, obtain a first estimation of λ and then refine this value thanks to a robust non-linear optimisation.

2.4.1 Ground Points Identification

This process can be split into two successive tasks, that is to say finding matches between features points of each image and identifying, among those matches, those which correspond to points lying on the ground. We can notice that the first task is already realized by SLAM process. Therefore, we will focus on the second step of the process.

A usual approach in ground points filtering consists in exploiting the road planarity. The classical way consists in using a robust homography estimation thanks to the RANSAC paradigm [14]. This solution relies on the hypothesis that the larger plane observed in the images is the floor. Nevertheless, the road is often poorly textured. Moreover, in dense traffic condition, the larger observed plane may be the chest or the door of a car (see figure 2). Therefore, this approach may lead to a large amount of scale factor misestimation.

To prevent this problem, we prefer to simply filter points from their 3D position. Because the ground/camera distance is a constant, ground points can be identified from their vertical distance to the camera. To handle scale factor drift induced by the monocular SLAM process, a tolerance on this point/camera distance is introduced. In the following



(a) (b)

Figure 2. **Ground point selection.** (a) is the result obtained with classical largest homography support method. (b) is the result of the proposed method. Red crosses are interest points eliminated because over the skyline, orange points are candidates rejected by the respective used criterion. Green points are the final 2D interest points detected to be on the floor.

experiments, this tolerance is fixed to 15 cm. This solution relies on the hypothesis that our method is able to correct the scale factor often enough to avoid a too important drift. As we will see, the following experiments tend to confirm this assumption. Otherwise, we could imagine, for example, to use the method proposed in [12] in order to reestimate the global scale factor during turnings.

Even if we do the hypothesis that the λ propagated by the SLAM is almost correct, experiments show that this value is often too far from reality to allow the correct convergence of a non-linear λ optimisation. This is due in particular to false ground interest points detection. Thus, we will first show how to obtain a linear approximation of λ and then how to refine it.

2.4.2 Scale Factor Computation

Once the m couples of 2D observations lying on the road are detected, a coarse value of λ can be estimated. Equation (2) is equivalent to say that the two vectors $\mathbf{q}_{\mathcal{C}_2}$ and $\mathsf{H}\mathbf{q}_{\mathcal{C}_1}$ are collinear and thus their cross product is null: $\mathbf{q}_{\mathcal{C}_2} \times \mathsf{H}\mathbf{q}_{\mathcal{C}_1} = 0$. By developing this equality for the set of m couples, we can easily deduce that $\lambda\mathsf{A} = \mathsf{B}$ where A and B are two $(3 \times m)$ vectors. The linear least-squares solution of this equation is then

$$\lambda = \mathsf{A}^{+}\mathsf{B} \tag{4}$$

where $\mathsf{A}^{+} = (\mathsf{A}^t\mathsf{A})^{-1}\mathsf{A}^t$ is the pseudoinverse of A.

Then, to reach a fine estimation of λ, its value can be optimized with a non-linear optimisation process. A Levenberg-Marquardt algorithm [17] is used to optimise λ with respect to the symmetric transfer error of the matches:

$$\mathcal{F}(\lambda) = \sum_i \rho(||\mathbf{q}^i_{\mathcal{C}_2} - \mathsf{H}(\lambda)\mathbf{q}^i_{\mathcal{C}_1}||^2 + ||\mathbf{q}^i_{\mathcal{C}_1} - \mathsf{H}(\lambda)^{-1}\mathbf{q}^i_{\mathcal{C}_2}||^2) \tag{5}$$

where $(\mathbf{q}^i_{\mathcal{C}_j})_i$ are the 2D interest point set on the road previously detected. ρ is the Tukey M-Estimator [17] and its threshold is based on Median Absolute Deviation (MAD).

2.4.3 Validation of Scale Factor Estimation

As you can see in figure 2, only a few 2D points may be on the ground plane (sometimes none). Thus, an *a posteriori* validation of lambda value must be realized. To be validated, the estimated λ value must allow a correct transfer error (*i.e.* below 2 pixels) for a sufficient set of ground interest points. We arbitrarily fix this threshold to 4. If λ does not respect this constraint, the scale factor propagated by the SLAM process is kept unchanged. If λ is valid, let's show how it is used to correct the SLAM reconstruction.

