Image and Vision Computing 32 (2014) 54–63

Contents lists available at ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

Direct model based visual tracking and pose estimation using mutual information[☆]

Guillaume Caron [a,b,*], Amaury Dame [a,c], Eric Marchand [a]

[a] INRIA Rennes/IRISA, Lagadic, Rennes, France
[b] Université de Picardie Jules Verne, MIS Laboratory, Amiens, France
[c] University of Oxford, Active Vision Group, Oxford, United Kingdom

ARTICLE INFO

Article history:
Received 19 September 2012
Received in revised form 23 July 2013
Accepted 30 October 2013

Keywords:
Tracking
3D pose estimation
Mutual information

ABSTRACT

This paper deals with model-based pose estimation (or camera localization). We propose a direct approach that takes into account the image as a whole. For this, we consider a similarity measure, the mutual information. Mutual information is a measure of the quantity of information shared by two signals (or two images in our case). Exploiting this measure allows our method to deal with different image modalities (real and synthetic). Furthermore, it handles occlusions and illumination changes. Results with synthetic (benchmark) and real image sequences, with static or mobile camera, demonstrate the robustness of the method and its ability to produce stable and precise pose estimations.

1. Introduction

Camera tracking and pose estimation are critical for robotic applications such as localization, positioning tasks or navigation. The use of a monocular vision sensor in these contexts is full of potential since images bring very rich information on the environment. The problem of camera pose estimation is equivalent to camera localization. We aim to design a new camera pose estimation method that uses the mutual information as a visual feature.

Camera localization has received much interest in the last few years. Visual Simultaneous Localization And Mapping [1–3] or, in the computer vision community, Structure From Motion with bundle adjustment optimization [4,5] are common ways of estimating the camera pose, or relative pose. These approaches reconstruct the environment and estimate the camera position simultaneously but are prone to drift (although a loop can be detected). Visual odometry is another way to retrieve the relative pose of the camera [6] but estimations drift irremediably.

However, if a 3D model on the environment is already available to the robot, drift, exploration and loop closure issues can be withdrawn. In [7], it has been shown that the use of 3D information on the environment ensures a better precision in pose estimation. This 3D information allows the camera pose estimation, when a unique camera is embedded on a mobile platform, precise and without drifting, if the robot moves near referenced [8], or even georeferenced [9] landmarks. This is ensured since pose estimation is essentially a mono image problem.

For a few years, 3D models of cities or urban environments have been made available through various digitized town projects over the world. The French National Institute of Geography (IGN) digitalized streets and buildings of the XIIth arrondissement of Paris in France (Fig. 1(a)). Hence, we aim to exploit this textured 3D model to localize a vehicle using vision, i.e. to estimate the pose of the camera in the real scene merging the information brought by the real image (Fig. 1(b)) and the virtual world (Fig. 1(c)) in a multi-modality scheme.

Model-based pose estimation is a problem tackled since several years working with various feature types: points [10,11], lines [12], both [13] or wireframe models [14–16]. These works dealt with geometrical features but only a few other works take into account the photometric information explicitly in the pose estimation and tracking. Some of them mix geometric and photometric features [17,18]. Photometric features (image intensity) can directly be considered to estimate the homography and then the relative position between a current and a reference image [19]. A more recent approach proposes to estimate such transformation using information theoretic approaches. Mutual information shared by a planar textured model and images acquired by the camera are used to estimate an affine transformation [20], or a homography [21].

The contribution of this paper is to generalize the latter work to general 3D models defined by a textured mesh, since this is a common way in computer vision or computer graphics, to represent a virtual scene. Hence, this work formulates the pose optimization problem as the maximization of the mutual information shared by a real image and a virtual view rendered from a given pose.

Even if we try to maximize the mutual information between the current image and a model of the scene projected or transferred in this image, as in [21], the model of the scene in the latter work is just a

[☆] This paper has been recommended for acceptance by Aleix M. Martinez.
[*] Corresponding author at: 33 rue Saint-Leu, 80039 Amiens Cedex, France.
E-mail addresses: guillaume.caron@u-picardie.fr (G. Caron), adame@robots.ox.ac.uk (A. Dame), eric.marchand@irisa.fr (E. Marchand).

0262-8856/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.imavis.2013.10.007

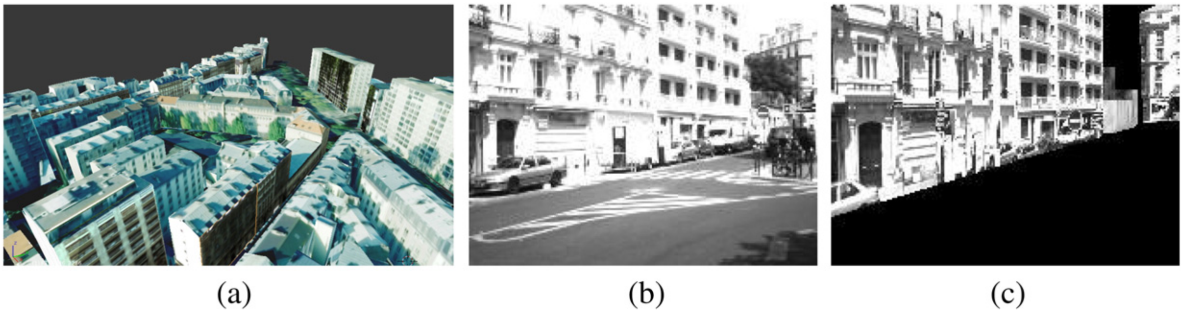


Fig. 1. (a) The textured 3D model of the XIIth arrondissement of Paris, (b) real image acquired in a street and (c) its corresponding synthetic view.

reference template that is warped within the current one in order to estimate the parameters of a homography. Therefore, with the mutual information as a similarity function, the optimization process presented in [21] is inspired from the inverse compositional approach introduced by Baker and Matthew [19] (a sequel of the famous KLT) and adapted to consider the specificities of the mutual information. Parameters to be estimated in [21] belong to sl(3), the special linear group of dimension 3.

In this paper, as in [21], we try to maximize the mutual information between the current image and a model of the scene. Theoretical background is then indeed very similar. Nevertheless, in our case the model of the scene is a 3D model that is projected in the image according to the current estimated camera pose (using the GPU). In this paper, the parameters to be estimated are then a pose (that belongs to se(3), the special Euclidian group). Optimization space and optimization techniques are then very different.

The proposed method for pose estimation using a virtual reference scene is close to the work of Dame and Marchand [22], where a real camera is moved to a desired pose in a visual servoing control law, except that:

- in our case, the camera is virtually moved to its optimal pose, corresponding to the real image whereas in [22] they physically move the camera using a robot and real images only.
- in [22], this last real desired image is furthermore acquired by the same camera as the one that is used to control the robot and, thus, no camera geometric or photometric difference exists between them, contrary to the real/virtual camera case as ours. This emphasizes the importance of using a robust similarity measure such as MI.
- [22] use only a 2D image as reference and consider a fronto-parallel desired planar scene whereas the current paper deals with any scene structure.

Despite these differences, some theoretical aspects of the current paper are shared with [22], but differences and asset are highlighted in next sections.

The remainder of the paper is organized in three main parts. First, the general formulation of the model based on visual pose estimation as a non linear optimization problem is introduced in Section 2. Then, in Section 3 the maximization of the mutual information to optimize the pose is detailed. Finally, results are presented, in Section 4, the behavior of the proposed pose estimation method, its precision and its robustness, before the conclusion.

2. Pose estimation: problem definition

Pose estimation is considered in this work as a full-scale non linear optimization problem. Hence, for a new image, the pose is computed by minimizing the error between measurements in the image and the projection of a 3D model of the scene for a given pose. Since camera motion between two images is assumed to be small the pose obtained for the previous image is a good initial guess for the pose of the new image. The initialization problem is only encountered for the first image acquired by the camera. This issue is more of a detection, matching and recognition problem and is out of the scope of this paper, even if an obvious solution is mentioned in the last experiment (Section 4.2: GPS initial guess at the entrance of city, for the localization experiment).

