

The constrained SLAM framework for non-instrumented augmented reality

Application to industrial training

M. Tamaazousti¹ · S. Naudet-Collette¹ · V. Gay-Bellile¹ ·
S. Bourgeois¹ · B. Besbes¹ · M. Dhome²

Received: 24 October 2014 / Revised: 23 June 2015 / Accepted: 23 September 2015 /
Published online: 19 October 2015
© Springer Science+Business Media New York 2015

Abstract This paper addresses the challenging issue of marker less tracking for Augmented Reality. It proposes a real-time camera localization in a partially known environment, i.e. for which a geometric 3D model of one static object in the scene is available. We propose to take benefit from this geometric model to improve the localization of keyframe-based SLAM by constraining the local bundle adjustment process with this additional information. We demonstrate the advantages of this solution, called constrained SLAM, on both synthetic and real data and present very convincing augmentation of 3D objects in real-time. Using this tracker, we also propose an interactive augmented reality system for training application. This system, based on a Optical See-Through Head Mounted Display, allows to augment the users vision field with virtual information accurately co-registered with the real world. To keep greatly benefit of the potential of this hand free device, the system combines the tracker module with a simple user-interaction vision-based module to provide overlaid information in response to user requests.

Keywords Augmented reality · Tracking · Constrained SLAM · Optical see-through · Head mounted display · Hand-based interaction

✉ S. Naudet-Collette
sylvie.naudet@cea.fr

M. Dhome
dhome@lasmea.univ-bpclermont.fr

¹ CEA, LIST, Vision and Content Engineering Laboratory, F-91191 Gif-sur-Yvette, France

² Pascal Institut, UMR 660, Blaise Pascal University, 63000 Clermont Ferrand, France

1 Introduction

Real-time augmentation of a 3D object with a moving monocular camera is a challenging research topic. It requires to accurately compute the pose of the camera with respect to this 3D object. In the literature vision-based localization solutions can be classified into two categories. The first category of methods estimate the localization of the camera with respect to a 3D object by using an *a priori* knowledge of a model of the object (geometry and/or appearance). These methods are called model-based localization. Some of them assume that the model is completely known. They use only the information of the 3D model (geometry and/or appearance) to estimate the camera pose by matching 3D features extracted from the model with their corresponding observations in the image [23]. Other approaches question this completely known model assumption and consider the localization of a partially known object whose appearance has to be updated. It consists in localizing the camera by using the 3D model while enriching it through on-line reconstruction of primitives. Nevertheless, all model-based solutions imply that the object of interest is always visible and occupies a large part of the images during the whole sequence for accurate and stable tracking. These solutions are generally subject to jittering effects and are sensitive to occlusions, mainly because they only use the visual information of the interest object in the current frame (the pose is determined independently for each frame).

The other category considers a camera moving in a completely unknown environment. SLAM (Simultaneous Localization And Mapping) solutions, [17, 33, 40] perform an on-line reconstruction of the primitives extracted in the images and use this reconstruction (usually a sparse 3D point cloud) to estimate the relative motion of a camera without any prior on the scene geometry. SLAM solutions are very stable since the whole information present in multi images is used to compute the camera poses. However, they are subject to three major drawbacks which prevent them to be used in a 3D object tracking context: the initial coordinate frame is arbitrary chosen, the scale factor of the scene is arbitrary set and the localization suffers from error accumulation due to image noise, matching errors, *etc.*

Recently, different works tried to solve some of these drawbacks by combining SLAM with model-based tracking. For example Bleser et al. [3] introduce a solution to set the coordinate frame and the scale factor of a SLAM algorithm by using a model-based tracking algorithm at the first frame. In [16], the algorithm automatically switches between model-based localization and SLAM algorithm. The former is used when the object covers a large area in the image while the latter is favored for the remaining time. These solutions use alternatively constraints from the model and from the multi-view geometry which does not guarantee that the estimated trajectory and the reconstruction of the scene are optimal for all constraints.

We propose a solution to unify all these localization methods in a single framework called constrained SLAM. The goal is to cumulate their benefits and limit their disadvantages. We consider that the camera moves in a partially known environment, *i.e.* for which a 3D model of a static object in the scene is available. This absolute information provided by the 3D model of the object is directly included in the bundle adjustment process. Thus, model and multi-view constraints are used simultaneously. In order to handle a wide range of 3D objects and scenes, two classes of constraints are proposed in this study. The first one allows to unify the SLAM and model-based methods by constraining the trajectory of the camera through the projection, in the images, of the 3D primitives extracted from the model. The second one constrains the 3D primitives, reconstructed by the SLAM and

associated to the object, to belong to the surface of the model. They have been designed to ensure real-time performance in a local bundle adjustment implementation. This paper is an extended version of our two previous papers [48, 49] that have presented independently the two constraints mentioned above without comparison. This paper gives a detailed and comprehensive overview of the constrained SLAM framework and proposes a comparative evaluation between the two classes of constraints. In addition to a detailed state of art, new results and new comparative evaluations, this paper highlights the possibility to easily combine this two classes of constraints. The benefits of the constrained bundle adjustment framework are also demonstrated on new real sequences.

Using this new tracker, we present an interactive augmented reality system which allows augmenting the users vision field with virtual information accurately co-registered with the real world. The system uses Optical See-Through (OST) glasses integrating a calibrated camera and allows to interactively augment in real time an industrial object with virtual sequences designed to train a user for specific maintenance tasks. The training leverages user interactions by simply pointing on a specific object component, with a laser pointer or with his finger, making the learning process more interesting and intuitive. This prototype is demonstrated for industrial training but it has a great potential for a broad range of applications in the field of training and learning.

Plan. Section 2 presents a state of the art of vision-based localization methods. In Section 3, we briefly describe the local bundle adjustment process associated to keyframe-based SLAM algorithms. Section 4 gives an overview of the proposed constrained bundle adjustment framework with a description of the additional steps required. Then, the two classes of model-based constraints are introduced in Section 5. The resulting constrained bundle adjustments are evaluated on synthetic and real data for different types of objects (textured and textureless), in Section 6. We also present in Section 8 the Augmented Reality application for industrial training. Finally, we give our conclusions and discuss future works in Section 9.

Notation. Matrices are designated by sans-serif fonts such as M . Vectors are expressed in homogeneous coordinates, e.g. $\mathbf{q} \sim (x, y, w)^T$ where T is the transposition and \sim the equality up to a non-zero scale factor. $\|\cdot\|$ represents the Euclidean distance. SLAM reconstruction is composed of N 3D points $\{\mathbf{Q}_i\}_{i=1}^N$ and m cameras $\{\mathbf{C}_k\}_{k=1}^m$. We note $\mathbf{q}_{i,k}$ the observation of the 3D point \mathbf{Q}_i in the camera \mathbf{C}_k and \mathcal{A}_i the set of camera indexes observing \mathbf{Q}_i with $n_i = \text{card}(\mathcal{A}_i)$. The projection matrix \mathbf{P}_k associated with the camera \mathbf{C}_k is given by $\mathbf{P}_k = \mathbf{K}\mathbf{R}_k(l_3 | -\mathbf{t}_k)$, where \mathbf{K} is the matrix of the intrinsic parameters and $(\mathbf{R}_k, \mathbf{t}_k)$ the extrinsic ones.

2 Related works

This section presents a state of the art of vision based localization methods. It distinguishes localization methods using only the knowledge of a 3D model of an object in the scene, methods without *a priori* knowledge on the environment, and finally the methods with partial knowledge of the environment (*i.e.* an unknown environment with as *a priori* a 3D model of an object in the scene).

2.1 Model-based solutions

We present in this section the methods that use only a model of an object in the scene (that is to say, they ignore the rest of the scene), to estimate the movement of the camera.

2.1.1 Localization with a static model

The principle of this family of localization methods is to establish a matching between features (points, edge, patch, *etc*) of the current image and features of the 3D model. Then, to determine the six degrees of freedom of the camera pose, the distance between the projection of the 3D features in the image and their corresponding 2D features is minimized. The methods used depend mainly on the nature of the model which may represent the shape of the object (geometric model), the appearance (photometric model), or both (photo-geometric model). Geometric model are widespread but 2D/3D correspondences are generally difficult to establish because edge features are few discriminating. To constrain these correspondences, existing solutions [6, 8, 13, 28, 30, 31] make the assumption that the movement of the object is small between one image to another which limits their robustness to fast camera displacement. These methods have the advantage to be robust to variations of the object appearance due to wear of the object (*eg.* stains) or to lighting condition changes.

Many works [24, 29, 39, 41] rather propose to use a 3D point cloud with appearance descriptor as model since it provides very discriminative features. The appearance of each feature of the 3D model can be compared to the appearance of the 2D features extracted in the current frame to determine if they match. These solutions are particularly robust since they can handle each frame independently. However the limited accuracy and repeatability of feature extraction can lead to jittering.

When a textured 3D surface model is available, it is possible to match patches of the model with their corresponding areas in the image. The mapping is then expressed as a problem of image alignment and the optimal pose of the camera corresponds to the one that best aligns in terms of an intensity criterion the current image with the textured faces of the 3D model. This approach was exploited by authors [2], who use the Efficient Second-order Minimization (ESM) algorithm to allow real-time processing. In addition, authors [4] propose a solution based on mutual information which increases the robustness to lighting conditions at the cost of higher processing time. While this kind of approach has the advantage of not requiring feature extraction and therefore is less prone to jittering, it requires small movements between consecutive images in order to predict an initial pose for the current frame which have to be relatively close to the solution. This method is also less robust to partial occlusion of the object. To take advantage of the intensity-based and points-based methods mentioned above, authors [22, 53] propose to combine both approaches in a single localization process. Thus, as long as the image alignment method (ESM) converges, it is used to obtain a jitter-free localization, and when it fails, a point matching method is used to reinitialize the localization process.

To conclude, the introduction of the appearance in the 3D model of the object simplifies the matching step between the features of the model and those of the image but at the price of a sensitivity to lighting conditions. In addition, it is not always possible to have such a model since it must represent not only the shape of the object but also its current appearance. Thus, if an object has undergone significant changes in appearance due to its exploitation (smudges, stains, *etc.*), the model must be updated to take into account these changes.

2.1.2 Localization of a dynamically updated model

To update the appearance of the object model, authors [21, 38, 44] propose to exploit a geometric 3D model and reconstruct a 3D point cloud on it, characterized by local descriptors of appearance. The 3D point cloud is obtained from interest points extracted in the images and back-projected onto the 3D model. The resulting map is then used to estimate poses of the object in the following frames, which are then used to reconstruct new 3D points. Note that this localization process needs to be set with an accurate initial camera pose, which can be a major constraint depending on the application.

These methods have a main advantage over model-based solution without model update. They are more robust since the local descriptors are computed from images corresponding to the current lighting conditions and the current appearance of the object. However, the 3D point cloud is simply reconstructed by back-projection, its accuracy depends on the quality of the initial camera pose and on the presence or not of occlusions. In addition, the localization and the reconstruction processes are used alternately without ever questioning the camera poses and positions of the 3D points estimated. These methods are then subject to error accumulation and jittering.

To solve these problems, Vacchetti et al. propose in [51, 52] to unify model-based solution with or without update. The camera poses are estimated not only from features reconstructed on-line, but also from those of the initial model (geometric [52] or photo-geometric [51]) of the object. Thus, the 3D features from the initial model can prevent the errors accumulation while the on-line reconstructed features help to better constrain the motion estimation between two frames.

In conclusion, model-based solutions with update of the model appearance improve the robustness to lighting conditions. However, the camera poses are still estimated using only the features corresponding to the object of interest, which requires that the object is always visible, not subject to large occlusions and covers substantial part in the image for an accurate localization.

2.2 SLAM solutions: localization in an unknown environment

Localization in an unknown environment use multi-view geometry constraints to estimate the movement of the camera from image observation. The features are generally points, edges, patches which are easy to track from an image to another. While some solutions estimate the camera poses along the sequence directly from image observations [36], most of existing solutions [17, 33] create of a 3D map of the environment. These methods are based on the assumption of a rigid scene.

There exist two main classes of SLAM algorithms:

- Keyframe-based SLAM methods based on bundle adjustment [50],
- Filtering-based SLAM methods based on filtering estimation techniques [7].