2.5. Scale Factor Integration

In the following, we note λ_i the scale factor computed between cameras $\mathcal{C}_i$ and $\mathcal{C}_{i+1}$. Observe that λ_i may not be estimated for each i value, because its *a posteriori* validation may fail (section 2.4.3). In our study case (figure 3), the scale factor has been successfully computed between $\mathcal{C}_i$ and $\mathcal{C}_{i+1}$ and between $\mathcal{C}_j$ and $\mathcal{C}_{j+1}$ but not for the camera couples between $\mathcal{C}_{i+1}$ and $\mathcal{C}_j$.

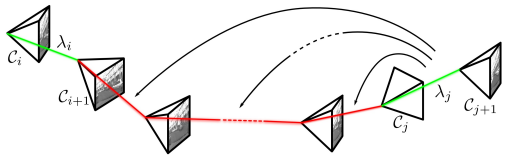


Figure 3. **Integration of scale factor in monocular SLAM.** To fix the scale λ_j between the 2 last cameras, we propose to modify the camera position history.

While monocular SLAM process provides a possible position for the current camera $\mathcal{C}_{j+1}$, the estimated λ_j implies another camera position prediction. To obtain it, it is for example possible to reestimate inter-cameras distances for camera couples between $\mathcal{C}_{i+1}$ and $\mathcal{C}_{j+1}$ with λ_j (figure 3). To fuse two camera position predictions, the more classical approach is to use a Kalman filter [3]. Nevertheless, even if Eudes *et al.* [4] have recently proposed a method to compute the covariance of local bundle adjustment, it does not take into account the potential scale factor drift between $\mathcal{C}_{i+1}$ and $\mathcal{C}_{j+1}$. Then the two predicted camera positions and their associated covariances may not be consistent. In this case, the Kalman filter may fail [8].

Thus, since the scale factor drift is uncontrolled in our study case, it is not possible to obtain trustworthy camera position information from the monocular SLAM process. Thus, the confidence can only be given to the position predicted with the estimation of λ_j, as done in [12] with the odometer. To refine the obtained geometry (*i.e.* the camera trajectory between $\mathcal{C}_{i+1}$ and $\mathcal{C}_{j+1}$ and the associated 3D point cloud), it should be necessary to apply a full bundle adjustment on the predicted geometry. However, because of real-time purpose, such a solution is not conceivable.

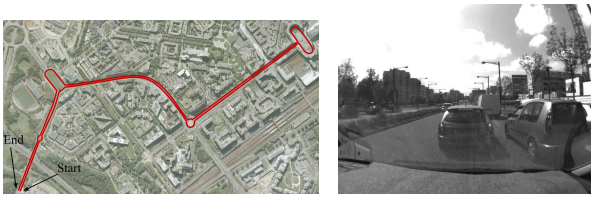
The solution we propose is to improve the prediction quality by analyzing scale factor drift behaviour. In fact, experiments (*e.g.* figure 6(b)) illustrate that the scale factor drift can often locally be considered as linear. So, the scale factors we apply to the history is no more λ_j but a linear approximation between λ_i and λ_j. Furthermore, 3D points observed by cameras $\mathcal{C}_{i+1}$ to $\mathcal{C}_{j+1}$ are triangulated thanks to the newly computed camera positions (which is a very fast process). It ensures that the scale factor propagated by monocular SLAM after the camera $\mathcal{C}_{j+1}$ will be correct. Numerical experiments (section 2.6) show that, while being

real-time, this solution supplies accurate camera positioning results.

In the following section, we will show that the full proposed scale factor correction process (*i.e.* its estimation and integration) is fully efficient on large-scale realistic sequences.

2.6. Experimental Validation

The scale factor estimation and integration have been successfully tested on a 4.5 kilometres long sequence (figure 4(a)). This sequence has been realised in real traffic condition (figures 2 and 4(b)) with a classical 640×480 perspective camera. In the following, we will consider the localisation provided by a trajectometer as ground truth.



(a) (b)

Figure 4. **Experimental sequence.** A 4.5 kilometres long sequence in real traffic condition.