2.1. Feature based pose estimation

Visual pose estimation has mostly been known through feature based approaches. Considering $\mathbf{r}$ is a vector representation of the three translations and three rotations pose ($\mathbf{r} = [t_X, t_Y, t_Z, \theta_X, \theta_Y, \theta_Z]$), the camera pose $\mathbf{r}^*$ must satisfy some properties measured in the images. Considering $\mathbf{s}(\mathbf{r})$, the projection of 3D scene features for the pose $\mathbf{r}$, the camera pose $\mathbf{r}^*$ is the pose ensuring that the error between $\mathbf{s}(\mathbf{r})$ and $\mathbf{s}^*$ (observation in the image) is minimal. The optimization problem can thus be written:

$$\hat{\mathbf{r}} = \arg\min_{\mathbf{r}} \|\mathbf{s}(\mathbf{r}) - \mathbf{s}^*\|. \quad (1)$$

The 3D model is classically made with geometrical features such as point [23] and line [14]. In that case, the main issue is to determine in each frame the correspondences between the projection of the model and features extracted from the image $\mathbf{s}^*$ and to track them over frames.

Errors or imprecision in the low level tracking leads to important error in the tracking and pose estimation process.

2.2. Direct pose estimation

To avoid these geometrical features tracking and matching issues, and also the loss of precision that these approaches introduce, other formulations that use images as a whole need to be proposed. It has to be noted that such direct approach has been widely considered for 2D tracking or motion estimation [19]. In such approach the idea is directly to minimize the error, the sum of squared differences (the SSD), between an image template $\mathbf{I}^*$ and the current image $\mathbf{I}$ transferred in the template space using a given motion model (usually a homography).

Theoretically, assuming that a 3D model of the scene is available, this process can scale to the pose estimation process. Indeed, in that case, the pose can be determined by minimizing the error between the image acquired by the camera $\mathbf{I}^*$ and the projection of the scene for a given pose $\mathbf{I}(hbfr)$. The cost function could be written as:

$$\hat{\mathbf{r}} = \arg\min_{\mathbf{r}} \sum_{\mathbf{x}} \left(\mathbf{I}(\mathbf{r}, \mathbf{x}) - \mathbf{I}^*(\mathbf{x})\right)^2. \quad (2)$$

In Eq. (2), $\mathbf{I}(\mathbf{r},\mathbf{x})$ can be obtained using a rendering engine. The latter virtual model, even mapped with photorealistic textures is rendered through any 3D engine (such as openGL) and the obtained image is nothing but a synthetic image. Hence, even if the cost function of Eq. (2) is free from geometric feature tracking or matching, illumination variation or occlusions highly affect the cost function causing the visual tracking to fail.

We propose to formulate another optimization criterion directly comparing the whole current and desired images. Rather than using a difference based cost function as the SSD, we define an alignment function between both images as the Mutual Information (MI) between $\mathbf{I}(\mathbf{r})$ and $\mathbf{I}^*$ [24,25]. MI is a measure of the quantity of information shared by the two images [24]. When MI is maximal, then the two images are registered. The MI similarity measure has been used for registration works [25] and more recently to track planes in image sequences [21] and visual servoing [22]. This feature has shown to be robust to noise, specular reflections and even to different modalities between the reference image and the current one. The latter advantage is particularly interesting in our work since we want to align a synthetic view with a real image.

We then propose an extension of [21] to the case of non planar model based pose estimation and tracking. This extension adapts the use of MI over SL(3) to the SE(3) space. It means the parameter space is the full six 3D pose parameters (three translations and three rotations).

3. Mutual information on SE(3)

As stated in Section 2, more or less classical cost functions for pose estimation (Eqs. (1) and (2)) have to be reformulated. The goal is to perform the registration of the model with respect to the image and it can be formulated as the optimization of the mutual information shared between the input real image $\mathbf{I}^*$ and the projection of the model. If $\mathbf{r}$ is the pose of the calibrated camera, the pose estimation problem can be written as [26]:

$$\hat{\mathbf{r}} = \arg\max_{\mathbf{r}} \mathrm{MI}\left(\mathbf{I}^*, \mathbf{I}(\mathbf{r})\right). \tag{3}$$

Virtual image $\mathbf{I}(\mathbf{r})$ is resulting from the projection of the model at given pose $\mathbf{r}$.

3.1. Mutual information

MI is defined in [24] by the entropy $\mathbf{H}$ of images $\mathbf{I}$ and $\mathbf{I}^*$ and their joint entropy:

$$\mathrm{MI}\left(\mathbf{I}, \mathbf{I}^*\right) = \mathbf{H}(\mathbf{I}) + \mathbf{H}\left(\mathbf{I}^*\right) - \mathbf{H}\left(\mathbf{I}, \mathbf{I}^*\right). \tag{4}$$

Entropies $\mathbf{H}(\mathbf{I})$ and $\mathbf{H}(\mathbf{I}^*)$ and joint entropy $\mathbf{H}(\mathbf{I},\mathbf{I}^*)$ are a variability measure of a, resp. two, random variable $\mathbf{I}$, resp. $\mathbf{I}$ and $\mathbf{I}^*$. For $\mathbf{H}(\mathbf{I})$, if i are the possible values of $\mathbf{I}(\mathrm{x})$ ($i \in [0, N_c]$ with $N_c = 225$) and $p_\mathbf{I}(i) = Pr(\mathbf{I}(\mathbf{x}) = i)$ is the probability distribution function of i (obtained from image histogram), then the Shannon entropy $\mathbf{H}(\mathbf{I})$ of a discrete variable $\mathbf{I}$ is given by the expression:

$$\mathbf{H}(\mathbf{I}) = -\sum_{i=0}^{N_c} p_\mathbf{I}(i) \log\left(p_\mathbf{I}(i)\right). \tag{5}$$

In a similar way, we obtain the joint entropy expression:

$$\mathbf{H}\left(\mathbf{I}, \mathbf{I}^*\right) = -\sum_{i=0}^{N_c}\sum_{j=0}^{N_{c*}} p_{\mathbf{II}^*}(i,j) \log\left(p_{\mathbf{II}^*}(i,j)\right). \tag{6}$$

3.2. Mutual information based pose optimization

To determine the solution of Eq. (3), we consider a Newton's optimization method. To consider such an approach we have to exhibit the Jacobian and Hessian related to the mutual information. The Jacobian links the variation of the mutual information feature to the pose variation.

To solve Eq. (3) for the pose $\mathbf{r}$, a textured 3D model of the object to track is necessary and it has to be projected for each camera pose $\mathbf{r}$. To generate images of the 3D model, we used OpenGL as a 3D renderer and more particularly the Ogre3D library.[1] OpenGL allows to generate not only intensity images but also deepness images. More precisely, we obtain an image where each pixel contains the Z coordinate of the 3D point projected in this pixel. This is particularly interesting since the Z of each visible point appears in the Jacobian linking mutual information and pose variations as shown in Section 3.3.

Camera rotations and translations are highly correlated, as obviously X translation and Y rotation axes, for instance.

Hence, a simple steepest descent optimization approach using the direction given by the image Jacobian related to MI would not provide an accurate estimation of the optimum of MI. Therefore, a second order optimization approach as a Newton's like method is necessary.