They have both their own advantages. However, as described in [45], keyframe-based SLAM are more accurate since they allow to reconstruct more 3D features in real-time. In the following, we present only keyframe-based SLAM solutions. They adapt the Structure from Motion (SfM) techniques [43] for sequential processing in real-time [9, 33, 36]. It consists in using on-line and incrementally conventional computer vision tools (feature matching, triangulation, pose estimation and bundle adjustment). While the camera poses are estimated for all the images, adding new features to the map is performed only for some frames called keyframes. These keyframes can be selected on a sliding temporal window

around the current frame [33] or on a spatial criterion as in [17, 18]. The camera poses and the 3D features of the map are simultaneously refined with a Bundle Adjustment (BA). The BA is the iterative optimization of both camera poses and map points which minimizes the reprojection error, called also the multi-view constraint, *i.e.* the distance between the expected and actually measured projected points of a map within the images. While off-line solutions [40] use a global bundle adjustment that refines the whole reconstruction of a video sequence in a single optimization process, on-line solutions [17, 33] use a local bundle adjustment that optimizes sequentially only a limited number of camera poses and the 3D points they observe¹.

Thanks to this multiview constraints, SLAM solutions are very stable since they minimize a reprojection error simultaneously in multi-images using the whole 2D information in images. However, these methods have limitations which prevent their use in many applications (*e.g.* 3D object tracking). In fact, SLAM solutions estimate the motion of the camera in an arbitrarily coordinate frame and suffer from error accumulation due to image noise, matching errors, *etc.* When only one camera is used as sensor, the reconstruction is performed up to a scale factor. This latter is fixed arbitrarily at the beginning of the process and must be propagated throughout the sequence. It is difficult to propagate this scale since it is not observable, thus monocular SLAM methods suffer also from drift of the scale factor.

2.3 Localization in a partially known environment

Localization methods with respect to a 3D model, with or without update, exploit, in the image, only the visual information corresponding to the object. However, if they are static relative to the rest of the scene, unknown elements around it can provide additional constraints (multi-view constraints such as in the SLAM method). The localization approaches considering a partially known environment exploit both geometric constraints imposed by the 3D model of the object of interest and the constraints of the multi-view geometry to estimate the camera pose.

A first solution to combine these two approaches was proposed by [3], they propose to initialize a SLAM algorithm from a pose obtained by a model-based solution. This pose is then used to reconstruct a first 3D point cloud by retro-projecting the interest points of the image on the 3D model of the object. Only the SLAM process is then used to estimate the pose of the camera in the following frames. This approach provides a coordinate frame and an initial scale to the SLAM. However, it is prone to error accumulation when the camera moves away from the initial 3D point cloud. In addition, this initial reconstruction is based on a localization from a single view point, which involves a potential error of the camera pose along the optical axis. This error will be maintained throughout the process since the first camera pose is never questioned.

To deal with the error accumulation problem, different approaches propose to alternate SLAM and model-based methods. Thus, authors [16] use a spatial criterion to switch between the two methods. The camera is localized with a 3D geometric model when the object covers a large area in the image (small distance camera/object), and with a SLAM otherwise. Other solutions [11, 12] reset the error accumulation of the SLAM with the pose returned by a model-based solution when it is obtained with a high confidence. These two approaches remain suboptimal because they do not optimize simultaneously all the constraints provided by the scene (constraints from the model and multi-views).

¹In [17] a global bundle adjustment is also performed in a dedicated thread

In [26], a SLAM reconstruction is refined with a cost function that include simultaneously model-based and multi-view constraints. However only the known part of the environment is used, all the point reconstructed by the SLAM process around the object of interest are ignored in the optimization. It fails when the object of interest is hidden, out of the field of view or covers a small part in the images.

2.4 Discussion and position

The solutions of localization in a partially known environment are those that offer the maximum of benefits when the object is static relative to the scene. However, existing solutions do not use simultaneously all the constraints available (model and multi-view) or use them simultaneously but only on the known part of the environment (on the object). The former approach does not guarantee that the estimated trajectory and the reconstruction of the scene are optimal for all constraints where as the latter approach provides inaccurate localization when the object of interest covers a small area in the image.

In this paper, we propose a solution of partially known localization based on a SLAM algorithm incorporating simultaneously constraints from a 3D model of the object and multi-view constraints of the whole scene (known and unknown part of the environment). Thus, when the object is visible in the image, the model constraints ensure an accurate global localization in the same coordinate frame and at the same scale as the object of interest. This object of interest is continuously accurately located even if it is hidden, out of the field of view or represents few pixels in the images by using the map of the environment.

3 Bundle adjustment for keyframe-based SLAM

Bundle adjustment (BA) minimizes the sum of square differences between the projected 3D points and the associated image observations. This geometric distance is called the reprojection error. The optimized parameters are the coordinates of the N 3D points and the 6 extrinsic parameters of the m camera poses, thus the total number of parameters is $3N + 6m$. The cost function of the bundle adjustment is given by:

$$\mathcal{E} \left(\{R_j, \mathbf{t}_j\}_{j=1}^m, \{\mathbf{Q}_i\}_{i=1}^N \right) = \sum_{i=1}^N \sum_{j \in \mathcal{O}_i} d^2(\mathbf{q}_{i,j}, P_j \mathbf{Q}_i), \quad (1)$$

where $d^2(\mathbf{q}, \mathbf{q}') = \|\mathbf{q} - \mathbf{q}'\|^2$ is the point-to-point distance.

For long sequences, the BA becomes rapidly very time consuming and not adapted for real-time localization even with an efficient sparse implementation (like proposed by [50]). To tackle this limitation, the local bundle adjustment framework has been proposed [17, 33]. The idea is to reduce the number of estimated parameters by optimizing only a subset of reconstructed points and camera poses. In its original implementation, the optimized parameters are the T most recent camera poses (where T is selected to maintain real-time performance²) and the reconstructed points they observe. In [17], the cameras and the 3D points optimized in the local bundle adjustment are selected with a spacial criterion³.

² $T = 3$ in [33].

³Note that due to a parallel implementation in a mapping and a tracking threads, a global bundle adjustment is also performed in [17] while keeping real-time performance.

Equation (1) is iteratively minimized by the Levenberg-Marquardt (LM) [32] algorithm in our experimentations. By taking advantage of its specific sparse structure, the BA can be efficiently implemented as described in [27, 50].

4 The constrained bundle adjustment framework

4.1 Formalism of the constrained bundle adjustment

We introduce an original solution for camera localization in a partially known environment by incorporating the geometric constraints provided by the 3D model of a static object in the scene in the bundle adjustment. The proposed constrained bundle adjustment uses a compound cost function composed of a known part (*i.e.* with model-based constraints) and an unknown part (*i.e.* without model-based constraints) of the environment:

$$\mathcal{E} = \mathcal{E}_E + \lambda \mathcal{E}_M, \quad (2)$$

where \mathcal{E}_E , \mathcal{E}_M are the cost functions associated to the unknown and the known parts of the environment respectively, and λ is a weight that controls the influence of each term. The term \mathcal{E}_E is the classical reprojection error used in the bundle adjustment, defined by the (1) while the term \mathcal{E}_M includes the model-based constraints on multi-view. These model-based constraints can be of two types.

The first class constrains the trajectory of the camera through the projection, in the images, of the 3D primitives extracted from the model. These primitives are not optimized in the bundle adjustment process *i.e.* they have zero degree of freedom (DoF). Including the first constraint in the constrained bundle adjustment framework results in the unification of model-based localization and SLAM solutions. The second class of model-based constraint imposes that 3D features reconstructed by the SLAM process and associated to the object belong to the surface of the model. It results in a reduction of the DoF of the reconstructed features in the bundle adjustment. Including this second constraint in the constrained bundle adjustment framework unifies dynamically updated model-based localization with SLAM solutions.

The proposed constrained bundle adjustment framework requires additional steps to establish the model-based constraints. Features (2D or 3D features depending on the chosen constraints) have to be associated to the model as described in Section 4.2.1 and robust estimation has to be used to deal with wrong data-to-model associations. Moreover, the introduced compound cost functions require to balance the influence of each term during their minimization. Note that we privileged constraints expressed in pixel units to facilitate the weighting of the resulting compound cost function. These additional steps are described below.

4.2 Additional steps of constrained bundle adjustment

In this section, we describe the additional steps introduced by the proposed constrained bundle adjustment framework. It is an iterative process that includes data-to-model associations, robust estimation to deal with wrong associations and a dedicated weighting scheme to trade off the two terms of the resulting compound cost function.

4.2.1 Data-to-model associations

To establish the model-based constraints (represented by the term \mathcal{E}_M in (2)) in the bundle adjustment, 2D features extracted from the images (localization constraints) or 3D features estimated by the SLAM process (reconstruction constraints) have to be associated to the 3D model of the object of interest. In the first case it results in 2D/3D correspondences while in the second case 3D/3D correspondences are obtained.

For the two examples of model-based constraints presented in this paper, we describe the 3D/2D associations step between edges extracted in the images and 3D segments of the model in Section 5.1.3 and the 3D/3D associations process between points reconstructed by the SLAM process and planes of the model in Section 5.2.3.

4.2.2 Robust estimation

Inaccuracies in the coordinates frame registration and in the SLAM reconstruction introduce many wrong data-to-model associations that will fail the optimization process. To deal with those outliers a robust estimation is used through the Geman-McClure M-estimator $\rho(r, c) : \mathbb{R} \rightarrow [0 \cdot \cdot 1]$ where

$$\rho(r, c) = \frac{r^2}{r^2 + c^2}, \quad (3)$$

with, r is a residual error of any cost functions described in this document by (1), (8) or (6), and c is the rejecting threshold. It is automatically estimated with the Median of Absolute Deviation (MAD) such as $c = \text{median}(\mathbf{r}) + 1.4826\text{MAD}(\mathbf{r})$ where \mathbf{r} is a vector concatenating the residuals of any cost function. Note that the MAD assumes a normal distribution of the residuals.

$$\text{MAD} = \text{median}_i(|\mathbf{r}_i - \text{median}_j(\mathbf{r}_j)|). \quad (4)$$

Equation (2) can be rewritten as follows with $\lambda = 1$:

$$\mathcal{E} = \rho(\mathcal{E}_E, c_1) + \rho(\mathcal{E}_M, c_2), \quad (5)$$

the M-estimator is applied to the two terms of the function because it normalizes the residuals. In the rest of the paper we discard the weighting parameter λ and control the influence of each term directly through the rejecting thresholds.

4.2.3 Weighting through robust estimation

The two cost functions in (2) share the same unit, which is the pixel. On the other hand, they do not necessarily share the same magnitude. The error residuals associated to the known part of the environment have generally higher values. This makes their combination less trivial than expected thus we model the known and unknown parts of the environment in a typical bi-objective least square problem.

One challenging issue in bi-objective minimization is to control the influence of each term. This is usually done through a weighting parameter that is fixed experimentally or via cross-validation [10]. We propose a simple alternative: the influence of each term is directly controlled through the rejecting threshold of the robust estimator. We have seen above that a robust estimator have to be used to deal with wrong data-to-model associations *i.e.* for the cost functions of the known part of the environment. We also apply the robust estimation

to the cost functions of the unknown part of the environment since the Geman-McClure M-estimator normalizes the residual.

Thus there are several possibilities to control the influence of each term through the rejecting threshold. We will explore three of them:

- combination 1: $c_1 = c_{Env}$ and $c_2 = c_{Model}$
- combination 2: $c_1 = c_2 = c_{All}$
- combination 3: $c_1 = c_2 = \max(c_{Model}, c_{Env})$

where, c_{Model} is the rejecting threshold estimated on the model-based residuals such as those used in (6), (8), c_{Env} is the rejecting threshold estimated on the residuals of the unknown parts of the environment such as those used in (1) and c_{All} is the rejecting threshold estimated on all residuals.

The first difference between these combinations is that the combination 2 assumes that residuals of known and unknown parts of the environment share approximately the same normal distribution, while combinations 1 and 3 do not assume such hypothesis. The second difference is that the combinations 2 and 3 use the same rejecting threshold for all the residuals while combination 1 considers one rejecting threshold for each kind of residual. The combination 3 computes this threshold on the part of the environment that has the highest residual values. It is most of the time the known part of the environment, expected when the object of interest is hidden or not in the field of view of the camera. Thus, combination 3 will usually favor the model-based constraints during the optimization process while guaranteeing that the multi-view relationship of the unknown part of the environment are still verified. These three combinations are evaluated and compared on synthetic data in Section 6.1.2.