The reconstruction obtained with Mouragnon *et al*. [10] monocular SLAM algorithm can be found in figure 5. 1296 keyframes and 39304 points have been created over the 4.5 kilometres. Thus, in average, a keyframe is created every 3.4 metres. Comparing the obtained reconstruction with ground truth highlights the scale factor drift. Note that, since no scale factor is provided by this method, a global scale factor has been fixed on the first cameras thanks to ground truth data for comparison purpose. The drift phenomenon is numerically confirmed on figure 6(b) which represents the percentage of error of successive inter-cameras distances between the obtained reconstruction and the ground truth. Observe that this figure must be compared to figure 6(a) which represents the same error but in meters.

For this sequence, normal $\mathbf{n}$ and distance d ground truth was unknown. Thus, note that the used values are very rough. The obtained result is visually displayed in figure 5 and confirmed on figure 6. We can observe that our method allows avoiding the scale factor drift. On this sequence, our method succeeds in scale factor computation for 55% of the cameras couples. That is to say that the scale factor can be computed every 6.2 metres in the mean. Nevertheless, note that those estimations are not well distributed on the entire sequence because of traffic, lack of texture on the ground, *etc*. The mean obtained inter-cameras distance error is 6.81% (*i.e*. 0.21m) with a standard deviation of 5.84% (*i.e*. 0.20m).

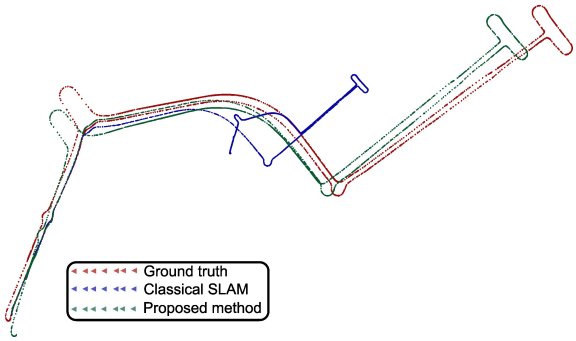
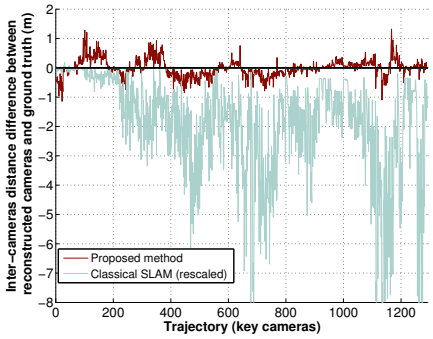
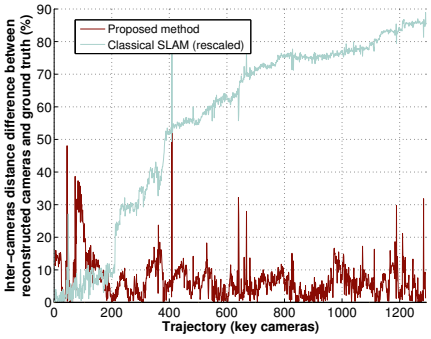


Figure 5. **Reconstructions comparison.** The proposed method (in green) prevents the scale factor drift of classical monocular SLAM process (in blue).



(a)



(b)

Figure 6. **Numerical comparison.** This figure represents the inter-cameras distance error for corresponding reconstructed cameras couples and ground truth couples in meters (a) and in percentages (b).

Nevertheless, because of residual error accumulation of incremental SLAM process, the absolute camera distance between the obtained reconstruction and the ground truth grows drastically (figure 5). In the following, we propose to tackle this problem for dense urban areas.

3. Towards Global Localisation Using 3D City Models

The goal of this section is to show that the positioning error accumulation in dense urban area can be efficiently corrected by using a coarse georeferenced 3D model of the environment, once the scale factor drift is corrected.

3.1. Related Works

In the car case, the more classical used absolute sensor is the GPS system [1]. Nevertheless, the precision of this sensor may be low, in particular in dense area because of signal occlusion by buildings. Another approach consists in using georeferenced 3D city models. Sourimant *et al*. [16] propose to use them in an inline localisation process. In order to compute a georeferenced 3D point cloud, they do not use multiview geometry to estimate the 3D position of feature points, but the intersection of their backprojections with the 3D model. For this solution, their work hypothesis is that the 3D model is perfect and that no object obscures it. Otherwise, the camera localisation may highly drift.

The solution we propose is based on our previous work [6] which exploits coarse 3D city models only composed of vertical planes representing the mean planes of building fronts and with a precision of about 2 meters, as those provided by GIS database. The idea of our previous work was to fit an entire SLAM reconstruction with this model in a posterior process to correct both scale factor drift and error accumulation. This solution is efficient because the use of the entire SLAM reconstruction brings many geometric constraints to the problem.