Using a first order Taylor expansion of the MI similarity function at the current pose $\mathbf{r}_k$ in the non linear pose estimation gives:

$$\mathrm{MI}(\mathbf{r}_{k+1}) \approx \mathrm{MI}(\mathbf{r}_k) + \mathbf{L}_{\mathrm{MI}}^T \dot{\mathbf{r}} \Delta_t. \tag{7}$$

Δ_t is the period of time necessary to transform $\mathbf{r}_k$ into $\mathbf{r}_{k+1}$ using the pose variation$\dot{\mathbf{r}}$(which can be seen as the virtual camera velocity $\mathbf{v} = \dot{\mathbf{r}}$). The pose is updated thanks to $e^{[\mathbf{v}]}$, the exponential map on SE(3):

$$\mathbf{r}_{k+1} = e^{[\mathbf{v}]}\mathbf{r}_k. \tag{8}$$

$\mathbf{L}_{\mathrm{MI}}$ (Eq. (7)) is the image Jacobian related to MI, i.e. the Jacobian matrix linking the variation of MI and the pose variation. This leads to:

$$\mathbf{L}_{\mathrm{MI}}^T(\mathbf{r}_{k+1}) \approx \mathbf{L}_{\mathrm{MI}}^T(\mathbf{r}_k) + \mathbf{H}_{\mathrm{MI}}(\mathbf{r}_k)\mathbf{v}\Delta_t, \tag{9}$$

where $\mathbf{H}_{\mathrm{MI}}(\mathbf{r}_k)$ is the MI Hessian matrix. The goal is to maximize the MI so we want the system to reach the pose $\mathbf{r}_{k+1}$ where the variation of MI with respect to the pose variation is zero: $\mathbf{L}_{\mathrm{MI}}(\mathbf{r}_{k+1}) = 0$. Setting $\Delta_t = 1$ in Eq. (9), the approximated increment that leads to a null MI variation is:

$$\mathbf{v} = -\mathbf{H}_{\mathrm{MI}}^{-1}(\mathbf{r}_k)\mathbf{L}_{\mathrm{MI}}^T(\mathbf{r}_k). \tag{10}$$

As demonstrated in [22], in order to have a good estimation of the Hessian after convergence, rather than using the Hessian $\mathbf{H}_{\mathrm{MI}}^{-1}(\mathbf{r}_k)$, we use $\mathbf{H}_{\mathrm{MI}}^{*^{-1}}$ estimated at the desired position $\mathbf{r}^*$ $\left(\mathbf{H}_{\mathrm{MI}}^{*^{-1}} = \mathbf{H}_{\mathrm{MI}}^{-1}(\mathbf{r}^*)\right)$:

$$\mathbf{v} = -\mathbf{H}_{\mathrm{MI}}^{*^{-1}}\mathbf{L}_{\mathrm{MI}}^T. \tag{11}$$

$\mathbf{L}_{\mathrm{MI}}$ refers to the interaction matrix related to MI computed at current position $\mathbf{r}_k$. Of course, the optimal pose $\mathbf{r}^*$ is unknown but the Hessian matrix at the optimum $\mathbf{H}_{\mathrm{MI}}^{*^{-1}}$ can be estimated without knowing $\mathbf{r}^*$, considering Z = Z* (each image point has its own Z), since consecutive poses are close (see the end of Section 3.3). $\mathbf{L}_{\mathrm{MI}}$ is recomputed with current Z of each image point at each iteration of the optimization process.

3.3. Jacobian

Knowing entropy and joint entropy expressions (Eqs. (5) and (6)), MI (Eq. (4)) is developed as:

$$\mathrm{MI}\left(\mathbf{I}, \mathbf{I}^*\right) = \sum_{i,j} p_{\mathbf{II}^*}(i,j) log\left(\frac{p_{\mathbf{II}^*}(i,j)}{p_\mathbf{I}(i)p_{\mathbf{I}^*}(j)}\right). \tag{12}$$

[1] Ogre3D, Open Source 3D Graphics Engine, http://www.ogre3d.org.

From Eq. (12) and simplifications allowed by the chain derivation rule [27], the interaction $\mathbf{L}_{\mathrm{MI}}$ and Hessian matrices $\mathbf{H}_{\mathrm{MI}}$ are expressed as:

$$\mathbf{L}_{\mathrm{MI}} = \sum_{i,j} \mathbf{L}_{p_{\mathbf{II}^*}} \left(1 + log\left(\frac{p_{\mathbf{II}^*}}{p_{\mathbf{I}^*}}\right)\right) \tag{13}$$

and

$$\mathbf{H}_{\mathrm{MI}} = \sum_{i,j} \mathbf{L}_{p_{\mathbf{II}^*}}^{\mathrm{T}} \mathbf{L}_{p_{\mathbf{II}^*}} \left(\frac{1}{p_{\mathbf{II}^*}} - \frac{1}{p_{\mathbf{I}^*}}\right) + \mathbf{H}_{p_{\mathbf{II}^*}} \left(1 + log\left(\frac{p_{\mathbf{II}^*}}{p_{\mathbf{I}^*}}\right)\right), \tag{14}$$

where the set of possible values is not mentioned for clarity. The MI measure imposes the complete computation of the interaction matrix (no approximation as it is usually done with standard features) [21].

To face the derivation rules of the MI function, probabilities $P_{\mathbf{I}}(i)$ are interpolated by B-spline functions, written ϕ. Thus, the final analytical formulation of $P_{\mathbf{I}}(i)$, that can be considered as a normalized image histogram, becomes:

$$p_{\mathbf{I}}(i) = \frac{1}{N_{\mathbf{x}}} \sum_{\mathbf{x}} \phi\left(i - \bar{\mathbf{I}}(\mathbf{x})\right), \tag{15}$$

where the possible gray values are now $\bar{\mathbf{I}}(\mathbf{x}) \in [0, N_c - 1]$. Hence, this expression allows to reduce the histogram number of bins [28] in order to decrease the dimensionality of the problem but also to smooth the MI cost function profile [22].

Thus, from the joint probability:

$$p_{\mathbf{II}^*}(i, j, \mathbf{r}) = \frac{1}{N_{\mathbf{x}}} \sum_{\mathbf{x}} \phi\left(i - \bar{\mathbf{I}}(\mathbf{x}, \mathbf{r})\right) \phi\left(j - \mathbf{I}^*(\mathbf{x})\right), \tag{16}$$

we deduce its variations with respect to the camera pose, that is the interaction matrix (Eq. (13)) and the Hessian (Eq. (14)) :

$$\mathbf{L}_{p_{\mathbf{II}^*}(i,j,\mathbf{r})} = \frac{1}{N_{\mathbf{x}}} \sum_{\mathbf{x}} \mathbf{L}_{\phi\left(i - \bar{\mathbf{I}}(\mathbf{x},\mathbf{r})\right)} \phi\left(j - \mathbf{I}^*(\mathbf{x})\right) \tag{17}$$

$$\mathbf{H}_{p_{\mathbf{II}^*}(i,j,\mathbf{r})} = \frac{1}{N_{\mathbf{x}}} \sum_{\mathbf{x}} \mathbf{H}_{\phi\left(i - \bar{\mathbf{I}}(\mathbf{x},\mathbf{r})\right)} \phi\left(j - \mathbf{I}^*(\mathbf{x})\right). \tag{18}$$

The variation of ϕ is got by applying the chain derivation rule:

$$\mathbf{L}_{\phi\left(i - \bar{\mathbf{I}}(\mathbf{x},\mathbf{r})\right)} = -\frac{\partial \phi}{\partial i} \mathbf{L}_{\bar{\mathbf{I}}} \tag{19}$$

$$\mathbf{H}_{\phi\left(i - \bar{\mathbf{I}}(\mathbf{x},\mathbf{r})\right)} = \frac{\partial^2 \phi}{\partial i^2} \mathbf{L}_{\bar{\mathbf{I}}}^T \mathbf{L}_{\bar{\mathbf{I}}} - \frac{\partial \phi}{\partial i} \mathbf{H}_{\bar{\mathbf{I}}}. \tag{20}$$