5 The model-based constraints

In this section, we describe one example of localization constraint and one example of reconstruction constraint. The former is achieved through a line constraint (Section 5.1) and is well suited to deal with textureless objects whereas the latter uses a planar constraint (Section 5.2) applied on some 3D points reconstructed by the SLAM process. The planar constraint requires a textured object since 3D points associated to the object of interest have to be reconstructed by the SLAM process to establish the constraint. We particularly take care in their formulation and integration in a local bundle adjustment implementation to maintain real-time performances.

5.1 SLAM constrained in localization

5.1.1 Line constraint

For textureless objects, we propose to use model-based constraints provided by the sharp edges of the 3D model (*i.e.*, for a polygonal model, edges formed by two triangles and whose dihedral angle is inferior to a certain threshold).

Similar to [8], these sharp edges are identified from their dihedral angle. However, such strategy can result in a large number of irrelevant sharp edges. Indeed, small elements,

such as screw, can generate a large number of sharp edges (eg. the thread area of a screw) whereas these edges are useless for the tracking. Moreover, some edges can be located inside the object and will never be visible during the tracking. To prevent these problems, a graph algorithm is used to identify the connected components among the sharp edges. The connected components with a small bounding box are then removed. Then, the visibility of each remaining edge is evaluated from a sphere of view. The edges whose visibility is null or poor among these viewpoints are then removed. The resulting connected components are finally run with a depth-first search algorithm and sampled with a regular step into a set of infinitesimal 3D segments $\{\mathbf{L}_i\}_{i=1}^s$. Each infinitesimal segment \mathbf{L}_i , referred to as edgelet, is parametrized by a center point \mathbf{M}_i and a direction \mathbf{D}_i . This set of edgelets $\{\mathbf{L}_i\}_{i=1}^s$ constitutes the known part of the environment. Segments have previously been used by Klein et al. in [18] to improve the agility of keyframe-based SLAM algorithms against rapid camera motions. They propose a whole process, including edges extraction and matching on consecutive keyframes, edge triangulation and a dedicated Bundle Adjustment to combine point and edge features. In our cases, we already have a map of 3D segments provided by the geometric model of the known object. We use this available information to improve the accuracy of keyframe-based SLAM algorithms.

5.1.2 The cost function

The cost function of the line constraints is an extension of classical edge based tracking criterion to the mutli-view case. It minimizes the orthogonal distance between the projection of the segment’s center point $\mathbf{P}_j\mathbf{M}_i$ and its associated edge feature $\mathbf{m}_{i,j}$ in the image, see e. g. [56] for more details. Let’s assume 3D/2D associations between the 3D segments \mathbf{L}_i of the model and edges $\mathbf{m}_{i,j}$ extracted at keyframes $j \in \mathcal{S}_i$ (see Section 5.1.3), the line constraints are given by:

$$\mathcal{E}_M \left(\{\mathbf{R}_j, \mathbf{t}_j\}_{j=1}^m \right) = \sum_{i=1}^s \sum_{j \in \mathcal{S}_i} |\mathbf{n}_{i,j} \cdot (\mathbf{m}_{i,j} - \mathbf{P}_j\mathbf{M}_i)|, \tag{6}$$

where $\mathbf{n}_{i,j}$ is the normal of the projected direction $\mathbf{P}_j\mathbf{D}_i$. Note that, this cost function depends only on the camera parameters. It is similar to the cost function of model based tracking solutions (e.g. [56]) but extended to the multi-view case. All the 3D points reconstructed by the SLAM belong to the unknown part of the environment for the BA with line constraints.

5.1.3 3D/2D associations between edges and 3D segments

For (6), 3D segments extracted from the model have to be associated to edges in the image. A visibility test is first used to get only the subset of visible 3D segments for each keyframe of the BA. The visible 3D segments are then projected in each keyframe and a one-dimensional search is performed to find gradient maxima along the normal of the projected segments. In practice, we find that keeping only the nearest edge with an almost similar orientation is sufficient since the initial poses (i.e. before the BA) estimated by the SLAM process at each keyframe are good estimates.

Table 1 Bundle adjustment with line constraints

- Visibility test to find a subset of visible 3D segments for each keyframe $j \in \mathcal{S}_i$.
- Projection of the visible 3D segments \mathbf{L}_i in each keyframe $j \in \mathcal{S}_i$.
- Association of 3D segments \mathbf{L}_i to image edges $\mathbf{m}_{i,j}$.
- Compute the rejecting thresholds c_1 and c_2 .
- Minimization of (7) with the Levenberg Marquardt (LM) algorithm [32].

The steps of the bundle adjustments with line constraints are summarized in Table 1. The resulting compound cost function of the constrained bundle adjustment with line constraints is then given by (7).

$$\begin{aligned}
 \mathcal{E} \left(\{\mathbf{R}_j, \mathbf{t}_j\}_{j=1}^m, \{\mathbf{Q}_i\}_{i=1}^N \right) &= \underbrace{\sum_{i=1}^N \sum_{j \in \mathcal{A}_i} \rho \left(d^2(\mathbf{q}_{i,j}, \mathbf{P}_j \mathbf{Q}_i), c_1 \right)}_{\text{Unknown part of the environment } (\mathcal{E}_E)} \\
 &+ \underbrace{\sum_{i=1}^l \sum_{j \in \mathcal{S}_i} \rho \left(|\mathbf{n}_{i,j} \cdot (\mathbf{m}_{i,j} - \mathbf{P}_j \mathbf{M}_i)|, c_2 \right)}_{\text{Known part of the environment } (\mathcal{E}_M)}. \tag{7}
 \end{aligned}$$

5.2 SLAM constrained in reconstruction

5.2.1 Planar constraint

Let’s assume that some 3D points reconstructed by the SLAM algorithm have been previously associated to the object of interest, see Section 5.2.3 for details. We note \mathcal{M} this set of 3D points indexes that will be used in the planar constraints and \mathcal{U} the set of remaining 3D point indexes that constitute the unknown part of the environment, with $\text{card}(\mathcal{M}) + \text{card}(\mathcal{U}) = N$. Some previous works take benefit of the piecewise planar structure of the observed scene to improve the quality of SLAM algorithms, e.g. [1, 25, 42, 47]. However we propose an alternative planar constraint that is more relevant for real-time performance in a local bundle adjustment framework.

5.2.2 The proposed cost function

Without lost of generality, considering an object model defined by a set of facets (plane π_i). The main idea is that a 3D point \mathbf{Q}_i lying on a plane π_i has only two degrees of freedom. Lets \mathbf{M}^{π_i} the transfer matrix between the coordinate frame of plane π_i and the world coordinate frame attached to the object, then $\mathbf{Q}_i = \mathbf{M}^{\pi_i} \mathbf{Q}_i^{\pi_i}$, where $\mathbf{Q}_i^{\pi_i} = (X^{\pi_i}, Y^{\pi_i}, 0, 1)$ and (X^{π_i}, Y^{π_i}) are the coordinates of \mathbf{Q}_i in coordinate frame of plane π_i . This relation can be used for planar constraints in a BA by minimizing the following cost function:

$$\mathcal{E}_M \left(\{\mathbf{R}_j, \mathbf{t}_j\}_{j=1}^m, \{\mathbf{Q}_i^{\pi_i}\}_{i \in \mathcal{M}} \right) = \sum_{i \in \mathcal{M}} \sum_{j \in \mathcal{A}_i} d^2(\mathbf{q}_{i,j}, \mathbf{P}_j \mathbf{M}^{\pi_i} \mathbf{Q}_i^{\pi_i}). \tag{8}$$

Note that the matrices \mathbf{M}^{π} associated to each plan of the model can be precomputed. In practice, the reconstructed 3D points do not exactly belong to the planes of the model. A

Table 2 Bundle adjustment with planar constraints

- Association of the 3D points $(\mathbf{Q}_i)_{i \in \mathcal{U} \cup \mathcal{M}}$ to the known or the unknown parts of the environment.
- Projecting the 3D points $(\mathbf{Q}_i)_{i \in \mathcal{M}}$ on their associated plane π_i .
- Compute the rejecting thresholds c_1 and c_2 .
- Minimization of (9) with the Levenberg Marquardt (LM) algorithm [32].
- Re-triangulation of 3D points $(\mathbf{Q}_i)_{i \in \mathcal{M}}$ to re-examine the data-to-model association for the next bundle.

preliminary step is then required to project each 3D point \mathbf{Q}_i on its associated plane $\{\pi_i\}_{i \in \mathcal{M}}$ (see Section 5.2.3 for more details).

5.2.3 3D/3D associations between points and planes

For (8), a preliminary step is required to decide which 3D points \mathbf{Q}_i of the SLAM reconstruction belong to the model. This point-to-model association problem is achieved through ray tracing from the different observations $\{\mathbf{q}_{i,j}\}_{j \in A_i}$ of \mathbf{Q}_i . Thus, $\text{card}(A_i)$ votes are obtained. When 3D points are assigned to different planes by different camera poses (for example 3D points near the boundaries) we take the majority choice. This association step classifies which 3D points belong to the known or the unknown parts of the environment. Note that 3D points that have been associated to one plane π_i of the model have to be projected on it before minimizing (8) and are re-triangulated after the optimization to re-examine the point-to-model associations for the next bundle.

The steps of the bundle adjustments with planar constraints are recapitulated in Table 2. The resulting compound cost function of the constrained bundle adjustments with planar constraints is then given by (9).

$$\begin{aligned}
 \mathcal{E} \left(\{\mathbf{R}_j, \mathbf{t}_j\}_{j=1}^{N_c}, \{\mathbf{Q}_i\}_{i \in \mathcal{U}}, \{\mathbf{Q}_i^{\pi_i}\}_{i \in \mathcal{M}} \right) &= \underbrace{\sum_{i \in \mathcal{U}} \sum_{j \in A_i} \rho \left(d^2(\mathbf{q}_{i,j}, \mathbf{P}_j \mathbf{Q}_i), c_1 \right)}_{\text{Unknown part of the environment } (\mathcal{E}_E)} \\
 &+ \underbrace{\sum_{i \in \mathcal{M}} \sum_{j \in A_i} \rho \left(d^2(\mathbf{q}_{i,j}, \mathbf{P}_j \mathbf{M}^{\pi_i} \mathbf{Q}_i^{\pi_i}), c_2 \right)}_{\text{Known part of the environment } (\mathcal{E}_M)} \quad (9)
 \end{aligned}$$

6 Experimental results on synthetic and real data

The quality of a tracking method depends on several criteria:

- A quality result criterion: accuracy, stability, robustness to movement, to occlusions, illumination conditions, *etc.*
- A easy to deploy criterion: robustness to inaccurate initialization, inaccurate 3D model, *etc.*
- A performance criterion: processing time (real-time), ability to handle complex models, *etc.*

In this section we evaluate the proposed constrained bundle adjustment framework on synthetic and real data. To evaluate it with respect to criteria mentioned above, different series

of experiments are conducted. We use the keyframe-based SLAM algorithm described in [33]. It is an on-line localization algorithm that achieves real-time performances through local bundle adjustment applied on a sliding window of triplets of keyframes. At each keyframe, only the poses associated to the three last keyframes and the 3D points they observed are optimized.

We implement and compare three local bundle adjustment algorithms:

- The original one described in [33] called LBA_E in the rest of the paper. It minimizes (1) by the Levenberg-Marquardt algorithm.
- The proposed LBA_LC&E algorithm. It minimizes (7) that takes line constraints into account with the procedure described in Table 1.
- The proposed LBA_PC&E algorithm⁴. It minimizes (9) that takes planar constraints into account with the procedure described in Table 2.

Real-time performances are obtained by taking advantage of the sparse block structure of the normal equations as described in [50]. Note that we also modify the initialization step of [33]. As proposed in [3], we use an approximate pose (*e. g.* obtained by model-based solutions) that roughly registers the world and the 3D model coordinate frames, the initial 3D point cloud is obtained through back-projection of the observations extracted on the first frames.

6.1 Evaluation on synthetic data

In this section we compare the three local bundle adjustment algorithms described above LBA_E, LBA_LC&E and LBA_PC&E on a synthetic sequence generated with a 3D computer graphics software. We firstly describe the synthetic sequence. Then we compare the three combinations proposed in Section 4.2.3 for the LBA_LC&E and the LBA_PC&E algorithms on this sequence. Finally, we evaluate these three algorithms, in terms of accuracy and robustness to inaccurate initialization. The quality of the resulting localization is measured as the 3D RMS at keyframes between the ground truth and the on-line estimated poses after local bundle adjustment.

