Direct model based visual tracking and pose estimation using mutual information

Guillaume Caron, Amaury Dame, Eric Marchand

Direct model based visual pose estimation: the mutual information feature

Guillaume Caron[1], Amaury Dame[2] and Eric Marchand[3]

[1] *INRIA Rennes/IRISA, Lagadic, Rennes, France*
guillaume.caron@inria.fr
[2] *IRISA/CNRS, Lagadic, Rennes, France*
amaury.dame@inria.fr
[3] *Université de Rennes 1, IRISA/INRIA Rennes, Lagadic, Rennes, France*
eric.marchand@irisa.fr

Abstract

This paper deals with model-based pose estimation (or camera localization). The model is rendered as a virtual image and 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.

Keywords:
Omnidirectional vision, stereovision, spherical optimization, tracking

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

then equivalent to camera localization. We aim to design a new camera pose estimation method.

Camera localization has received much interest in the last few years. Visual Simultaneous Localization And Mapping [1, 2, 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 need to make a loop to correct the drift. 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However, if a 3D model on the environment is already known by the robot, exploration and loop closure issues can be withdrawn.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. It makes the pose estimation of the camera, embedded on a mobile platform, precise with no drifting, if the robot moves near these referenced [8], or even georeferenced [9] landmarks.

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 virtual 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

(a) (b) (c)

Figure 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.

wireframe models [14, 15, 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. In [20, 21], mutual information shared by a planar textured model and images acquired by the camera is used to estimate an affine transformation or an homography.

The contribution of this paper is to generalize the latter work to general 3D models defined by a 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.

The proposed method for pose estimation using a virtual reference scene is close to the work of Dame *et al.* [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 [22] physically moves a camera using a robot an real images only.

- [22] uses 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 trumps are highlighted in next sections.

The reminder of the paper is organized in three main parts. First, the general formulation of the model based 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 present, in Section 4, the behavior of the proposed pose estimation method, its precision and its robustness, before 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 min-

imizing 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 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 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 $\mathbf{s}^*$ (the observation in the image) is minimal. The optimization problem can thus be written:

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

The 3D model is classically made with geometrical features such as point, line, etc. 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 lead to important error in the tracking and pose estimation process.

2.2. Direct pose estimation

To avoid this 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 have 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 an 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 $\mathbf{I}(\mathbf{r})$. The cost function could be written as:

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

In (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 equation (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 $\mathbf{I}^*$ [23, 24]. MI is a measure of the quantity of information shared by the two images [23]. When MI is maximal, then the two images are registered. The MI similarity measure has been used for registration works [24] and more recently to track planes in image sequences [20] 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 [20, 25, 22] to the case of non planar model based pose estimation and tracking.

3. Mutual information on SE(3)

As stated in Section 2, more or less classical cost functions for pose estimation (eq. (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 $\mathcal{M}$. If $\mathbf{r}$ is the pose of the calibrated camera, the pose estimation problem can be written as [26]:

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

Virtual image $\mathbf{I}(\mathcal{M}, \mathbf{r})$ is resulting from the projection of the model $\mathcal{M}$ at given pose $\mathbf{r}$. From a first order Taylor expansion of the MI function at the current pose $\mathbf{r}$, the link between the variation of the mutual information feature and the pose variation is expressed, that is the Jacobian. The increment to apply to the pose is then obtained using a Newton's optimization like method.

To solve this function, 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 [27]. OpenGL allows not only to generate 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.

3.1. Mutual Information

MI is defined [23] by the entropy $\mathbf{H}$ of images $\mathbf{I}$ and $\mathbf{I}^*$ and their joint entropy:

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

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 $\mathbf{I}(\mathbf{x})$ ($i \in [0, N_c]$ with $N_c = 255$) 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:

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

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

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

3.2. Mutual Information based pose optimization

Camera rotations and translations are 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:

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

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:

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

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 equation (9), the approximated increment that leads to a null MI variation is:

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

As in [20], 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 $\mathbf{r}^*$ ($\mathbf{H}_{\mathrm{MI}}^{*^{-1}} = \mathbf{H}_{\mathrm{MI}}^{-1}(\mathbf{r}^*)$):

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

$\mathbf{L}_{\mathrm{MI}}$ 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 part 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 (eq. (5) and (6)), MI (eq. (4)) is developed as:

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

$\mathbf{L}_{\mathrm{MI}}$ and $\mathbf{H}_{\mathrm{MI}}$ are then analytically expressed as in [22] which imposes the full computation of the Hessian matrix (no approximation as it is usually done with standard features) [20].