Assuming a Lambertian scene, at least for short displacements, the interaction matrix related to the intensity at a point $\mathbf{L}_{\bar{\mathbf{I}}}$ and its Hessian $\mathbf{H}_{\bar{\mathbf{I}}}$ is obtained as follows [29]:

$$\mathbf{L}_{\bar{\mathbf{I}}} = \nabla \bar{\mathbf{I}}\, \mathbf{L}_{\mathbf{x}} \quad \text{and} \quad \mathbf{H}_{\bar{\mathbf{I}}} = \mathbf{L}_{\mathbf{x}}^{\mathrm{T}} \nabla^2 \bar{\mathbf{I}}\, \mathbf{L}_{\mathbf{x}} + \nabla_x \bar{\mathbf{I}}\, \mathbf{H}_x + \nabla_y \bar{\mathbf{I}}\, \mathbf{H}_y, \tag{21}$$

where $\nabla \bar{\mathbf{I}} = \left(\nabla_x \bar{\mathbf{I}}, \nabla_y \bar{\mathbf{I}}\right)$ are the image gradients, $\nabla^2 \bar{\mathbf{I}}$ are the gradients of image gradients and $\mathbf{L}_{\mathbf{x}}$ is the Jacobian of a point that links its displacement in the normalized image plane to the camera velocity. $\mathbf{H}_{\mathbf{x}}$ and $\mathbf{H}_{\mathbf{y}}$ are the Hessians of the two point coordinates with respect to the camera velocity [30]. The Jacobian $hbfL_{\mathbf{x}}$ is given by [31]:

$$\mathbf{L}_{\mathbf{x}} = \begin{bmatrix} -1/Z & 0 & x/Z & xy & -\left(1 + x^2\right) & y \\ 0 & -1/Z & y/Z & 1 + y^2 & -xy & -x \end{bmatrix}. \tag{22}$$

The Jacobian depends on both the position (x, y) of the point in the normalized image plane and its depth Z in the camera frame. Z is obtained from the 3D engine rendering our textured 3D model, using the Z-buffer. Therefore, the Jacobian and Hessian can be exactly computed [22]. During the iterative process of the optimization, the virtual camera moves, causing the depth of each point to change. The Jacobian and Hessian matrices are therefore changing at each iteration. Since Z^* is needed (Eq. (11)), we assume that the depth of points between current and desired poses is not so different and fix, for each point, $Z^* = Z$, since consecutive poses are close. Therefore, at convergence, the estimation of $\mathbf{H}^*$ will be accurate.

The algorithm presented in Fig. 2 sums up all the processes of the mutual information based pose estimation and tracking approach.

To illustrate the convergence and behavior of the MI optimization, an initial pose distant from 2.5 cm and 3.3° from the optimal one is set, for a real image of the "tea box" sequence (see Section 4.2.1 for more detailed results). Then, it is interesting to see the evolution of MI over iterations of pose optimization (Fig. 3) as it is smooth and reaches logarithmically its maximum value.

4. Results

This section tackles results obtained with our proposed method and their evaluation in simulation and under several real conditions, increasing progressively difficulty and extent of experiments. Simulation results aim to evaluate the potential precision of the method. The first real experiment deals with a moving object and a static camera whereas the two others deal with a moving camera.

4.1. Simulation results

The mutual information based pose estimation method has been evaluated on a synthetic image sequence. A dataset from the benchmarks of TrakMark [32] is used for this (Fig. 4(a)). The dataset is named "Conference Venue Package 01" and the virtual camera has motion composed of translation, panning and tilting, the most challenging motion of this TrakMark dataset.

Our algorithm succeeds to retrieve the camera motion all along the benchmark sequence of 1210 images (see the first result presented in the video submitted as Supplementary material). The precision estimation is evaluated both in the image and in 3D.

Image differences, between reference images from the dataset and images obtained at optimal poses computed thanks to our method, are a good way to qualitatively evaluate estimated poses. Image differences should uniformly have a gray value of 128 over 256 levels, at least where the tracked 3D model is visible (the ceiling is not taken into account), when both images are identical. Fig. 4(b) and (d) shows some difference images at different optimal locations. To give more intuition about this result, an image difference at an initial camera pose is shown in Fig. 4(c). Since the optimal and initial camera poses are not identical, obviously the difference between desired and current images is not uniform at a level of 128.

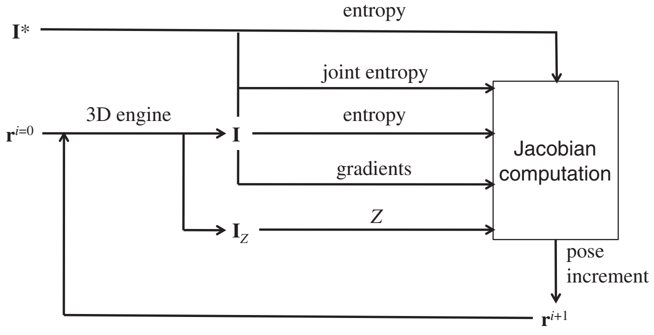
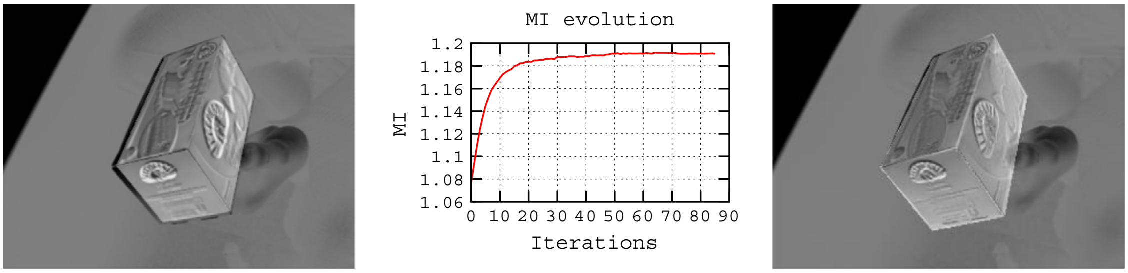


Fig. 2. Synopsis of the mutual information pose estimation algorithm. The process loops until the mutual information between **I** and **I*** is stable.



(a) (b) (c)

Fig. 3. (a) The initial pose is distant from 2.5 cm and 3.3°. (b) Evolution of Mutual Information over iterations for one real image. (c) The difference image after convergence of the MI cost function (to be compared with (a)).

After having evaluated results qualitatively in the images, the evaluation of estimations is done quantitatively in 3D. The estimated trajectory is extremely close to the ground truth which is enclosed in a 5 m × 8 m × 0.7 m volume. Translation and rotation errors are presented in Fig. 5 (green curves). The translation error is the norm of the difference between real and estimated translations. The rotation error is computed as follows, considering $\mathbf{R}^*$ is the ground truth rotation matrix and $\mathbf{R}$ is the estimated one:

1. Compute the “difference” rotation matrix $\mathbf{R}_d = \mathbf{R}^*\mathbf{R}^T$.
2. Decompose $\mathbf{R}_d$ into an axis and angle of rotation with Rodrigues' rotation formula.
3. The rotational error between ground truth and estimation is the absolute value of this angle.

Errors are displayed in Fig. 5. They have shown to be better than model based tracking approaches, using contours as geometric features [33] (red curves in Fig. 5, with a mean position error of around 15 mm, which is twice lower than the feature based on one, and a mean orientation error of 0.15°, i.e. 2.6 times lower than the feature based approach).

4.2. Results on real scenes

4.2.1. Validation on the “tea box” sequence

A first evaluation on real images is led with a static camera in the field of view of which a box is moved with coupled translation and rotation motions (Fig. 6). Faces of the box were scanned to map textures on its 3D model. Obviously, with respect to the rendered image, the real box presents a different illumination, specular reflections and partial occlusions with fingers. Despite these perturbations, the tracking succeeds all along the 500 image sequence (see the second result presented in the video submitted as Supplementary material). Fig. 6 shows two snapshots on the image sequence with, for each one, the real image, the synthetic image with virtual camera at optimal pose and the Z-buffer needed for the geometrical part of the interaction matrix (Eq. (22)).

Difference images in Fig. 6 allow us to evaluate the quality of the tracking since we do not have ground truth for this experiment. We can however note that the virtual model is perfectly aligned with the real box in the image, whatever the orientation is.