6.1.1 The cubes sequence

In this sequence, illustrated in Fig. 1, the scene is composed of two cubes over a textured ground with other smaller cubes that partially occult them. The object of interest, *i. e.* for which a 3D model is available, is composed by the two main cubes of one meter side length. The camera trajectory is a circle of 3 meters radius around the main cubes.

6.1.2 Combination choices

We compare for the LBA_LC&E and the LBA_PC&E algorithms the three combinations proposed in Section 4.2.3 on the "cubes sequence". For this experiment, the initial camera pose, associated to the first frame is given by the ground truth. Localization results are represented in Fig. 2 (right). The combination 2 presents the worst results in terms of accu-

⁴LC means Lines Constraints, PC means Planar Constraints for the model constraints, *i. e.* the known part of the environment and E means that the multi-view relationship of the unknown part of the environment are taken into account.

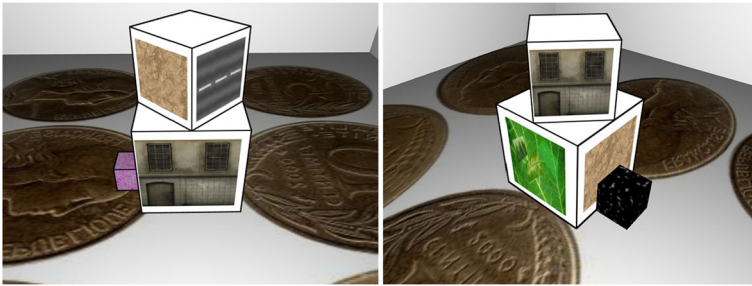


Fig. 1 Two images of the "cubes sequence"

racy. This can be explained by the fact that the robust estimator threshold is estimated for an unimodal distribution of the residuals. It appears to be a wrong assumption in practice as seen on Fig. 2 (left). As expected the magnitude of the residual errors associated to the known part (reprojection error associated to the constraints, (6) and (8)) of the environment are usually higher than those associated to the unknown one (classic reprojection error, (1)). Thus the combination 2 yields in an underestimated rejecting threshold that will discard most of the model-based residuals in the optimization. Finally, the best combination is the third one. This proves that the model-based constraints has to be favored during the optimization process while guaranteeing that the multi-view relationships of the unknown part of the environment are still verified. Note that similar results have been obtained with the LBA_PC&E algorithm and are presented in Fig. 3. In the rest of the paper we only consider the combination 3 for (9) and (7).

6.1.3 Accuracy comparison of the three LBA algorithms

The first experiment evaluates the accuracy of the resulting localization of the three algorithms for a perfect initialization given by the ground truth. Results in terms of accuracy are shown on Fig. 4. The LBA_E algorithm is subject to error accumulation: position and orientation errors are up to 50 cm and 5 degrees respectively at the end of the sequence. Constrained local bundle adjustments are drift free, position and orientation errors do not increase over time and are close to zero. LBA_LC&E and LBA_PC&E are as accurate on this sequence. Adding the model-based constraints improve drastically the accuracy of SLAM

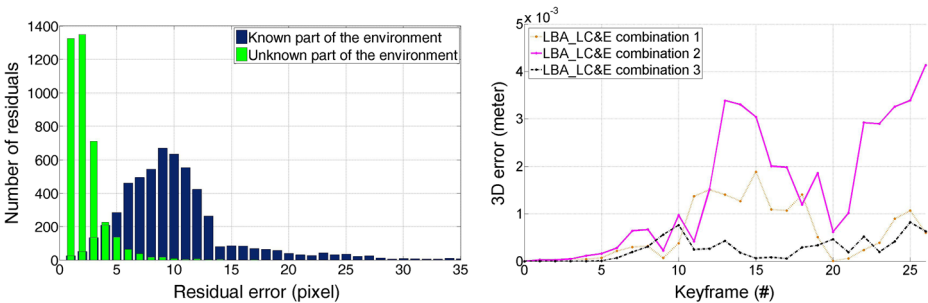


Fig. 2 Left: residual errors distributions for the line constraint algorithm. In green, (resp. in blue) the distribution of the residual errors associated to the unknown part (resp. the known part) of the environment. Right: errors in position (resp. orientation) expressed in meter (resp. degree) for the different combinations

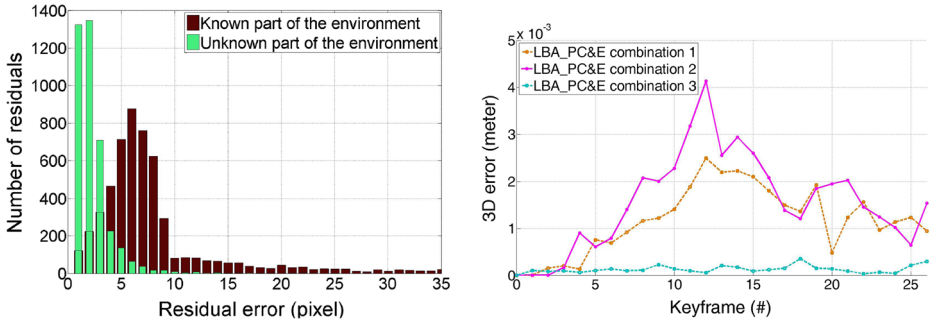


Fig. 3 Left: residual errors distributions for the planar constraint algorithm. In green, (resp. in red) the distribution of the residual errors associated to the unknown part (resp. the known part) of the environment. Right: errors in position (resp. orientation) expressed in meter (resp. degree) for the different combinations

localization. Note that, this figure also presents results obtained with a classic model-based tracker [56] which is subject to jitter.

6.1.4 Robustness to an inaccurate initialization

The second experiment, evaluates the robustness of the constrained LBA to inaccurate initialization. Perturbations with increasing magnitudes are performed on the camera pose at first frame. Their amplitudes fluctuate between 1 % and 6 % of the circle radius formed by the camera trajectory. The results are averaged over 10 random trials. Figure 5 shows that both constrained local bundle adjustments deal with inaccurate initialization: after a certain amount of camera poses the 3D errors are stabilized to small values. Note that the LBA_PC&E algorithm converges faster than the LBA_LC&E one, on this sequence. This can not be taken as an absolute truth. It depends mainly on the sequence and the 3D model complexity. This sequence is clearly more subject to wrong 2D/3D associations than 3D/3D ones. We do not present results against initialization inaccuracy for the LBA_E algorithm since it is obvious that it has no reason to reduce the starting error.

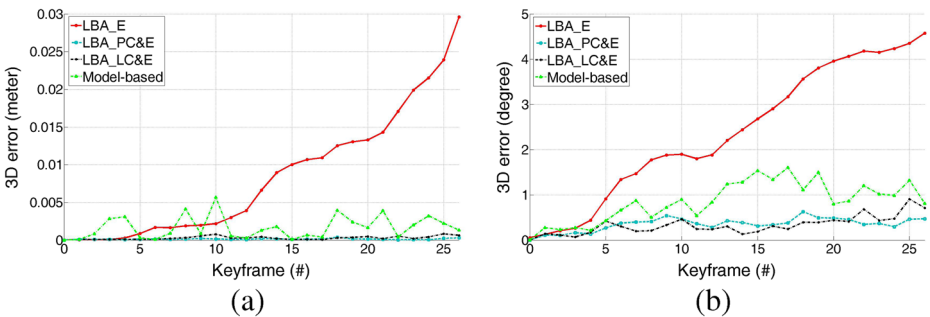


Fig. 4 a (resp. b): Errors in position (resp. orientation) for the different LBA algorithms and a classic model-based tracker [56]

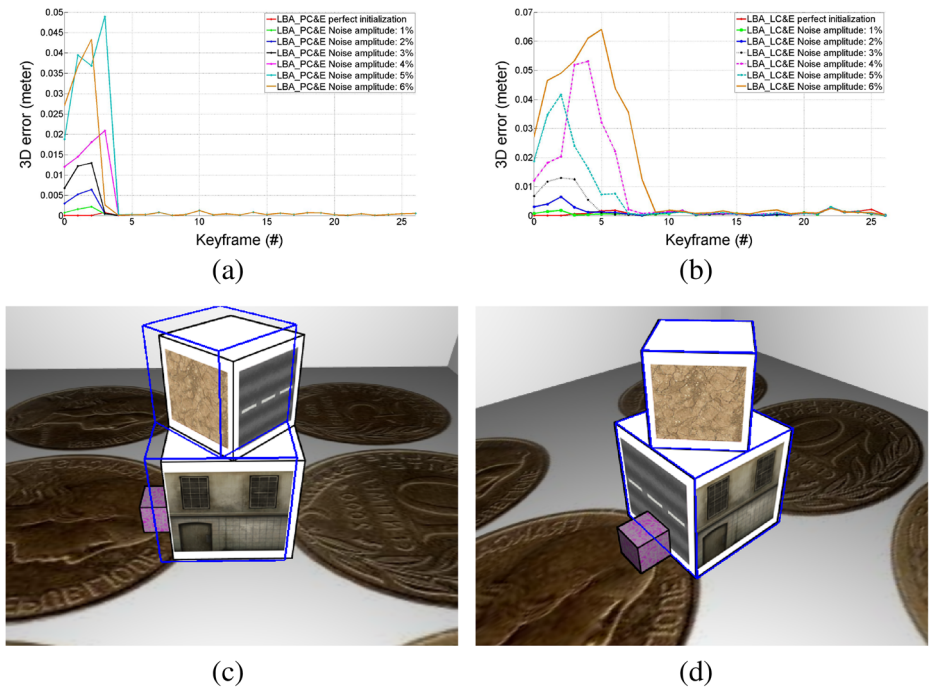


Fig. 5 a (resp. b) Results obtained by the LBA_PC&E (resp. LBA_LC&E) algorithm for different magnitudes of perturbation ($\in [1\% \dots 6\%]$ of the radius of the camera trajectory) applied on the first camera pose. c Illustration of the discrepancy error in the image between the reprojected model and the object of interest for a perturbation magnitude of 6% and d the resulting pose estimated after few frames

6.2 Evaluation on real data

In this section we evaluate on real data the proposed constrained local bundle adjustments for on-line localization of a camera in a partially known environment. This is also illustrated for augmented reality applications on textured or textureless 3D objects in the Section 8. The sequences have been acquired with a low-cost IEEE1394 GUPPY camera providing (640×480) images at 30 frames per second.

6.2.1 Tracking of a textureless 3D object

Comparison constrained SLAM vs model-based tracking. The object of interest is a textureless toy car representing a Lamborghini Gallardo. The 3D model used in our experiments is composed of 14000 facets. We extract 1186 3D segments from this model as seen in Fig. 6. We compare SLAM that includes the local bundle adjustment with line constraints algorithm LBA_LC&E with a classic model-based tracker which, by definition, uses only the known part of the environment. This latter is an improved version of the model-based tracker of [8] that includes multiple hypothesis for data associations as in [52, 57]. The comparison is done on a challenging sequence that presents large variations in scale, fast motions, partial occlusions of the toy car, etc.

Figure 7 presents the results obtained by the sequential SLAM with the LBA_LC&E refinement algorithm and the model-based tracker on this sequence. The coordinate frames

Fig. 6 The 3D segments extracted from the Lamborghini model



registration on the first frame is not accurate due to the fact that the coded marker is roughly positioned near to the car. The both methods manage to correct this registration error after



Fig. 7 Localization in a partially known environment composed of a textureless 3D object. Left: results obtained with a model-based tracker similar to the one proposed in [8]. Right: results obtained with a sequential SLAM that uses the proposed LBA_LC&E algorithm

few frames: the front and the back of the car are well projected on the images as seen in Fig. 7 (left).

On the other hand the sequential SLAM with the LBA_LC&E refinement algorithm outperforms the model-based tracker. In fact, the latter failed when fast motion occurs due to a small convergence basin of contour tracker as seen in Fig. 7 (right). Moreover its resulting localization is subject to jitter. Our proposed localization algorithm uses both the informations providing by the known and the unknown parts of the environment. It yields an accurate, stable and robust localization.