To respect the differentiability conditions for the MI, probabilities $p_{\mathbf{I}}(i)$ are interpolated using B-splines functions. They allow the image histogram binning [28] in order to reduce the dimensionality of the problem and also to smooth the MI cost function profile [22].

$\mathbf{L}_{\mathrm{MI}}$ and $\mathbf{H}_{\mathrm{MI}}$ are finally function of the Jacobian of the intensity related to an image point $\mathbf{L}_{\bar{\mathbf{I}}}$ and its Hessian $\mathbf{H}_{\bar{\mathbf{I}}}$. Making the assumption of a Lambertian scene, at least for small displacements, they are found using [29]:

$$\mathbf{L}_{\bar{\mathbf{I}}} = \nabla\bar{\mathbf{I}}\,\mathbf{L}_{\mathbf{x}} \quad \text{and} \quad \mathbf{H}_{\bar{\mathbf{I}}} = \mathbf{L}_{\mathbf{x}}^{\top}\nabla^2\bar{\mathbf{I}}\,\mathbf{L}_{\mathbf{x}} + \nabla_x\bar{\mathbf{I}}\,\mathbf{H}_x + \nabla_y\bar{\mathbf{I}}\,\mathbf{H}_y, \tag{13}$$

where $\nabla\bar{\mathbf{I}} = (\nabla_x\bar{\mathbf{I}}, \nabla_y\bar{\mathbf{I}})$ 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. $\mathbf{H}_x$ and $\mathbf{H}_y$ are the Hessians of the two point coordinates with respect to the camera velocity [30]. The Jacobian $\mathbf{L}_{\mathbf{x}}$ is given by:

$$\mathbf{L}_{\mathbf{x}} = \begin{bmatrix} -1/Z & 0 & x/Z & xy & -(1+x^2) & y \\ 0 & -1/Z & y/Z & 1+y^2 & -xy & -x \end{bmatrix}. \tag{14}$$

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 traditional Z-buffer. Here is one of the differences between the current paper and [22] since in our case, the Z of each point is available whereas it is supposed the same for all points in [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 the depth of points between current and desired poses

are 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 figure 2 sums up all the processes of the mutual information based pose estimation and tracking approach.

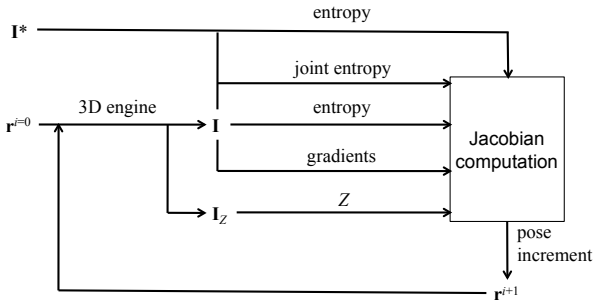
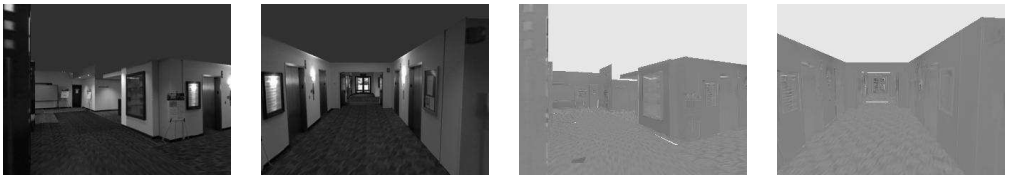


Figure 2: Synopsis of the mutual information pose estimation algorithm. The process loops until the mutual information between $\mathbf{I}$ and $\mathbf{I}^*$ is stable.