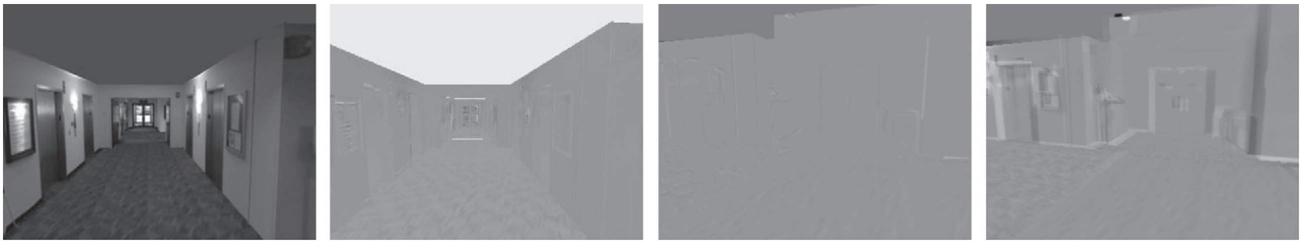
4.2.2. Tracking of a unique building

Dealing with more complex scenes is a challenge and we present here a result of the tracking of a building using the proposed method. We got a textured model of the building and took a video using a smartphone without known calibration. We used the smartphone camera specifications to compute a set of intrinsic camera parameters, which are clearly not optimal. Despite approximations, the tracking of the building succeeds along the sequence of 765 images (Fig. 7 and see the third result presented in the video submitted as Supplementary material). One must note the motion is mainly made of rotations around the vertical axis (approx. Y axis of the camera) as estimations plotted in Fig. 7(c) shows. These estimations are very stable.

4.2.3. Application to vehicle localization

Another goal of our mutual information based pose estimation is to estimate the pose of a moving camera, embedded on a vehicle, using its images (Fig. 8) and a textured 3D model of the city in which the car is driven.

To give an insight about the required precision of the initial pose in this context, the convergence area of the MI cost function is drawn in Fig. 9 for two cases. They show that the maximum of the MI cost function is reachable from more than 70 cm from the optimal pose or 50 cm combined with a 2.5° heading error (local maximum observed at −3°). The latter characterization of the cost function convergence area allows us to compute the approximated maximum car speed at which the proposed method should work. Considering a 25-frame per second camera and its maximum displacement between two views of 70 cm (that corresponds to the measured convergence area), the proposed technique should deal with a car speed of up to 63 km/h, which is above the speed limit in cities, the targeted application place. It is



(a) image 725 (b) difference 725 at optimal (c) difference 125 at optimal pose (d) difference 125 at initial pose

Fig. 4. An image of the benchmark (a). The registration quality is shown by difference images between the desired image and the image at optimal pose (b–c). To compare, a difference using an initial pose, which is obviously not the optimal one, is shown (d).

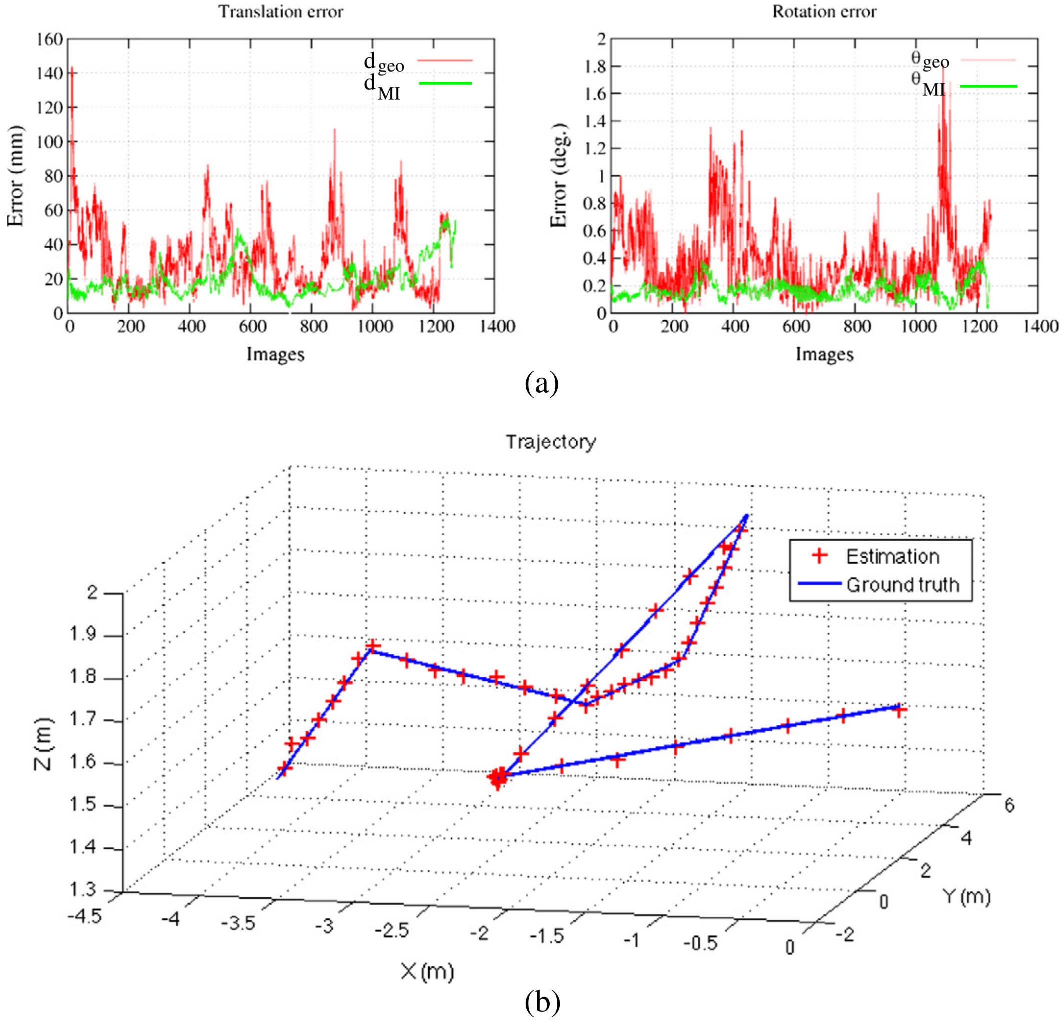


Fig. 5. Estimation errors in position and in orientation (a) over all the sequence (red, estimation errors using the geometric method of [33] and green, our MI based method), with respect to the ground truth (trajectories in (b): red crosses plot estimations and blue lines are the ground truth).