Comparison constrained SLAM vs SLAM. The object of interest is a bogie (a chassis) of a train (about 3 meters). The 3D geometric model used in the experiments consists of approximately 30,000 triangles. 2,000 3D segments are then extracted therefrom. The rest of the room is the unknown part of the environment. A comparison is made between the SLAM with classic bundle adjustment noted LBA_E and the proposed refinement algorithm LBA_LC&E. The comparison is carried out on a difficult sequence which has significant illumination variations and partial occlusions of the bogie. Note that as the object of interest is slightly textured, the initialization (that is to say, the registration on the first image) is roughly obtained by positioning approximately, on the first image, a coded marker on the ground near the bogie. The results are shown in Fig. 8. SLAM with the refinement algorithm LBA_E do not correct the initialization error and is subject to accumulation of errors (Fig. 8 at left) whereas with the local constrained bundle adjustment LBA_LC&E, after few frames the initial error are removed and the localization does not drift over time (Fig. 8 at right). Adding the model-based constraint significantly improves the accuracy of the localization.

6.2.2 Tracking of a textured 3D object

In this section we compare three local bundle adjustments (LBA) :

- the classic LBA defined by (1) (LBA_E),
- the LBA only constrained by plan model defined by (8) (LBA_PC),
- the LBA minimizing the plan model constraints and the multi view relationships of the whole scene, defined by (9) (LBA_PC&E).

The objective is to demonstrate that using the multi-view relationships of the whole scene improves the accuracy and the robustness compared to methods that use only the known part of the environment (model-based localization), *e. g.* [25] and those that do not use model constraints (classic SLAM) *e. g.* [33]. The object of interest is a toy car representing a Citroen C4 with a 3D model composed of 1,600 triangles. The evaluation is done on a real sequence that presents large variations in scale, fast motions, lighting variations, partial and total occlusions, etc. Coordinate frame registration on the first frame is performed by matching it with a keyframe registered offline on the model.

Results. Figure 10 presents the results obtained by the three local bundle adjustment algorithms. The coordinate frame registration seems accurate on the first frame (the model is well projected as seen in Fig. 9) but after turning around the car we observe that this is not really the case. The LBA_E refinement algorithm can not correct this inaccuracy as seen on Fig. 10 (top left) and thus will keep it (and probably augment it) during the whole sequence. LBA_PC&E and LBA_PC algorithms manage to correct this registration error after few frames: the front and the back of the car are perfectly projected on the images. On the other hand the LBA_PC&E algorithm outperforms the LBA_PC one when the object is occluded

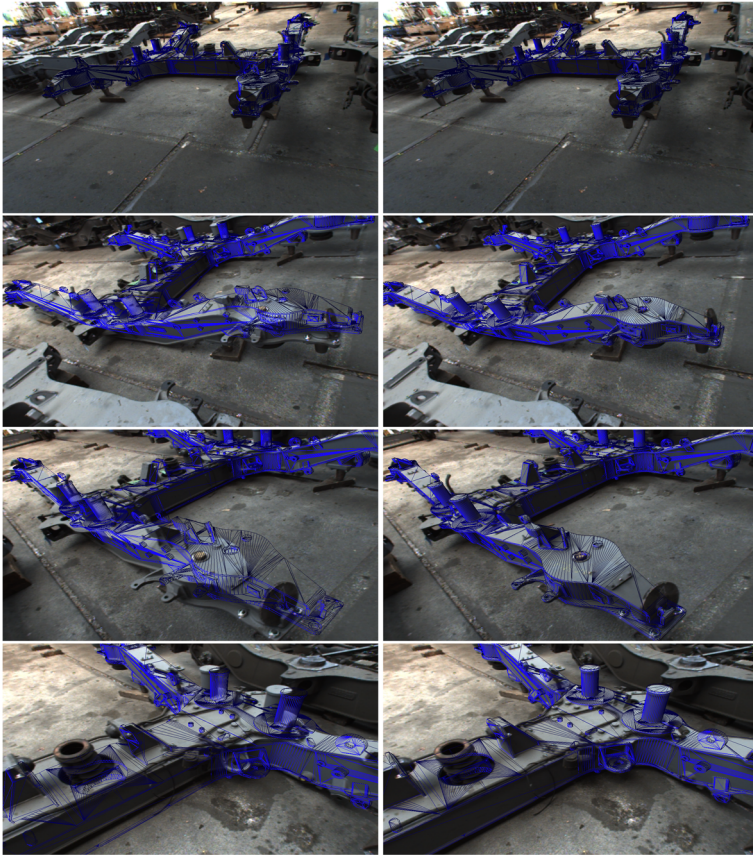


Fig. 8 Comparison of localization results obtained with the classic SLAM (*left*) and SLAM constrained by 3D segments (*right*)

or takes a small part in the images as seen in Fig. 10 (right). It successfully manages to localize, in real-time, the toy car during the whole sequence. Combining the information providing by the known and the unknown parts of the environment yields an accurate and a robust localization.

Fig. 9 The coordinate frames registration on the first frame. From this point of view the pose seems to be correct since the model is well projected on the car



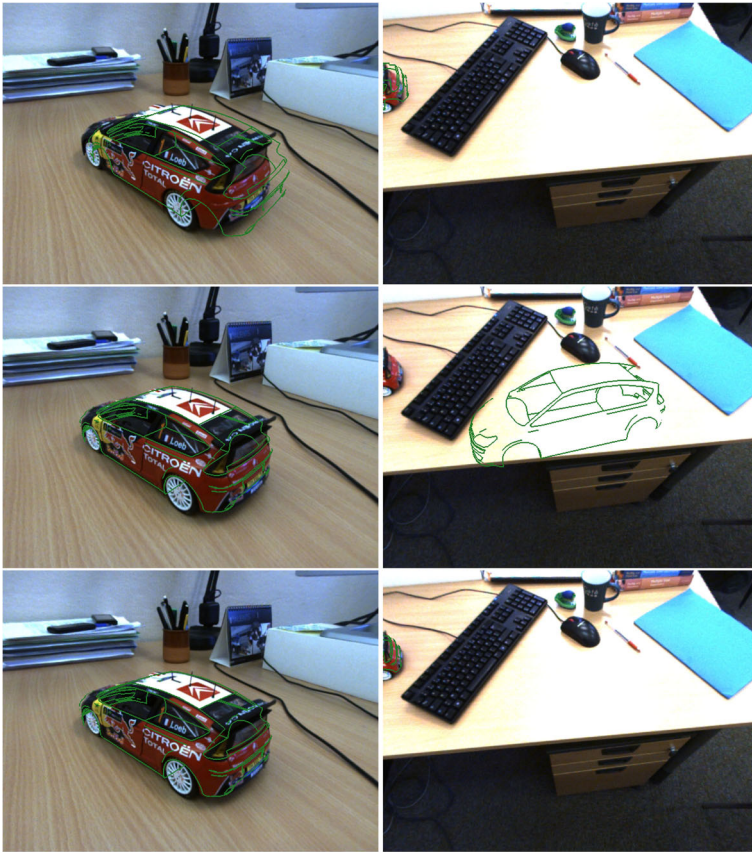


Fig. 10 Localization in a partially known environment composed of a textured object. Top, Middle, Bottom: results obtained with the LBA.E, LBA.PC and LBA.PC&E refinement algorithms respectively

6.3 Robustness to model inaccuracies

This section describes a method for the 3D reconstruction of an unknown object. We also compare the registration quality obtained by the proposed tracking solution using a CAD model and the model reconstructed with our framework.

6.3.1 Easy and fast 3D object modeling

Our tracking algorithm requires a 3D model to estimate the camera trajectory. If an accurate CAD model is preferable, it is frequently unavailable: they might not exist (eg. craft or artistic object) or can be unreachable (eg. restricted access due to confidentiality). In these cases, our framework includes an application that allows to reconstruct a 3D mesh of the object on the fly. As illustrated in Fig. 11, this application is a three steps process:

- Initial reconstruction: provides an initial reconstruction of the object and its surrounding environment;

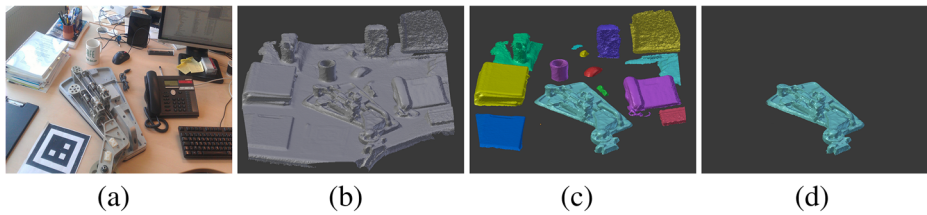


Fig. 11 The different steps of the reconstruction process. **a** the scene including the object of interest (part of an exo-skeleton). **b** The initial reconstruction (1.17 million polygons). **c** Output of the object segmentation step (each cluster is represented with a different color). **d** Best cluster after simplification (30 000 polygons)

- Object segmentation: extracts the 3D object mesh from the initial reconstruction and defines its coordinate frame;
- Model simplification: realizes a simplification of the object mesh to provide a tracking model compatible with real-time performances.

The first step relies on the KinectFusion process [35]. This solution has been selected for its deployment speed and the quality of the reconstruction it provides without requiring neither user expertise nor expensive hardware. Moreover, this solution is able to reconstruct a large variety of objects, including textureless objects, and can be extended to cover large volume [54, 55]. To be exploitable by the tracking process, the surrounding environment must be removed from the model, remaining only the geometry of the object of interest. To achieve this task, the second step of the modeling process assumes that the object of interest lies on a flat surface. Because the scene may include several planar surfaces, the user identifies this particular plan by putting a 2D marker (eg. a coded target) on its surface. Assuming this marker was observed by the 3D camera during the initial reconstruction step, its location and orientation in the reconstructed model coordinate frame can be estimate [46]. The reconstruction can then be expressed in the coordinate frame of the marker, providing a non-arbitrary coordinate frame that will also be used for the tracking. Since the equation of the planar surface is known⁵, points located under this plane are suppressed and the remaining points are partitioned into entities by a clustering algorithm. The object model is then identified as the most frequently observed cluster during the first reconstruction step. The ultimate step consists in simplifying the object model to reach a number of polygons compatible with real-time performances. This is achieved using a Quadric Edge Collapse decimation algorithm.

6.3.2 CAD model vs. reconstructed model for the line constraint

Obviously, the 3D model provided by a reconstruction process is subject to artifacts. Indeed, even if the reconstruction is relatively accurate (Fig. 14), sharp edges are rounded while reconstruction noise may induce some fictive sharp edges. The objective of this experiment is therefore to evaluate the impact of these artifacts on the localization process.

Two mechanical objects with their associated CAD model were selected for this experiment (see Fig. 13): a car cylinder head and a part of an exo-skeleton. These objects were reconstructed with the algorithm introduced in [34] using an Asus XTion 3D camera. Since

⁵The plane equation $z = 0$ can be refined by selecting the surrounding point cloud and using a plane fitting algorithm.

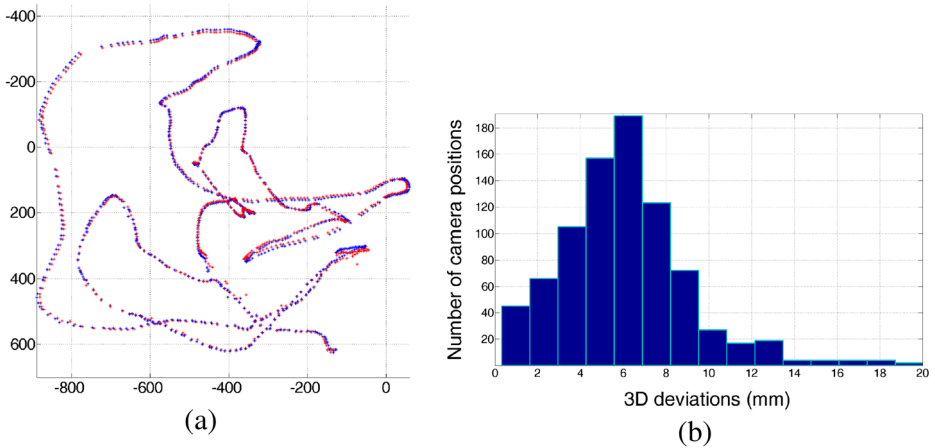


Fig. 12 Evaluation quantitative of the deviation of the reconstructed model to the CAD model. **a:** top-view of the obtained trajectories (unit: mm) on the cylinder head sequence with the CAD model (*blue stars*) and reconstructed mode (*red stars*). **b:** the distribution of 3D-errors between the trajectories

we do not dispose of a trajectory ground truth, the effects of reconstruction artifacts on the tracking process is assessed by comparing the localization of the camera reached with the CAD model and the reconstructed model. Figure 12a represents the obtained trajectories on the cylinder head sequence with the two types of models while Fig. 12b represents the distribution of the 3D deviations between the two trajectories. The mean deviation obtained on the whole sequence of the cylinder head is of 6 mm (see Fig. 12) for a 30 cm object observed from about 1m away. Note that similar results have been obtained on the exo-skeleton and Fig. 13 illustrates the registration quality for the two objects and for both kinds of model. Therefore, the impact of the reconstruction artifacts on the registration can be considered as negligible (Fig. 14).