However, this solution must be adapted for the on-line localisation problem since the constraints provided by the knowledge of the entire sequence are lost in this case. Indeed, while the vehicle is moving in a straight line, constraints provided by the model are reduced to the front of the two sides of the street. Therefore, the model no longer provides constraints on the displacement along the street axis. Moreover, the scale factor is only constrained by the distance that separates these two walls. Since the uncertainty of the wall position can be important (about 2 meters) with respect to the street width, the resulting scale factor might be quite inaccurate.

Nevertheless, in some configurations, 3D city models can provide enough constraints to estimate a global localisation. In fact, turnings in the camera trajectory provide such a configuration and are easy to detect. Therefore, in the following, we will introduce a solution that uses a 3D city model to estimate a global localisation after a turning. First, we will briefly expose a solution to detect turnings in trajectory. Then, we will expose how the 3D city model is exploited to reach a global localisation. And finally, an experimental evaluation of the process will be provided.

3.2. Integrating 3D City Models

In this section, we will explain how to detect the cases which are in favour of camera position correction. Then we will describe how the camera position is robustly corrected thanks to the 3D model.

3.2.1 Turning Detection

The turnings detection relies on a polygonalisation of the camera trajectory. For each new key frame, the entire trajectory is approximated by a polyline [7]. The rise of the number of polyline segments signals a turning. Then, cameras and 3D points can be split into two groups: before and after turnings.

3.2.2 3D Point Cloud - 3D Model Fitting

ICP methods [11] are designed to fit two 2D or 3D models, in our case the 3D SLAM reconstruction and the 3D city model. ICP are composed of two specific steps which are iterated: the data association and the computation of the transformation between the two models for this association. In the following, we will describe those two steps for our problem.

Data Association. The goal of data association is to match each entity of the first model with an entity of the second one. In our case, each reconstructed 3D point $\mathcal{Q}_i$ must be associated with its corresponding plane in the 3D city model $\mathcal{M}$. Since this plane is unknown, the usual approach consists in selecting the nearest one [6]. To improve the robustness, we had the constraint that the plane and the point must share a similar surface normal orientation. The surface normal of a point is estimated with Molton *et al*. [9] method. Once the normal is computed for $\mathcal{Q}_i$, it can be associated to its nearest plane among those of $\mathcal{M}^*$, *i.e*. the subset of planes whose normal is consistent with the one associated to $\mathcal{Q}_i$:

$$\forall \mathcal{Q}_i, \mathbf{\Pi}_{h_i} = \underset{\mathbf{\Pi} \in \mathcal{M}^*}{argmin}\ d(\mathcal{Q}_i, \mathbf{\Pi}) \tag{6}$$

where d is the normal distance. Notice that the distance d takes into account that the planes are finite: to be associated to a plane $\mathbf{\Pi}$, a 3D point $\mathcal{Q}_i$ must have its normal projection inside $\mathbf{\Pi}$ bounds. The hypothesis done in the data association step is that the SLAM reconstruction is not too far from its real position, in particular to avoid confusing two successive possible streets. The following experiments will confirm that such assumption is generally correct.

The matching step is then validated if enough points are matched before and after the turning. Otherwise, no transformation is computed. In the following experiments, a minimum of 150 points is required.

Transformation Computation. The goal of the transformation $\mathcal{T}$ we are looking for is to correct the orientation and the position of the last keyframes. Because the altitude of the camera is known and that the only rotation we want to correct is around the vertical axis (in the world coordinate frame), $\mathcal{T}$ has only 3 DOF. Observe that in theory, the scale factor could be reestimated by using the street width between buildings. Nevertheless, because of the used 3D model low precision (about 2 meters for each building front positioning), experiments showed us that this new estimated scale factor would be very inaccurate. Thus we decide not to call the scale factor into question.