4. Results

4.1. Simulation results

The mutual information based pose estimation method has been evaluated on a synthetic images sequence. A dataset from the benchmarks of TrakMark [31] is used for this (Fig. 3(a) and 3(b)). 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.



(a) image 0 (b) image 725 (c) difference 0 (d) difference 725

Figure 3: Some source images of the benchmark (a-b). The registration quality is shown by difference images between desired and optimal image optimal pose (c-d).

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 be grey when both images are identical. Figures 3(c) and 3(d) show some difference images at different locations along the trajectory. This is due to the mutual information measure which is robust to such issues, whereas more classical registration cost functions, like the SSD, are not.

After having evaluated results qualitatively in 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 $5m \times 8m \times 0.7m$ volume. Translation and rotation errors are presented in fig. 4. 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:

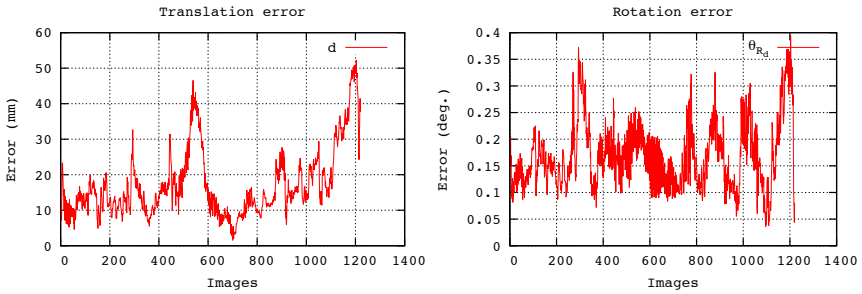
1. compute the "difference" rotation matrix $\mathbf{R}_d = \mathbf{R}^* \mathbf{R}^\mathsf{T}$
2. decompose $\mathbf{R}_d$ into an axis and angle of rotation with Rodrigues' rotation formula
3. the rotational error between ground truth and estimation is the absolute value of this angle

Errors are displayed in figure 4. They have shown to be better than model based tracking approaches, using geometric features, with a mean position error of around 15 mm, which is twice lower than the feature based one, and a mean orientation error of 0.15 degrees, *i.e* 2.6 times lower than the feature based approach.

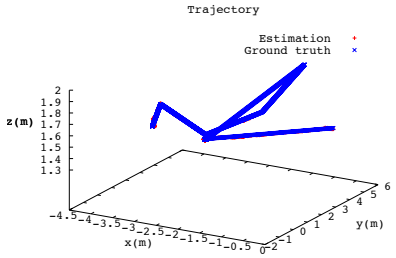
4.2. Results on real scenes

4.2.1. Validation on a simple "Tea Box"

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. 5). Faces of the box were scanned to map textures on its 3D model. Obviously, the real box present a different illumination, than the model, specular reflections and partial occlusions with fingers. Despite these perturbations, the tracking succeeds all along the 500 images sequence (see the second result presented in the video submitted as supplementary material). Figure 5 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. (14)).



(a)



(b)

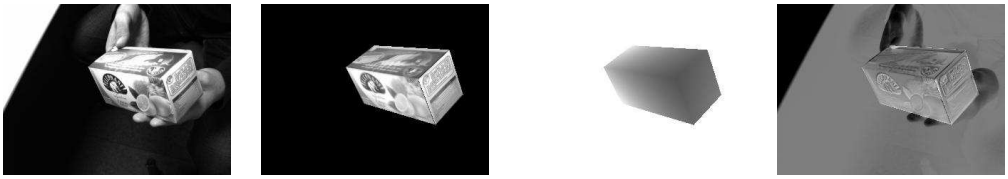
Figure 4: Estimation errors in position and in orientation (a) over all the sequence, with respect to the ground truth (trajectories in (b)).