Fig. 13 Registration with a CAD model (*first row*) and with the reconstructed model (*second row*). The 3D mesh is represented in blue

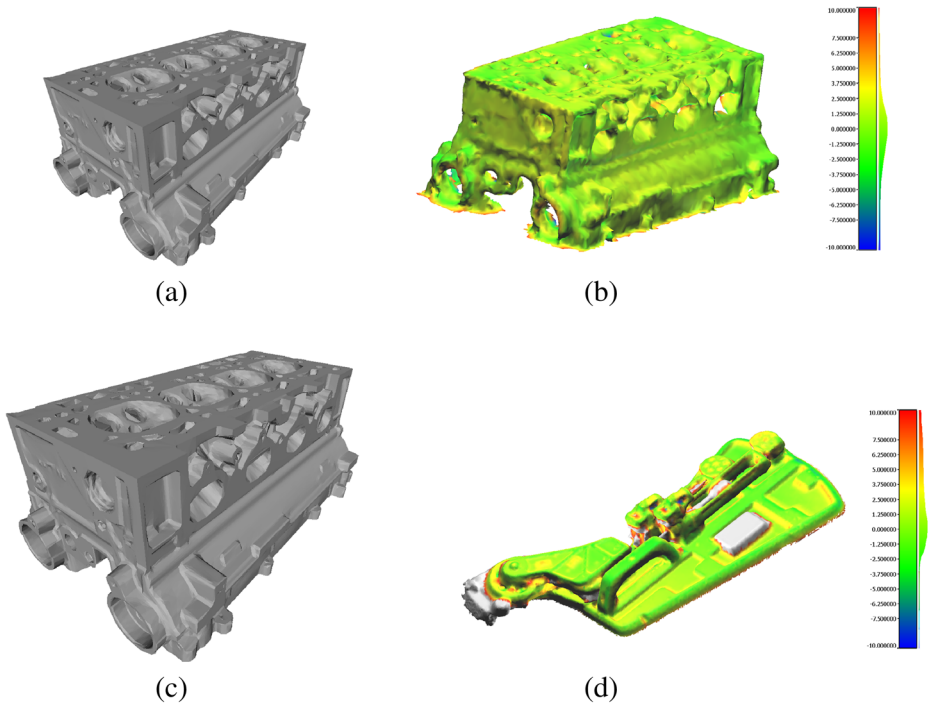


Fig. 14 Comparison of CAD model (a & c) and reconstructed models (b & d) for a cylinder head and a part of an exo-skeleton. The hue of reconstructed model represents the surface deviation to the CAD model, while gray areas represent elements that were missing in the CAD model

6.4 Bundle adjustment with multiple constraints

Some objects are difficult to handle with the constraints presented in the Section 5. For example on the miniature model of an aircraft types A320 shown in the Fig. 15, the SLAM approach with bundle adjustment LBA_LC&E fails because there are not enough 3D segments that can be extracted on the model. Indeed, the problem with the extraction step of 3D segments is that the curved parts of the model of interest does not return any segment. On the model of the plane, the 3D segments are extracted only from the lateral wings and the horizontal and vertical stabilizers. On the other hand the SLAM approach with bundle adjustment LBA_PC&E also fails in this example because there is not enough texture on the plane. The planar constraints requires textured objects since a 3D point cloud of the object must be reconstructed by the SLAM process. Combining the two constraints would theoretically solve these problems. The constrained SLAM framework proposed in this article is very flexible and allows to combine different types of constraints.

We realized an implementation of this solution which combines the line constraint with the planar one. The result is a cost function of three terms. As a first implementation, we combined them with a similar strategy that the one described in Section 4.2.3. We note that this bundle adjustment with multiple constraints, provides better results than the bundle adjustments constrained by lines or plans of the model. Indeed, as shown in the Fig. 15, both planar and line constraints do not provide sufficient constraints to avoid error accumulation:

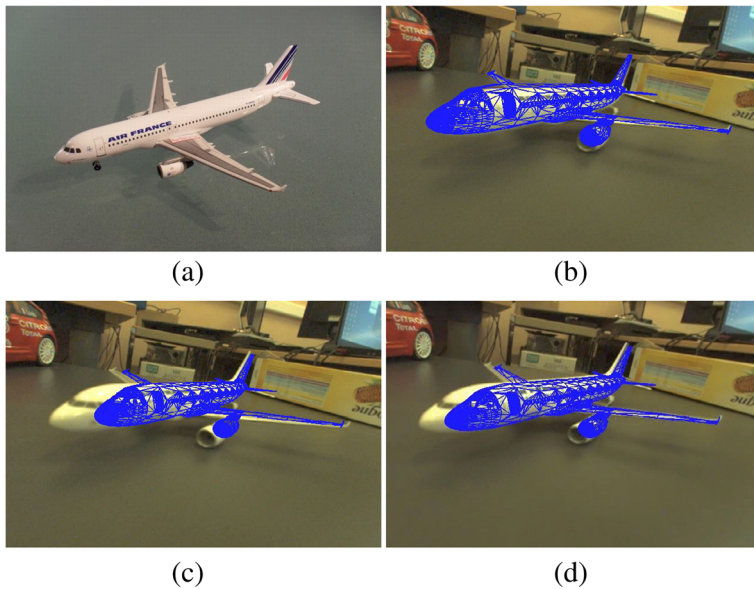


Fig. 15 Localization in a partially known environment composed of a 3D object with curved parts. **a**: the original image of the miniature aircraft. **b**: result obtained by the SLAM with both line and planar constraints for the bundle adjustment. **c**: results obtained by the SLAM with the line constrained bundle adjustment. **d** results obtained by the SLAM with the planar constrained bundle adjustment (**e**). Note that the CAD model does not exactly match the miniature aircraft used in this experiment: the two reactors have not the right scale and the horizontal stabilizers have the wrong orientation

the SLAM with the algorithm LBA_LC&E (Fig. 15c) and the SLAM with the algorithm LBA_PC&E (Fig. 15d) fail to accurately localize during the whole video sequence.

The first results obtained by combining the constraints are encouraging because this can handle a wider range of objects. However, further study is needed particularly on the weighting of multiple constraints.

7 Application to augmented reality

The resulting constrained bundle adjustment has been evaluated on different types of objects through various real-time experiments. We particularly apply the constrained SLAM framework for Augmented Reality purpose. We present one realization in the context of sales support in store or dealership and one for the industrial maintenance. The first realization is on utility vehicle as illustrated in Fig. 16. The personalization of the vehicle is done by changing, very realistically, the color of one or more elements of its bodywork. Different possible arrangements of the vehicle interior are also presented. It is a very promising market for augmented reality since this will allow sellers to show the full range of products they sell without having them all at the point of sale. The second realization is on industrial object as illustrated in Fig. 17. Different manipulations of the object elements are presented through Augmented Reality. The purpose is to illustrate for the users how to remove and eventually replace some part of the real object, *e. g.* valve, gas pipe. These two realizations enhance the accuracy and the stability of the resulting localization using our constrained



Fig. 16 Augmented Reality on a utility vehicle. *Top Left:* The tablet used for the experiments. *Top right:* The original color of the vehicle (white) is changed to green. A specific logo is also added. *Bottom:* The possible arrangements of the vehicle interior with virtual furnitures

SLAM framework. For both realizations, the LBA_{LC&E} refinement algorithm is used since the objects of interest are textureless.

These experiments differ in their complexity regarding user movement strength, important changes in scale, partial occlusions and lighting condition. This camera localization solution shows very good reliability since it deals with the problems of accurate coordinate frame registration and scale factor setting, jittering, occlusions and real-time performance.

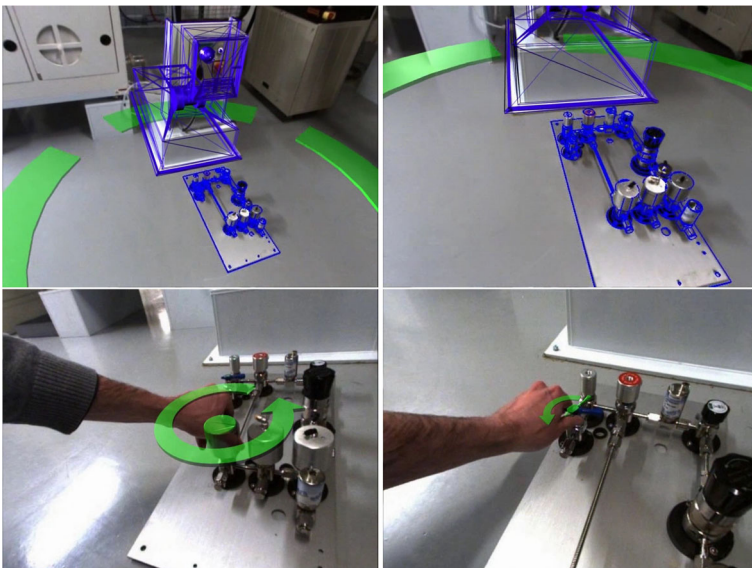


Fig. 17 Augmented Reality on an industrial object. The original object is augmented with some virtual indications for handling (removing or replacing) elements of the object

The overall system runs, on a laptop with an Intel(R) Core(TM) i7-4800MQ CPU @ 2.70GHz processor, at 11 ms with SD images (640x480) and at 15.3 ms with HD images (1280x720). It runs at 13 ms (75fps) on a tablet Windows Surface Pro 2 with an Intel(R) Core(TM) i5-4300U CPU -@1.90GHz processor, with SD images and at 25 ms with HD images. Note that, it is important for the deployment of such applications to run in real-time on a tablet.

8 An interactive augmented reality system for training applications

8.1 Prototype with optical see-through HMD

Using the tracking solution introduced in Section 4, we propose an interactive augmented reality system which allows augmenting the users vision field with virtual information co-registered with the real world. The system uses Optical See-Through head mounted displays (OST HMD) integrating a calibrated camera and allows to interactively augment an industrial object with virtual sequences designed to train a user for specific maintenance tasks. The architecture of this prototype involves two main vision-based modules: camera localization and user-interaction handling. The first module includes the proposed marker-less tracker and the second one includes fast image processing methods for laser dot or finger tracking.

One of the objectives of this prototype is to consider alternatives to touch screen-based user interfaces for Augmented Reality. Indeed, the most commonly used type of Augmented Reality is video see-through AR, where both virtual information and a video of a real object are shown on a handheld display devices. As a result, the majority of user interactions in handheld AR is realized using touch screens. Since HMD often do not have a touch screen, this raises the need for novel means to interact with real objects and digital information associated with them. Many researches have been investigated where a laser pointer or a finger is used to interact with the scene [5, 14, 15, 19, 20]. Kurz and al [20] propose to use a static calibrated camera for laser pointer tracking. Other researches employ an additional sensor like a RGBD sensor [14] or a Infra Red sensor [19] or color stickers on fingertips [15] for finger detection, in the case of mobile AR device. Hurst et al. [15] implemented various interaction capabilities on mobile phones by employing the camera of the system and color markers on fingertips.

The motivation of this work is to propose a wearable AR device which handles head tracking and interactions using only the camera affixed to the HMD. This represents a key aspect that simplifies the hardware architecture and improves the ergonomics of the system. However, the interaction module and the tracking module have to share not only the same camera but also the same CPU resources. So in order to not impact on the performance of the system, we have chosen a simple interaction module which is not computationally expensive. To validate the concept, the interactions are restricted to the 3D pointing of an element in the scene and run at less than 5 ms. We will see that this time is minimal compared to the latency time of the AR OST system presented after.