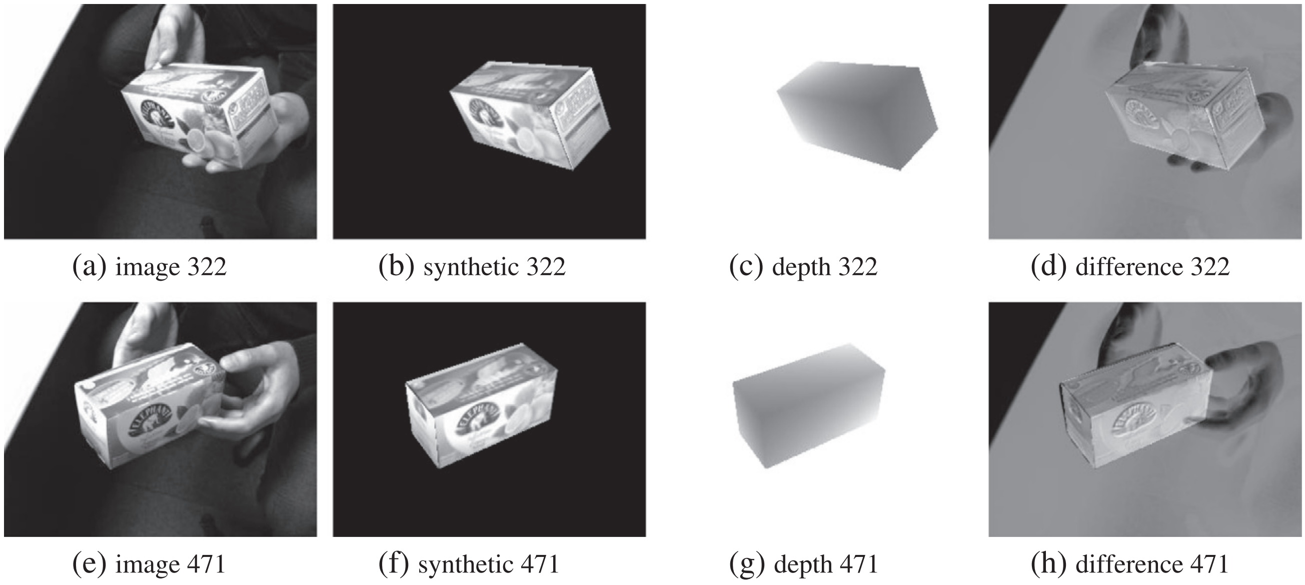
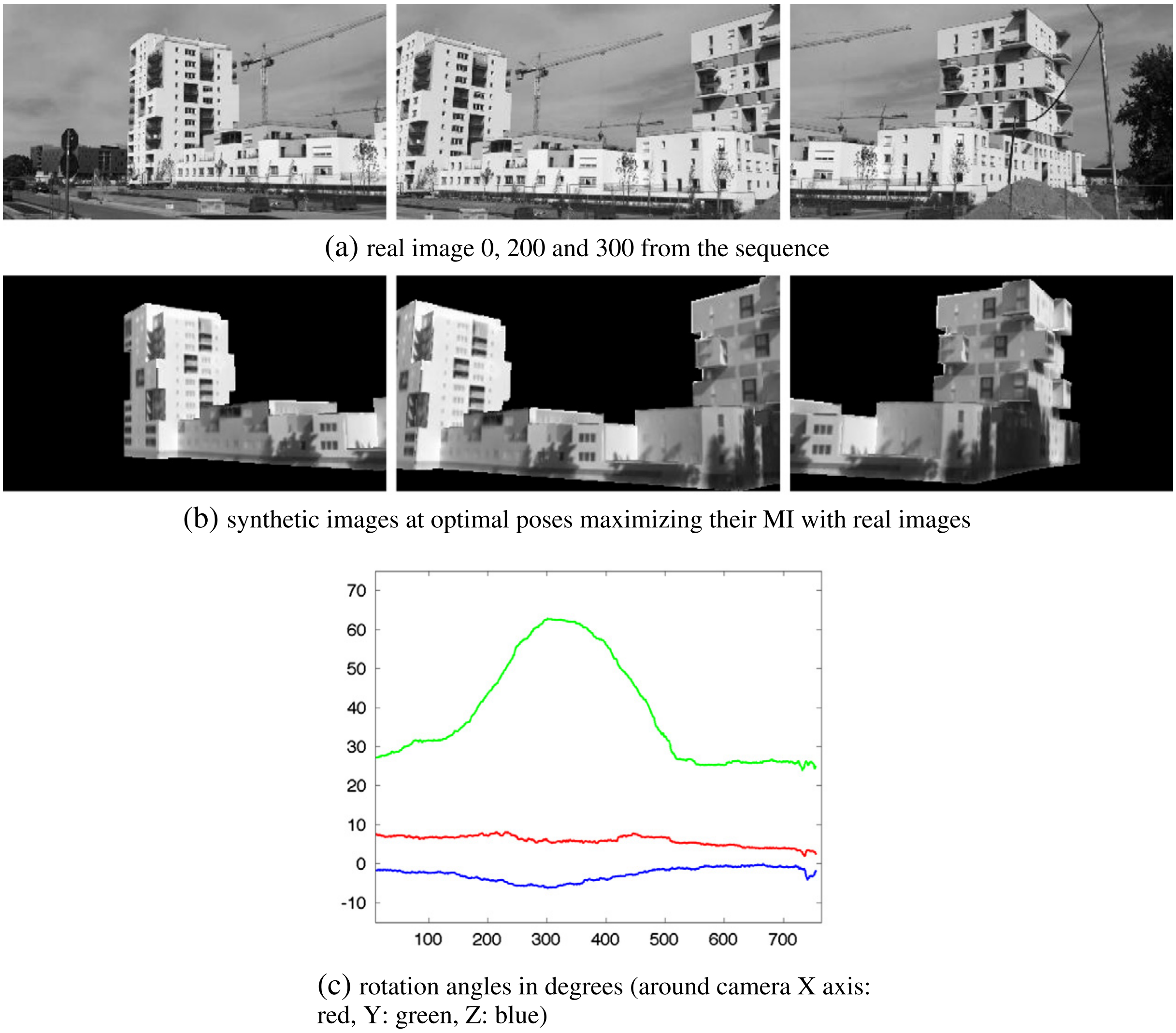

(a) image 322
(b) synthetic 322
(c) depth 322
(d) difference 322
(e) image 471
(f) synthetic 471
(g) depth 471
(h) difference 471

Fig. 6. Tracking a tea box over 500 images. (a, e) Three images on which (b, f) the synthetic view is registrated, with the *Z* for each pixel of the object (c, g). To see the tracking precision, differences between real and synthetic images are computed (d, h).



(a) real image 0, 200 and 300 from the sequence

(b) synthetic images at optimal poses maximizing their MI with real images

(c) rotation angles in degrees (around camera X axis: red, Y: green, Z: blue)

Fig. 7. Tracking results from a smartphone video (a) without knowing the optimal calibration. Camera poses are correct as synthetic images in (b) highlight this. Panel (c) shows the evolution of rotation angles along the image sequence.

however clear, considering the current processing time that, for the time being, data can currently be processed off-line only.

A way to obtain this initial pose can be to use a GPS in conjunction with inertial measurements of the car correlated over time and to the road knowledge to deduce a correct lateral and longitudinal position and heading of the car, and, hence, the camera. In a close future, the open service of the Galileo European localization system will offer a localization precision below 1 m [34] and would be used alone to initialize such a vision-based localization system. In this experiment, such equipments were not used and the initial pose was manually set, driving the virtual camera to produce a synthetic image as similar as possible to the first image of the real sequence.

Contrary to previous real experiments, we can superimpose the estimated trajectory over a satellite view of the city to evaluate qualitatively the estimation precision (Fig. 10(a)). The fact that the estimated trajectory is well aligned with streets and is on their center (single direction

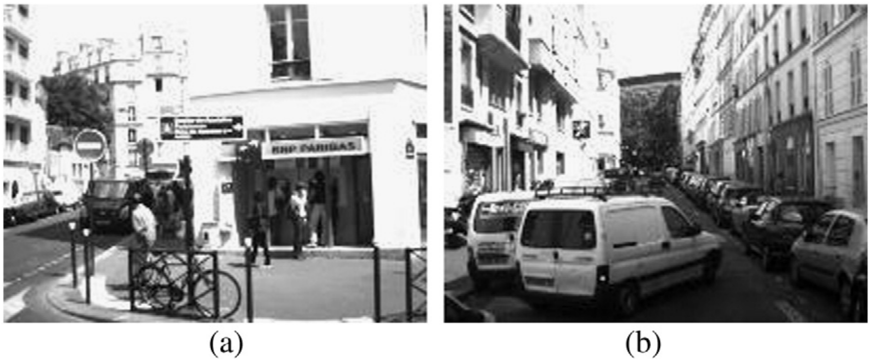


Fig. 8. Example images of the Paris XIIth sequence with occlusions of buildings by people and cars.

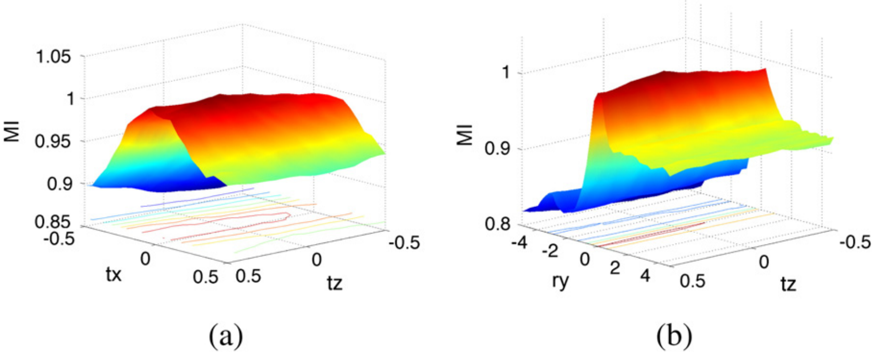


Fig. 9. MI cost function computed between a real image (the first image of the sequence in the city) and virtual images by varying two camera degrees of freedom: horizontal translations (tx and tz) in meters (a) and longitudinal translation (tz) in meter and heading (ry) in degrees (b).

street) highlights the stability and precision of the estimated poses, despite occlusions of buildings by cars (Fig. 1(b) and (c)), illumination changes, camera vibrations or the bend at the beginning of the sequence (bottom of Fig. 10(a) and see the last result presented in the video submitted as Supplementary material).