Difference images of figure 5 allows 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.

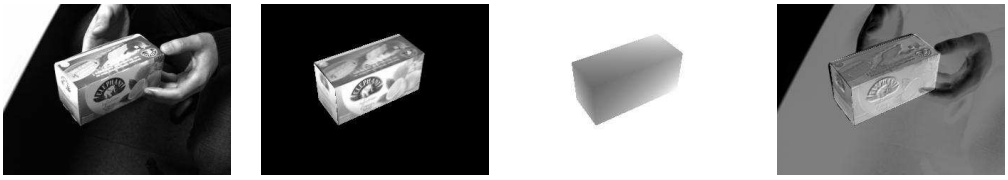
To illustrate the convergence, an initial pose distant from 2.5cm and 3.3° from the optimal one is set, for an image of the sequence. Then, it is interesting to see the evolution of MI over iterations of pose optimization (Fig. 6) as it is smooth and reaches logarithmically its maximum value.

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.

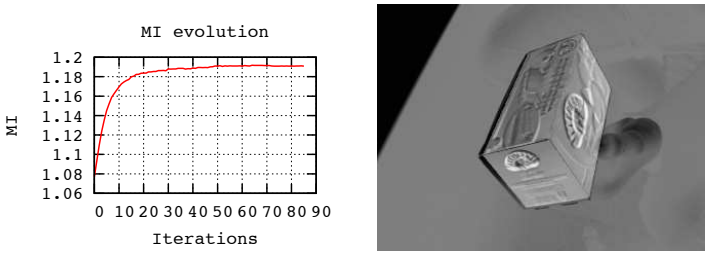


(a) image 322 (b) synthetic 322 (c) depth 322 (d) difference 322



(e) image 471 (f) synthetic 471 (g) depth 471 (h) difference 471

Figure 5: 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) (b)

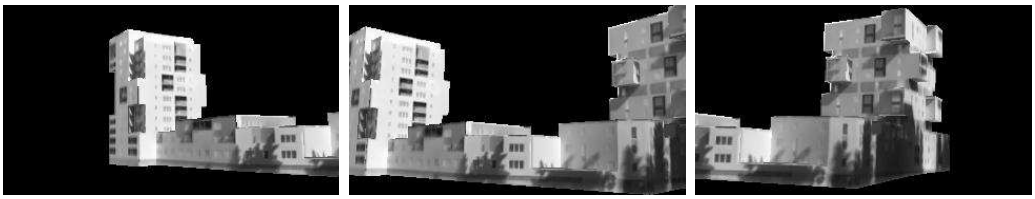
Figure 6: (a) Evolution of Mutual Information over iterations for one real image. (b) The initial pose is distant from 2.5cm and 3.3°.

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).

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. In this case, the initial pose at the beginning of the vision localization can be

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



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

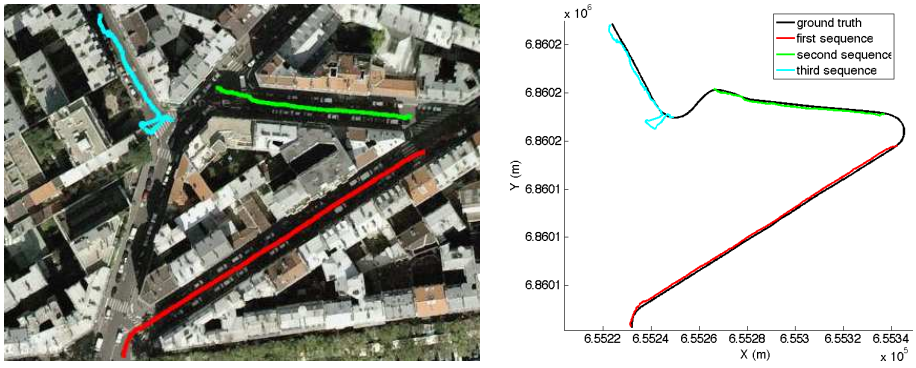
Figure 7: Tracking results from a smartphone video (a) without knowing the optimal calibration. Camera poses are correct as synthetic images in (b) highlight this.

obtained thanks to GPS before entering in the city.