The problem we want to solve is then:

$$\min_{\mathcal{T}} \sum_i \rho(d(\mathcal{T}(\mathcal{Q}_i), \mathbf{\Pi}_{h_i})). \tag{7}$$

where $\mathcal{T}(\mathcal{Q}_i)$ is the 3D point $\mathcal{Q}_i$ transformed by $\mathcal{T}$. The Tukey M-Estimator [5] ρ is used in order to be robust to residual point-plane bad association. 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. Even if this assumption could be done for each set of points associated to the same plane, it is not true for the whole reconstruction. So we propose to use a different M-estimator threshold ξ_{h_i} per model plane. This also implies that we have to normalize the Tukey values on each fragment. For $\mathcal{Q}_i$ associated to $\mathbf{\Pi}_{h_i}$:

$$\rho'_{h_i}(d(\mathcal{Q}_i, \mathbf{\Pi}_{h_i})) = \frac{\rho_{h_i}(d(\mathcal{Q}_i, \mathbf{\Pi}_{h_i}))}{\max\limits_{\mathcal{Q}_j \in \mathcal{S}_{h_i}} \rho_{h_i}(d(\mathcal{Q}_j, \mathbf{\Pi}_{h_i}))} \tag{8}$$

where $\mathcal{S}_{h_i}$ is the set of points associated to $\mathbf{\Pi}_{h_i}$ and ρ_{h_i} is the Tukey M-estimator used with the threshold ξ_{h_i} for the 3D point in $\mathcal{S}_{h_i}$.

With the cost function (8), each model plane will have a weight in the minimization proportional to the number of its associated 3D points. Then, plane with few points could be not optimized in favour of the others. To give the same weight to each plane, we must unify all the Tukey values of their 3D points with respect to their cardinal:

$$\rho^*_{h_i}(d(\mathcal{Q}_i, \mathbf{\Pi}_{h_i})) = \frac{\rho'_{h_i}(d(\mathcal{Q}_i, \mathbf{\Pi}_{h_i}))}{card(\mathcal{S}_{h_i})} \tag{9}$$

and the final minimization problem is:

$$\min_{\mathcal{T}} \sum_i \rho^*_{h_i}(d(\mathcal{T}(\mathcal{Q}_i), \mathbf{\Pi}_{h_i})). \tag{10}$$

that we solve using the Levenberg-Marquardt algorithm [17].

In the following we will show that the entire proposed method (*i.e.* the scale factor correction and 3D city models fitting) allows getting a consistent global position of a vehicle in dense urban area.

3.3. Experimental Results

In order to have 3D model information, we have tested the proposed method on the data used in our previous work [6]. Consequently, note that no ground truth is available for this sequence. The entire proposed process has been tested on a 1 kilometre long sequence (figure 8(b)) with a classical 640×480 perspective camera (see figure 8(a)). As in section 2.6), $\mathbf{n}$ and d have been roughly estimated.

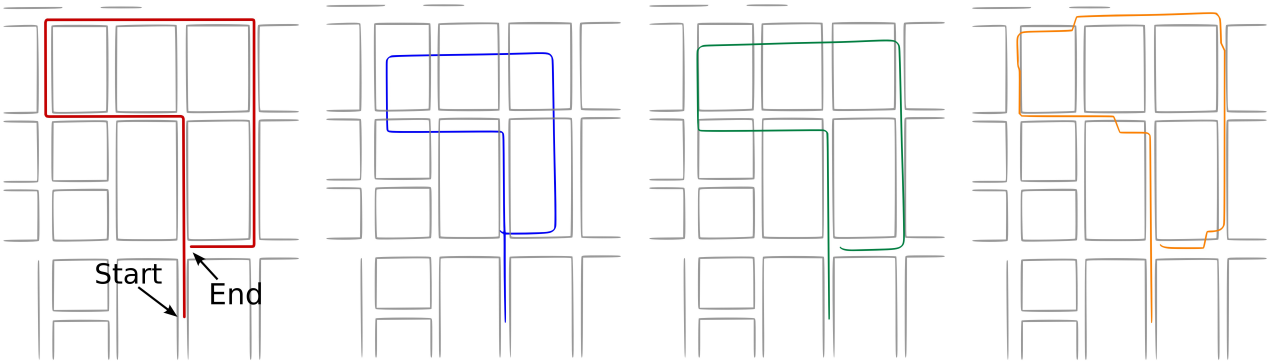
(a) (b)

Figure 8. **Dense urban sequence.** An example of video frame (a) and the followed trajectory (b).