Furthermore, many sensors were embedded on the acquisition car and a geo-referenced set of positions synchronized with images is available (black trajectory in Fig. 10(b)). Compared to this "ground truth", our method leads to a mean error of 1.56 m ($\sigma = 0.61$ m, max = 2.97 m, min = 0.04 m). For a total traveled distance of 286.58 m, the mean error ratio is 0.54%, with no drift accumulation.

Fair comparison with other computer vision based techniques is hard since SIFT matching like approaches, now conventional, are not applicable in our context due to differences in the textures.

One may note the estimated path is not continuous (between red and green paths and between green and cyan paths, Fig. 10(a)). This is due to several factors, such as: texture quality, percentage of building occultation, parallax issues. Fig. 11 shows cases where the proposed method diverges since real and synthetic images are not enough similar to allow any computer vision method to work. For information, between images in Fig. 11(a) and in (b), still 0 SIFT matches are made, and 23% of false matches over a total 26 matches are obtained with ASIFT (correct matches are only made on the right side of the image). For images in Fig. 11(c) and (d), no SIFT matches are made, 64 ASIFT matches are made but with 33 false matches (51.5%), including 7 on the ban generating parallax issues. A partial solution would be to do visual odometry on real images and fusing the MI based pose tracking in a Kalman filter to fill these gaps. This is, however, not the purpose of this work and the re-initialization was done by manually finding the pose.

Finally, we note an erratic estimation at the bottom of the cyan trajectory in Fig. 10(a), which can be explained by an important occlusion and still low quality texture mapping, with the mapping of a foreground building using a texture of a background building (Fig. 12). Of course, a better quality 3D model should withdraw these issues but this result shows our tracking still works in these particularly hard conditions and allows to retrieve a coherent pose estimation when the 3D scene is of better quality, later in the car motion.

5. Conclusion and future works

We have tackled a new direct visual tracking and pose estimation method involving the measure of mutual information shared by two images: a real reference image and a virtual view evolving as the pose is optimized, maximizing the mutual information. The difficulty was to manage to link the variation of the mutual information measure to the variation of the camera or object pose. Results show, in simulation as well as in real conditions, particularly a camera embedded on a moving car, the method is robust and precise without drifting.

In the current implementation, it has the drawback of being not real-time with approximately four seconds of processing for each image. This

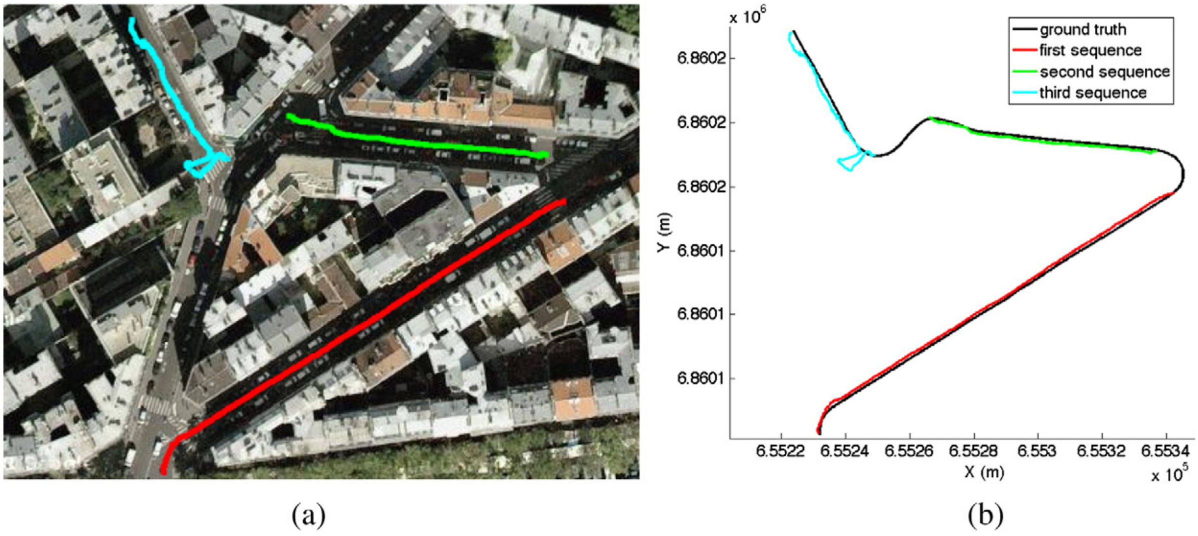


Fig. 10. Path (red) estimated by the mutual information based pose estimation, without any trajectory filtering. The stability of the trajectory estimation is clear and its precision is shown by the fact that the trajectory is well aligned in the street and there is not car or building "climbing" on each side.

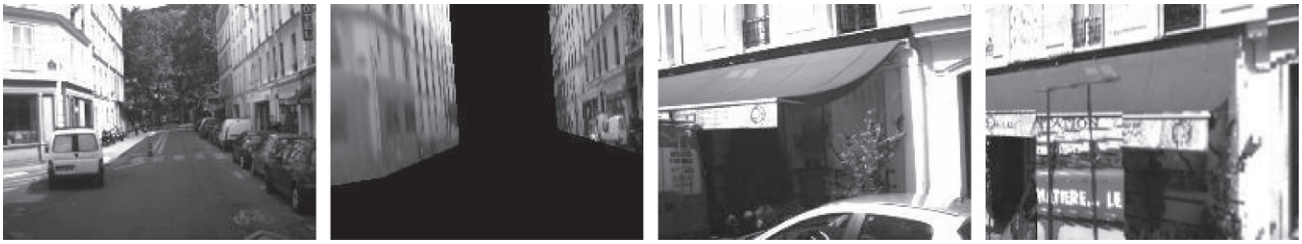

(a) real (b) synthetic (c) real (d) synthetic

Fig. 11. Two cases where the proposed method fails. (a–b) The 3D scene texture quality is extremely poor, particularly on the left. (c–d) A combination of partial occultation, erroneous texture mapping and parallax of ban, which is in 3D in reality but mapped to the vertical plane of a building in the 3D virtual scene.

is not suitable for real time purposes, but it does not deserve the theoretical advantages of our proposed method. However, a multi-resolution implementation with an incremental transformation complexity scheme, all implemented on GPU or FPGA with DSP, could solve this issue. Thus, we are clearly considering this aspect as perspectives of this work as part of a technology transfer toward an industrial partner in order to speed up the process by a significant factor (since many aspect of the computation can easily be parallelized). We also plan to tackle the low quality texture of reference model in future works.

Supplementary data to this article can be found online at http://dx.doi.org/10.1016/j.imavis.2013.10.007.

References

[1] N. Karlsson, E. Di Bernardo, J. Ostrowski, L. Goncalves, P. Pirjanian, M. Munich, The vSLAM algorithm for robust localization and mapping, IEEE Int. Conf. on Robotics and Automation, 2005, (Barcelona, Spain).