As depicted in Fig. 18, the proposed prototype features a tracked monocular OST HMD that solves the 3D registration issues and allows to accurately superimpose virtual maintenance procedures on an industrial object. The training system guides a learner step-by-step through an assembly/disassembly procedure for a specific object of interest. The user can interact with the system, by simply pointing to select a specific object component with an ordinary laser pointer or with his finger. The system is capable of interactively augmenting

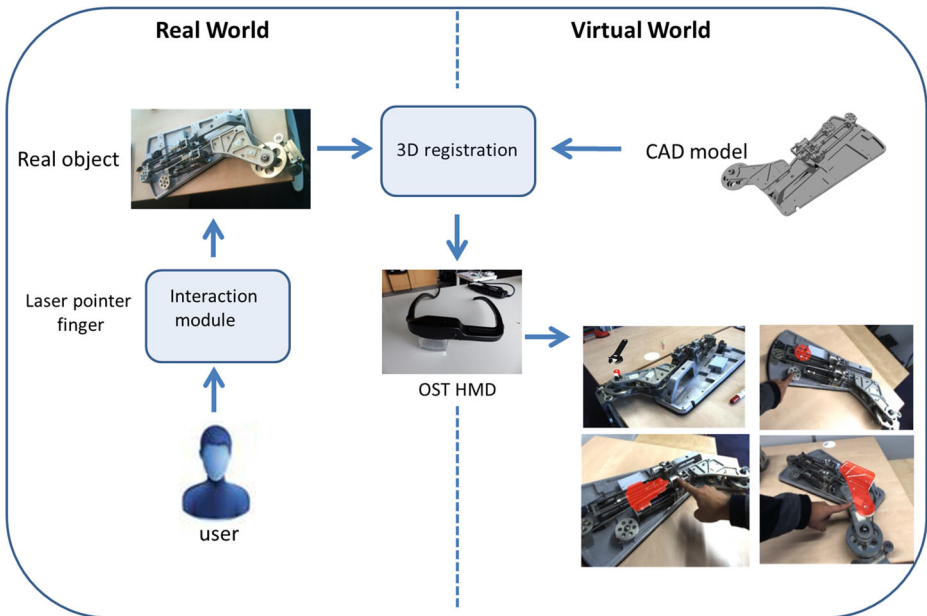


Fig. 18 The prototype architecture description

the selected component with an assembly/ disassembly animated virtual sequence by using the 3D CAD model.

The prototype incorporates a monocular OST glasses, an ordinary laser pointer (slim-line) and a laptop with an Intel(R) Core(TM) i7-4800MQ CPU @ 2.70GHz processor. The glasses are the Glasses Laster See Thru : an optical see-through technology with a Field of View (FOV) of 25° diag and a display definition of 800x600 pixels. A camera has been integrated on the Glasses Laster See Thru. This camera has a global shutter sensor so it does not suffer from motion artifacts (distortion effects) and its resolution is 640x480 pixels. The tracking process runs at 11 ms (90 fps). However, the most critical time for on OST HMD is the latency time which corresponds to the delay between the image acquisition and the resulting projection in the transparent glasses. This latency is the sum of the image acquisition time, the tracking process time and the glasses display time. We first measure the latency resulting from the image acquisition and the glasses display. By filming a chronometer and measuring the time between the image acquisition and its display on the glasses, we obtained a latency time of about 115 ms. By adding the time of the tracking process, we evaluate the latency time of the global system of about 126 ms.

The OST HMD, presented in Fig. 22 can easily display additional information such as 3D CAD objects or simple virtual animated sequences on the natural field of vision of the user. The Fig. 19 shows a picture of user's augmented view through the transparent glasses.

8.2 Interaction with augmented contents

Once the 3D position of the object is estimated by the tracking module, a vision-based process allows to interact with augmented contents. This process allows the user to interact with the system by pointing in the scene the elements for which he wishes to obtain virtual

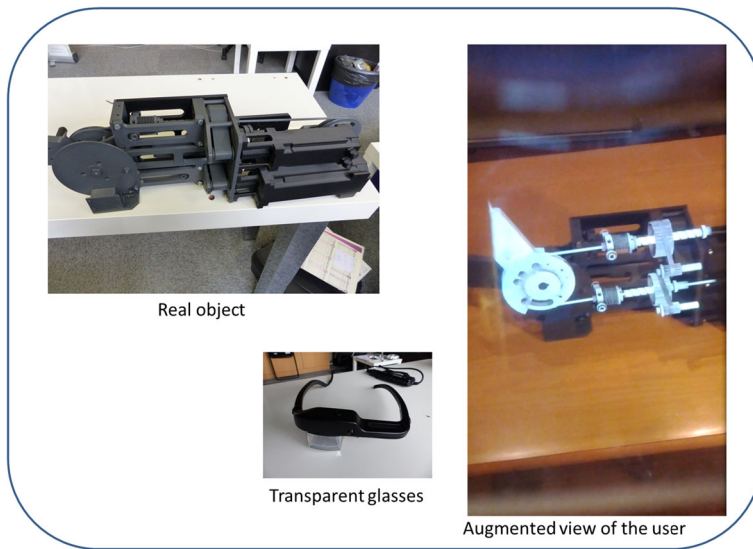


Fig. 19 The glasses (OST HMD), a workpiece and a picture of user's augmented view through the transparent glasses. This picture as been acquired by putting a camera in place of the user eye

information. This is done by pointing a laser pointer or directly with finger. The system detects the position in the image of the pointer or the finger and deduces the element pointed to by a ray tracing on the workpiece. The pointing interaction (with pointer or finger) is detected by the same camera used to localize the user wrt the object.

Once the 3D position of the object is estimated by the tracking module, a 2D Region of Interest (ROI) is defined. This ROI delimits the object area in image all the time even when the user is moving around the workpiece thanks to the tracking module. Then the red laser dot or the finger is localized in the ROI. By using the 3D CAD model and ray tracing techniques, the intersected object component with the laser spot is determined. Note that a spatio-temporal verification stage is included in order to confirm the users request. When an object-component is selected, the system superimposes co-registered instructions concerning the selected object component. For most applications, the center of the laser spot is detected from the weighted average of the image red or bright pixels. Red color detection on metallic object surfaces is challenging due to illumination changes, specular highlights and reflections. Using HSV color space is usefull to deal with these issues. Thus, a red filter in HSV color model is first applied. Then, isolated pixels were removed with a simple morphological operation. Finally, the laser dot is detected by applying a blob detection algorithm. It is important to mention, that we applied the same algorithm to detect users hand finger as a mode of interaction.

The Fig. 20, respectively Fig. 21, presents an augmentation concerning an assembling procedure that is activated in response to the users laser-based request, respectively to the user's finger designation (Fig. 22).

8.3 Usability study of the two interaction modalities

An informal usability study was conducted. The two interaction modalities of the AR prototype have been evaluated through various real-time experiments and by 8 users (members



Fig. 20 Pointer Laser Interaction

of the research team). The experiments differ in their complexity regarding the displacement magnitudes, important changes in scale. As the results of the feedback collected from users, the interaction module is easy to handle because it does not take a lot of time or effort. Besides, it proved to be intuitive and very useful for industrial maintenance training applications, specially for selecting dangerous components. Laser pointer interaction is a fast and precise technique for interacting over distances. The limitation of the finger based interaction arises when the user is pointing with more than one finger or when the user is pointing a small far component: the pointing is not very precise. Comparing with finger based interaction, the users found that laser pointing is less natural but is more precise

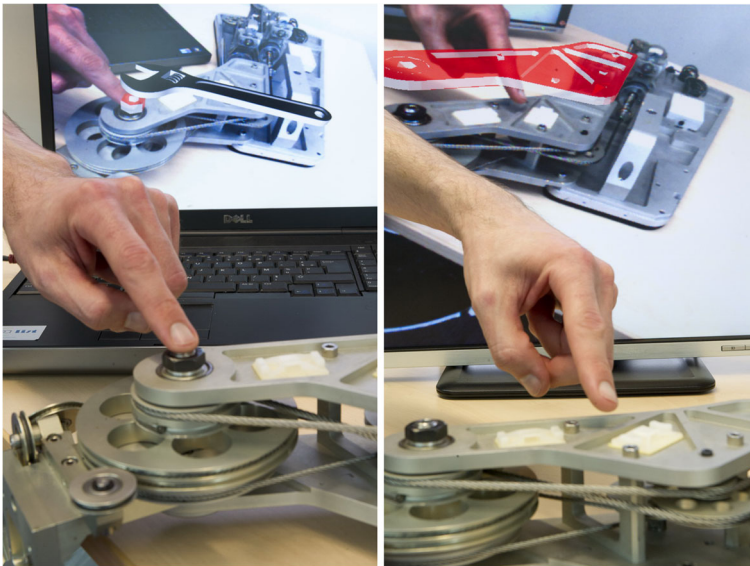


Fig. 21 3D augmentation concerning an assembling procedure that is activated in response to the users finger designation

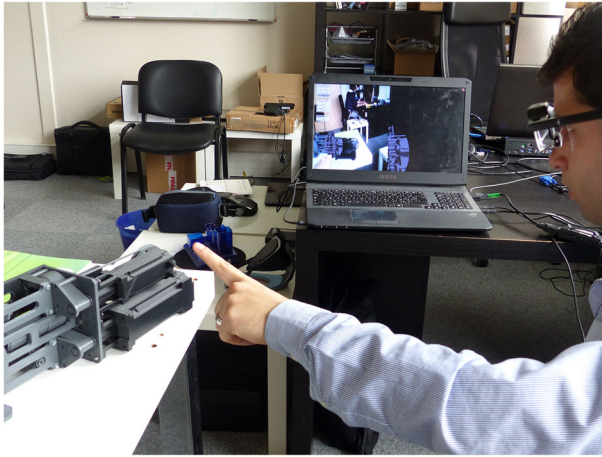


Fig. 22 The user equipped with OST glasses, requests dynamic augmented information by pointing his finger on a specific element of the workpiece. For better presentation of the experiment, the user's augmented view is re-transmitted into a LCD screen in the background

specially for selecting small components or when they move away from the object. However, when they are close to the object they prefer a direct interaction without the need of intermediate device.

9 Conclusion

This paper presents a constrained bundle adjustment framework for keyframe-based SLAM algorithms that improves the localization accuracy in a partially known environment. Two compound cost functions that include both information provided by the geometric model and the multi-view relationships of the known and the unknown parts of the environment have been proposed. It results in an unification of model-based and SLAM localization solutions benefiting from their respective advantages.

Experimental results on both synthetic and real data demonstrate that the proposed approach outperforms existing localization algorithms such as model-based trackers in terms of stability and classical keyframe-based SLAM in terms of accuracy and robustness to initialization. We also demonstrate that the unknown part of the environment should be used since it stabilizes the localization and allows to maintain the tracking when the object of interest is not or partially visible. We successfully apply our framework on real-time camera localization in a partially known environment for textured and textureless 3D objects even with an inaccurate 3D model.

We also described an interactive AR application prototype for industrial education and training applications. The system provides dynamic registered overlay instructions on an OST HMD in response to the user interactions. It allows to assure a precise and useful interaction for industrial training applications.

In our future work we will continue investigating of the tracking method. The proposed constraints can not handle curves and slightly textured objects like a plane. For this kind of object a solution may be a bundle adjustment constrained by occlusion edges. Some model-based approaches [37, 58] address this problem, but they lack accuracy and stability

since they use only the known part of the environment to localize the camera. Moreover, the proposed solution is based on the assumption that the object of interest is rigid and the scene is static but this is limiting for many applications. It would be interesting to generalize the approach of constrained SLAM for more complex contexts such as moving or articulated objects.

We also continue investigating existing and new methods of interaction such as visual gesture interfaces to virtual environments using only the embedded camera. Our work provides a first step for user interaction, in the restricted case of the pointing. Further research is needed to propose more interaction metaphors. The recent researches [5, 15] could guide this future work. Like mentioned by Hurst [15], one of the most important issues is how to switch between different types of operations. This requires to develop new concepts to allow the system to automatically distinguish between different gestures for various tasks.

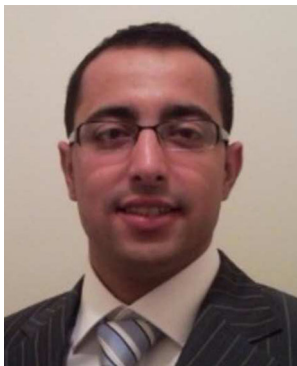
Acknowledgments We thank Laster Technologies company who provided the glasses prototype.