(a) (b) (c)

Figure 8: Example images of the Paris XIIth sequence with occlusions of buildings by people and cars. (c) shows a case where the algorithm diverges due to major occlusion (75% of the image) with respect to the model by a ban and a small truck.

So, 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. 9(a)). The fact that the estimated trajectory is well aligned with streets and is on their center (single direction street) highlights the stability and precision of the estimated poses, despite occlusions of buildings by cars (fig. 1(b) and 1(c)), illumination changes, camera vibrations or the bend at the beginning of the sequence (bottom of fig. 9(a) and see the last result presented in the video submitted as supplementary material).



(a) (b)

Figure 9: 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.

Furthermore, many sensors were embedded on the acquisition car and a geo-referenced set of positions synchronized with images is available (black trajectory in figure 9(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 travelled distance of 286.58 m, the mean error ratio is 0.54 %, with no drift accumulation.

Comparison has been tried with other computer vision based tool is hard since we tried to match SIFT features between real and synthetic images and for the majority, there is not any match. Using the online demo of ASIFT [32], an extension of SIFT to make it robust to affine transformations, a few matches are made but with several false pairs (fig. 10). However, the small number of correct matches cannot lead to correct and precise estimations as our proposed method did.

One may note the estimated path is not continuous (between red and green path and between green and cyan path, fig. 9(a)). This is due to several factors, such as: texture quality, percentage of building occultation, parallax issues. Figure 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 figures 11(a) and 11(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

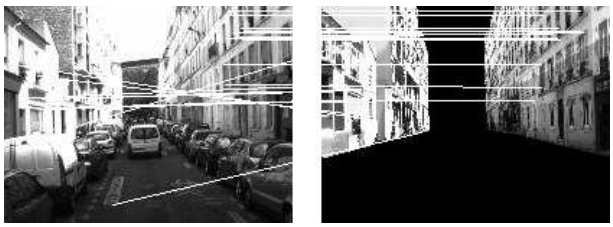


Figure 10: ASIFT matching between real and synthetic images. The synthetic image is obtained at the best pose corresponding to the real image. 26 point matches are made between 37000 ASIFT features detected in the real image and 30000 in the synthetic image. 5 are false matches: 19% of false matches.

right side of the image). For images of figures 11(c) and 11(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 fiter to fill these gaps.

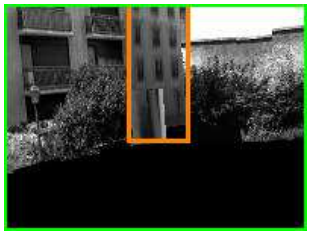
Finally, we note an erratic estimation at the bottom of the cyan trajectory in figure 9(a), which can be explained by an important occlusion and still low quality texture mapping, with the mapping of a foreground building using a texture of a background building (fig. 12). Of course, a better quality 3D model should withdraw these issues but this result shows our tracking still works in this particularly hard conditions and allows to retrieve a coherent pose estimation when the 3D scene is of better quality, later in the car motion.

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

Figure 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.

(a) Google streetview capture

(b) real



(c) synthetic

Figure 12: Issues encountered and explaining the erroneous estimation at the bottom of cyan trajectory in figure 9(a). 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.

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. However, a multi-resolution implementation with an incremental transformation com-

plexity scheme, all implemented on GPU could highly improve this issue. We also plan to tackle the low quality texture of reference model in future works.

References

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

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

[3] Silveira G, Malis E, Rives P. Monocular-vision based SLAM using line segments. IEEE Trans on Robotics 2008;24(5):969–79.

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

[5] Lhuillier M. Automatic scene structure and camera motion using a catadioptric system. Computer Vision and Image Understanding 2008;109(2):186–203.

[6] Comport A, Malis E, Rives P. Real-time quadrifocal visual odometry. Int J of Robotics Research, Special issue on Robot Vision 2010;29(2-3):245–66.