Figure 7 compares the obtained camera trajectory (*i.e.* the estimated position of the camera for each key frame) for 3 different reconstructions: the classical monocular SLAM method proposed by Mouragnon *et al.*, the reconstruction obtained with the proposed scale factor correction method and finally the result of the entire proposed method, *i.e.* the scale factor correction and the use of coarse 3D model. Note that for comparison purpose, the 3 reconstructions have been initialized with the same initial camera position, orientation and scale factor. The reconstruction contains 13276 points and 356 cameras (*i.e.* a keyframe every 2.8 metres in the mean).

This figure highlights the drift of classical SLAM process. We can see that the scale factor correction (successful for 45% of couples in this sequence, *i.e.* every 6.2 metres) prevents the scale factor drift, which is particularly visible on the width of the obtained reconstruction. Nevertheless, it also highlights the residual error accumulation. The full proposed method reconstruction shows the successful use of the associated 3D model to correct this error accumulation: the discontinuity of camera trajectory after each turning highlights those corrections when enough geometrical information was available. Observe that the ratio between the robustness and the reactivity of the fitting after each turning can be easily tuned with the number of point-plane association threshold (section 3.2.2).

The final obtained reconstruction shows that our entire method can be successfully used as an alternative for navigation system in dense urban area.



(a) (b) (c) (d)

Figure 7. **Dense urban sequence results.** (a) is the ground truth, (b) the original monocular SLAM reconstruction, (c) is obtained thanks to our scale factor process and (d) is the resulting reconstruction with our entire method.

4. Conclusion

In this paper, we have proposed an efficient system to localise a vehicle in a dense urban area by using a single camera and a coarse 3D model of the environment. First, we have shown that well-known scale factor drift of monocular SLAM process can be avoided. In fact, specific constraints allow us to express the road homography estimation as a problem with the single scale factor parameter. Then, this paper highlights that residual unavoidable error accumulation in SLAM process can be corrected by using the global information brought by coarse 3D city model.

Our experiments on large scale sequences show that the scale factor can be robustly estimated and that the full proposed process succeeds in positioning, in real-time, a car cruising in a dense city centre. Future work would include an enhancement of ground point detection and a global scale factor computation method. Then it would be interesting to substitute the 3D model by a road map in order to obtain a full working navigation system even outside dense cities.

References

[1] M. Agrawal and K. Konolige. Real-time localization in outdoor environments using stereo vision and inexpensive gps. In *ICPR*, pages 1063–1068, 2006.

[2] I. Arnold, C. Zach, J.-M. Frahm, and B. Horst. From structure-from-motion point clouds to fast location recognition. In *CVPR*, pages 2599–2606, 2009.

[3] J. Civera, O. G. Grasa, A. J. Davison, and J. M. M. Montiel. 1-point ransac for EKF-based structure from motion. In *IROS*, 2009. To appear.

[4] A. Eudes and M. Lhuillier. Error propagations for local bundle adjustment. In *CVPR*, pages 2411–2418, 2009.

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

[6] P. Lothe, S. Bourgeois, F. Dekeyser, E. Royer, and M. Dhome. Towards geographical referencing of monocular slam reconstruction using 3d city models: Application to real-time accurate vision-based localization. In *CVPR*, pages 2882–2889, June 2009.

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

[8] R. Mittu and F. Segaria. Common operational picture and common tactical picture management via a consistent networked information stream. In *Proceedings of the Command and Control Research and Technology Symposium*, 2000.

[9] N. Molton, A. Davison, and I. Reid. Locally planar patch features for real-time structure from motion. In *BMVC*, 2004.

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

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

[12] D. Scaramuzza, F. Fraundorfer, M. Pollefeys, and R. Siegwart. Absolute scale in structure from motion from a single vehicle mounted camera by exploiting nonholonomic constraints. In *ICCV*, 2009.

[13] D. Scaramuzza, F. Fraundorfer, and R. Siegwart. Real-time monocular visual odometry for on-road vehicles with 1-point ransac. In *ICRA*, 2009.

[14] D. Scaramuzza and R. Siegwart. Appearance guided monocular omnidirectional visual odometry for outdoor ground vehicles. *IEEE Transactions on Robotics, Special Issue on Visual SLAM. In press.*, 2008.

[15] N. Simond and P. Rives. Trajectography of an uncalibrated stereo rig in urban environments. In *IROS*, pages 3381–3386, 2004.

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

[17] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. Bundle adjustment - a modern synthesis. In *ICCV*, pages 298–372, 2000.