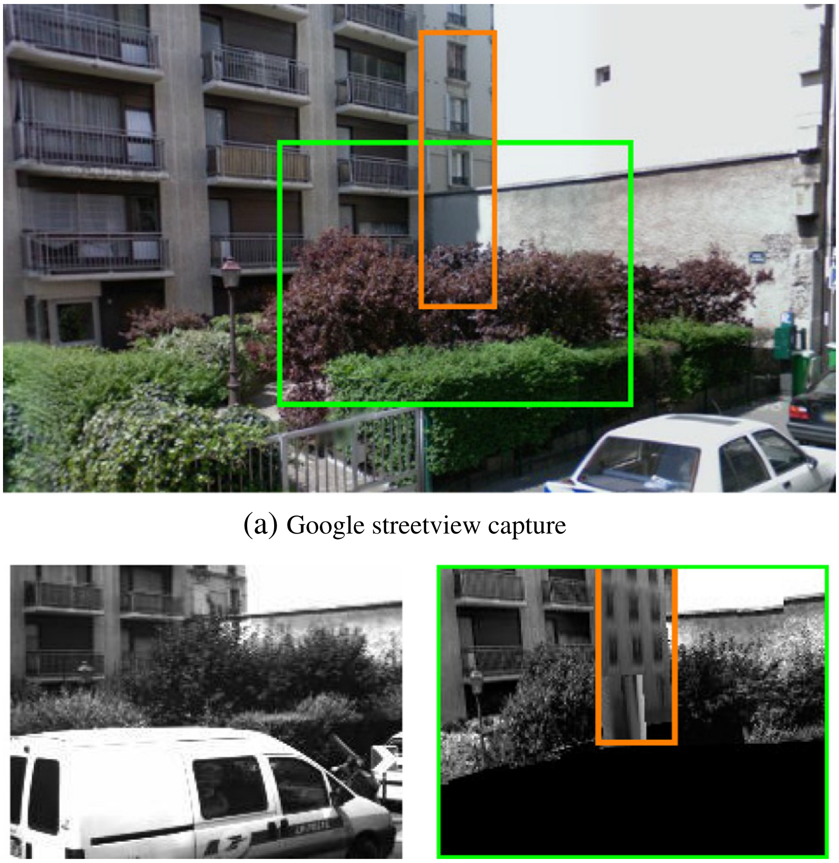
[2] T. Lemaire, S. Lacroix, Monocular-vision based SLAM using line segments, In: Int. Conf. on Robotics and Automation, 2007, (Roma, Italy).

[3] G. Silveira, E. Malis, P. Rives, Monocular-vision based SLAM using line segments, IEEE Trans. Robot. 24 (5) (2008) 969–979.

[4] B. Triggs, P. McLauchlan, R. Hartley, A. Fitzgibbon, Bundle Adjustment: A Modern Synthesis, Springer, 2000.

[5] M. Lhuillier, Automatic scene structure and camera motion using a catadioptric system, Comput. Vis. Image Underst. 109 (2) (2008) 186–203.

[6] A. Comport, E. Malis, P. Rives, Real-time quadrifocal visual odometry, Int. J. Rob. Res. Special issue on Robot Vision 29 (2–3) (2010) 245–266.


(a) Google streetview capture

(b) real (c) synthetic

Fig. 12. Issues encountered and explaining the erroneous estimation at the bottom of cyan trajectory in Fig. 10. The tracking succeeds despite strong occlusions (car and small trees, see (b) and (c)) and particularly texture mapping issue (highlighted in orange in (a) and (c)) where a texture of a building in the background is mapped on a foreground building.

[7] E. Royer, M. Lhuillier, M. Dhome, J.M. Lavest, Monocular vision for mobile robot localization and autonomous navigation, Int. J. Comput. Vis. 74 (2007) 237–260.

[8] P. David, Vision-based localization in urban environments, Army Science Conference, 2010, pp. 428–433.

[9] E. Frontoni, A. Ascani, A. Mancini, P. Zingaretti, Robot localization in urban environments using omnidirectional vision sensors and partial heterogeneous a priori knowledge, Int. Conf. on Mechatronics and Embedded Systems and Applications, MESA, Qingdao, China, 2010, pp. 428–433.

[10] R.M. Haralick, C. Lee, K. Ottenberg, M. Nolle, Analysis and solutions of the three point perspective pose estimation problem, IEEE Conf. on Computer Vision and Pattern Recognition, CVPR, 1991, pp. 592–598.

[11] V. Lepetit, P. Fua, Monocular model-based 3D tracking of rigid objects: a survey, Found. Trends Comput. Graphics Vision 1 (1) (2005) 1–89.

[12] B. Jiang, Calibration-free line-based tracking for video augmentation, Int. Conf. on Computer Graphics & Virtual Reality, CGVR, Las Vegas, USA, 2006, pp. 104–110.

[13] E. Rosten, T. Drummond, Fusing points and lines for high performance tracking, IEEE Int. Conf. on Computer Vision, vol. 2, 2005, pp. 1508–1511.

[14] A. Comport, E. Marchand, M. Pressigout, F. Chaumette, Real-time markerless tracking for augmented reality: the virtual visual servoing framework, IEEE Trans. Vis. Comput. Graph. 12 (4) (2006) 615–628.

[15] T. Drummond, R. Cipolla, Real-time visual tracking of complex structures, IEEE Trans. Pattern Anal. Mach. Intell. 24 (2002) 932–946.

[16] D.G. Lowe, Fitting parameterized three-dimensional models to images, IEEE Trans. Pattern Anal. Mach. Intell. 13 (5) (1991) 441–450.

[17] P. Georgel, S. Benhimane, N. Navab, A unified approach combining photometric and geometric information for pose estimation, British Machine Vision Conf., BMVC, 2008.

[18] M. Pressigout, E. Marchand, Real-time hybrid tracking using edge and texture information, Int. J. Robot. Res. 26 (7) (2007) 689–713.

[19] S. Baker, I. Matthews, Lucas–Kanade 20 years on: a unifying framework, Int. J. Comput. Vis. 56 (3) (2004) 221–255.

[20] N. Dowson, R. Bowden, Mutual information for Lucas–Kanade tracking (milk): an inverse compositional formulation, In IEEE Trans. on Pattern Analysis and, Machine Intelligence, vol. 30, 2008, pp. 180–185.

[21] A. Dame, E. Marchand, Accurate real-time tracking using mutual information, IEEE Int. Symp. on Mixed and Augmented Reality ISMAR, 2010, pp. 47–56, (Seoul, Korea).

[22] A. Dame, E. Marchand, Mutual information-based visual servoing, IEEE Trans. Robot. 27 (5) (2011) 958–969.

[23] D.F. DeMenthon, L.S. Davis, Model-based object pose in 25 lines of code, Int. J. Comput. Vis. 15 (1995) 123–141.

[24] C. Shannon, A mathematical theory of communication, Bell Syst. Tech. J. (1948) 27.

[25] P. Viola, W. Wells, Alignment by maximization of mutual information, Int. J. Comput. Vis. 24 (2) (1997) 137–154.

[26] G. Panin, A. Knoll, Mutual information-based 3D object tracking, Int. J. Comput. Vis. 78 (2008) 107–118.

[27] N. Dowson, R. Bowden, A unifying framework for mutual information methods for use in non-linear optimisation, European Conf. Computer Vision, 2006, pp. 365–378, (Graz, Austria).

[28] J. Pluim, J. Maintz, M. Viergever, Mutual information matching and interpolation artefacts, in: K. Hanson (Ed.), SPIE Medical Imaging, vol. 3661, SPIE Press, 1999, pp. 56–65.

[29] C. Collewet, E. Marchand, Photometric visual servoing, IEEE Trans. Robot. 27 (4) (2011) 828–834.

[30] J. Lapresté, Y. Mezouar, A Hessian approach to visual servoing, IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS, 2004, pp. 998–1003, (Sendai, Japan).

[31] F. Chaumette, S. Hutchinson, Visual servo control, part I: basic approaches, IEEE Robot. Autom. Mag. 13 (4) (2006) 82–90.

[32] TrakMark benchmarking, http://trakmark.net2009–2011.

[33] A. Petit, E. Marchand, K. Kanani, Tracking complex targets for space rendezvous and debris removal applications, IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'12, 2012, pp. 4483–4488, (Vilamoura, Portugal).

[34] European Spatial Agency (ESA), Galileo General Introduction, 2011. (http://www.navipedia.net/index.php/GALILEO_General_ Introduction).