[7] Royer E, Lhuillier M, M. D, Lavest JM. Monocular vision for mobile robot localization and autonomous navigation. Int J Comput Vision 2007;74:237–60.

[8] David P. Vision-based localization in urban environments. In: Army Science Conference. 2010, p. 428–33.

[9] Frontoni E, Ascani A, Mancini A, Zingaretti P. Robot localization in urban environments using omnidirectional vision sensors and partial heterogeneous a priori knowledge. In: Int. Conf. on Mechatronics and Embedded Systems and Applications, MESA. Qingdao, China; 2010, p. 428–33.

[10] Haralick RM, Lee C, Ottenberg K, Nolle M. Analysis and solutions of the three point perspective pose estimation problem. Tech. Rep.; Hamburg, Germany, Germany; 1991.

[11] Lepetit V, Fua P. Monocular model-based 3d tracking of rigid objects: A survey. Foundations and Trends in Computer Graphics and Vision 2005;1(1):1–89.

[12] Jiang B. Calibration-free line-based tracking for video augmentation. In: Int. Conf. on Computer Graphics & Virtual Reality, CGVR. Las Vegas, USA; 2006, p. 104–10.

[13] Rosten E, Drummond T. Fusing points and lines for high performance tracking. In: IEEE Int. Conf. on Computer Vision; vol. 2. 2005, p. 1508–11.

[14] Comport A, Marchand E, Pressigout M, Chaumette F. Real-time markerless tracking for augmented reality: the virtual visual servoing framework. IEEE Trans on Visualization and Computer Graphics 2006;12(4):615–28.

[15] Drummond T, Cipolla R. Real-time visual tracking of complex structures. IEEE Trans on Pattern Analysis and Machine Intelligence 2002;24:932–46.

[16] Lowe DG. Fitting parameterized three-dimensional models to images. IEEE Trans Pattern Anal Mach Intell 1991;13(5):441–50.

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

[18] Pressigout M, Marchand E. Real-time hybrid tracking using edge and texture information. Int Journal of Robotics Research, IJRR 2007;26(7):689–713.

[19] Baker S, Matthews I. Lucas-kanade 20 years on: A unifying framework. IJCV 2004;56(3):221–55.

[20] Dame A, Marchand E. Accurate real-time tracking using mutual information. In: IEEE Int. Symp. on Mixed and Augmented Reality, ISMAR. Seoul, Korea; 2010, p. 47–56.

[21] Dowson N, Bowden R. Mutual information for lucas-kanade tracking (milk): An inverse compositional formulation. In: In IEEE Trans. on Pattern Analysis and Machine Intelligence; vol. 30. 2008, p. 180–5.

[22] Dame A, Marchand E. Mutual information-based visual servoing. IEEE Trans on Robotics 2011;27(5):958–69.

[23] Shannon C. A mathematical theory of communication. Bell system technical journal 1948;27.

[24] Viola P, Wells W. Alignment by maximization of mutual information. Int Journal of Computer Vision 1997;24(2):137–54.

[25] Dame A, Marchand E. Second order optimization of mutual information for real-time image registration. IEEE Trans on Image Processing 2012;21(9):4190–203.

[26] Panin G, Knoll A. Mutual information-based 3d object tracking. Int J Comput Vision 2008;78:107–18.

[27] Ogre 3D team . Ogre3d, open source 3d graphics engine. http://www.ogre3d.org; 2000-2012.

[28] Pluim J, Maintz J, Viergever M. Mutual information matching and interpolation artefacts. In: Hanson K, editor. SPIE Medical Imaging; vol. 3661. SPIE Press; 1999, p. 56–65.

[29] Collewet C, Marchand E. Photometric visual servoing. IEEE Trans on Robotics 2011;27(4):828–34.

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

[31] Trakmark working group . Trakmark benchmarking. http://trakmark.net; 2009-2011.

[32] Morel JM, Yu G. Asift: A new framework for fully affine invariant image comparison. SIAM J Img Sci 2009;2:438–69.