References

1. Bartoli A, Sturm P (2003) Constrained structure and motion from multiple uncalibrated views of a piecewise planar scene. *Int J Comput Vis* 52(1):45–64
2. Benhimane S, Malis E (2006) Integration of euclidean constraints in template based visual tracking of piecewise-planar scenes. In: *International Conference on Intelligent Robots and Systems, IROS*
3. Bleser G, Wuest H, Stricker D (2006) Online camera pose estimation in partially known and dynamic scenes. In: *International Symposium on Mixed and Augmented Reality, ISMAR*
4. Caron G, Dame A, Marchand E (2014) Direct model-based visual tracking and pose estimation using mutual information. *Image Vis Comput* 32(1):54–63
5. Chun WH, Höllerer T (2013) Real-time hand interaction for augmented reality on mobile phones. In: *Proceedings of the 2013 International Conference on Intelligent User Interfaces, IUI '13, NY, USA*, pp 307–314
6. Comport AI, Marchand E, Chaumette F (2003) A real-time tracker for markerless augmented reality. In: *International Symposium on Mixed and Augmented Reality, ISMAR*
7. Davison AJ, Reid ID, Molton ND, Monoslam OS (2007) Real-time single camera slam. *Trans Pattern Anal Mach Intell* 29(6):1052–1067
8. Drummond T, Cipolla R (2002) Real-time visual tracking of complex structures. *Trans Pattern Anal Mach Intell* 24(7):932–946
9. Engels C, Stewènius H, Nistèr D (2006) Bundle adjustment rules. In: *Photogrammetric Computer Vision (PCV), ISPRS*
10. Farenzena M, Bartoli A, Mezouar Y (2008) Efficient camera smoothing in sequential structure-from-motion using approximate cross-validation. In: *European Conference on Computer Vision, ECCV*
11. Gay-Bellile V, Lothe P, Bourgeois S, Royer E, Naudet-Colette S (2010) Augmented reality in large environments: Application to aided navigation in urban context. In: *International Symposium on Mixed and Augmented Reality, ISMAR*
12. Gay-Bellile V, Tamaazousti M, Dupont R, Naudet-Collette S (2010) A vision-based hybrid system for real-time accurate localization in an indoor environment. In: *International Conference on Computer Vision Theory and Applications, VISAPP*
13. Harris C (1992) *Tracking with rigid objects*. In: MIT Press
14. Harrison C, Benko H, Omnitouch ADW (2011) Wearable multitouch interaction everywhere. In: *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology, UIST '11*, pp 441–450
15. Wolfgang H, Wezel CV (2013) Gesture-based interaction via finger tracking for mobile augmented reality. *Multimed Tools Appl* 62(1):233–258
16. Kempter T, Wendel A, Bischof H (2012) Online model-based multi-scale pose estimation. In: *Computer Vision Winter Workshop, CVWW*
17. Klein G, Murray D (2007) Parallel tracking and mapping for small AR workspaces. In: *International Symposium on Mixed and Augmented Reality, ISMAR*

18. Klein G, Murray D (2008) Improving the agility of keyframe-based slam. In: European Conference on Computer Vision, ECCV
19. Kurz D (2014) Thermal touch: Thermography-enabled everywhere touch interfaces for mobile augmented reality applications. In: Proceedings IEEE International Symposium on Mixed and Augmented Reality (ISMAR2014), pp 9–16
20. Kurz D, Hantsch F, Grobe M, Schiewe A, Bimber O (2007) Laser pointer tracking in projector-augmented architectural environments. ISMAR, pp 1–8
21. Kyrki V, Kragic D (2011) Tracking rigid objects using integration of model-based and model-free cues. *Mach Vis Appl* 22(2):323–335
22. Ladikos A, Benhimane S, Navab N (2007) A real-time tracking system combining template-based and feature-based approaches. In: International Conference on Computer Vision Theory and Applications, VISAPP
23. Lepetit V, Fua P (2005) Monocular model-based 3d tracking of rigid objects: A survey. *Found Trends Comput Graph Vis* 1(1):1–89
24. Lepetit V, Fua P (2006) Keypoint recognition using randomized trees. *Trans Pattern Anal Mach Intell* 28(9):1465–1479
25. Lothe P, Bourgeois S, Dekeyser F, Royer E, Dhome M (2009) Towards geographical referencing of monocular slam reconstruction using 3d city models: Application to real-time accurate vision-based localization. In: Computer Vision and Pattern Recognition, CVPR
26. Lothe P, Bourgeois S, Royer E, Dhome M, Naudet-Collette S (2010) Real-time vehicle global localization with a single camera in dense urban areas: Exploitation of coarse 3d city models. In: Computer Vision and Pattern Recognition, CVPR
27. Lourakis MIA, Argyros AA (2005) Is levenberg-marquardt the most efficient optimization algorithm for implementing bundle adjustment? In: International Conference on Computer Vision, ICCV
28. Lowe DG (1987) Three-dimensional object recognition from single two-dimensional images. *J Artif Intell* 31(3):355–395
29. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. *Int J Comput Vis* 60(2):91–110
30. Marchand E, Boutheymy P, Chaumette F, Moreau V (1999) Robust real-time visual tracking using a 2d-3d model-based approach. In: International Conference on Computer Vision, ICCV
31. Marchand E, Chaumette F (2002) Virtual visual servoing: a framework for real-time augmented reality. *Comput Graph Forum* 21(3):289–298
32. Marquardt D (1963) An algorithm for least-squares estimation of non linear parameters. *J Soc Ind Appl Math* 11(1):431–444
33. Mouragnon E, Lhuillier M, Dhome M, Dekeyser F, Sayd P (2006) Real time localization and 3d reconstruction. In: Conference on Computer Vision and Pattern Recognition, CVPR
34. Newcombe RA, Davison AJ, Izadi S, Kohli P, Hilliges O, Shotton J, Molyneaux D, Hodges S, Kim D, Kinectfusion AF (2011) Real-time dense surface mapping and tracking. In: IEEE International Symposium on Mixed and Augmented Reality, ISMAR
35. Newcombe RA, Lovegrove S, Dtam AJD (2011) Dense tracking and mapping in real-time. In: International Conference on Computer Vision, ICCV
36. Nistér D, Naroditsky O, Bergen J (2004) Visual odometry. In: Computer Vision and Pattern Recognition, CVPR
37. Petit A, Marchand E, Kanani K (2012) Tracking complex targets for space rendezvous and debris removal applications. In: International Conference on Intelligent Robots and Systems, IROS
38. Platonov J, Heibel H, Meier P, Grollmann B (2006) A mobile markerless ar system for maintenance and repair. In: International Symposium on Mixed and Augmented Reality, ISMAR
39. Rothganger F, Lazebnik S, Schmid C, Ponce J (2006) 3d object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints. *Int J Comput Vis* 66(3):231–259
40. Royer E, Lhuillier M, Dhome M, Chateau T (2005) Localization in urban environments: Monocular vision compared to a differential gps sensor. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR
41. Simon G (2011) Tracking-by-synthesis using point features and pyramidal blurring. In: International Symposium on Mixed and Augmented Reality, ISMAR
42. Simon G, Berger M-O (2002) Pose estimation for planar structures. *Comput Graph Appl* 22(6):46–53
43. Snavely N, Seitz SM, Szeliski R (2008) Modeling the world from internet photo collections. *Int J Comput Vis* 80(2):189–210
44. Sourimant G, Morin L, Bouatouch K (2007) Gps, gis and video fusion for urban modeling. In: Computer Graphics International Conference, CGI

45. Strasdat H, Montiel JMM, Davison AJ (2010) Real-time monocular slam: Why filter? In: International Conference on Robotics and Automation, ICRA
46. Peter FS (2000) Algorithms for plane-based pose estimation. In: Computer Vision and Pattern Recognition, CVPR, pp 1706–1711
47. Szeliski R (2006) Image alignment and stitching: a tutorial. *Found Trends Comput Graph Vis* 2(1):1–104
48. Tamaazousti M, Gay-Bellile V, Naudet-Collette S, Bourgeois S, Dhome M (2011) Nonlinear refinement of structure from motion reconstruction by taking advantage of a partial knowledge of the environment. In: Computer Vision and Pattern Recognition, CVPR
49. Tamaazousti M, Gay-Bellile V, Naudet-Collette S, Bourgeois S, Dhome M (2011) Real-time accurate localization in a partially known environment: Application to augmented reality on 3d objects. In: International workshop on AR/MR registration, tracking and benchmarking, ISMAR Workshop, TrakMark
50. Triggs B, McLauchlan PF, Hartley RI, Fitzgibbon AW (2000) Bundle adjustment - a modern synthesis. In: International Workshop on Vision Algorithms Theory and Practice, ICCVW
51. Vacchetti L, Lepetit V (2004) Stable real-time 3d tracking using online and offline information. *Trans Pattern Anal Mach Intell* 26(20):1385–1392
52. Vacchetti L, Lepetit V, Fua P (2004) Combining edge and texture information for real-time accurate 3d camera tracking. In: International Symposium on Mixed and Augmented Reality, ISMAR
53. Wagner D, Reitmayr G, Mulloni A, Drummond T, Schmalstieg D (2010) Real-time detection and tracking for augmented reality on mobile phones. *Trans Vis Comput Graph* 16(3):355–368
54. Whelan T, Johannsson H, Kaess M, Leonard JJ, McDonald JB (2013) Robust real-time visual odometry for dense RGB-D mapping. In: International Conference on Robotics and Automation, ICRA
55. Whelan T, Kaess M, Fallon MF, Johannsson H, Leonard JJ, Kintinuous JBM (2012) Spatially extended KinectFusion. In: RSS Workshop on RGB-D: Advanced Reasoning with Depth Cameras
56. Wuest H, Stricker D, Herder J (2007) Tracking of industrial objects by using cad models. *J Virtual Real Broadcast* 4(1)
57. Wuest H, Vial F, Stricker D (2005) Adaptive line tracking with multiple hypotheses for augmented reality. In: International Symposium on Mixed and Augmented Reality, ISMAR
58. Wuest H, Wientapper F, Stricker D (2007) Adaptable model-based tracking using analysis-by-synthesis techniques. In: Computer Analysis of Images and Patterns, CAIP



Mohamed Tamaazousti received his master's Degree in applied mathematical from the university of Orleans, France in 2009. He received the Ph.D degree in computer vision from the University Blaise Pascal, Clermont-Ferrand, France in 2013. He did his research in partnership with the LASMEA and LVIC laboratory under the supervision of Professor Michel Dhome. He is currently a permanent researcher at CEA LIST. His main researcher interest include structure from motion of rigid scene, real time vision-based localization and reconstruction for autonomous system and augmented reality applications.



Sylvie Naudet Collette is in charge of the strategy and program development supporting the creation and growth of a new activity in computer vision, video and multimedia analysis, in the CEA LIST, the academic French laboratory of technology research on smart systems. Before 2013, she was in charge of the 3D Vision team, composed of 15 Researchers. Its work focuses on image processing, computer vision, sensor fusion, with applications in 3D reconstruction, metrology, augmented reality and geolocation systems. She has managed several national and european projects in the field of 3D reconstruction, augmented reality, geolocation systems and metrology by vision She has also managed projects of technology transfer to the industry and contributed to the creation of 3 start-up. She has obtained a Ph.D in computer Vision of the University of Orsay in 1997. She wrote, as author and co-author, 30 papers on computer vision and augmented reality and holds 3 patents too.



Vincent Gay-Bellile received the PHD degree in computer science from the University Blaise Pascal, Clermont-Ferrand, France. He did his research in partnership with the LASMEA and LSVE laboratory under the supervision of Professors Jean-Thierry Lapresté, Adrien Bartoli and Doctors Patrick Sayd. He is currently a permanent researcher at CEA LIST. His main research interests are in structure from motion for rigid and non rigid scenarios. He received the Best Student Paper Award at CORESA '07, together with Adrien Bartoli and Patrick Sayd.



Steve Bourgeois received the PhD degree in computer vision from the University Blaise Pascal (France) in 2006 under the supervision of Professor Michel Dhome. Since 2006, he is a permanent researcher at CEA LIST where he works on industrial project and supervises different PhD students. Its main research interest are Structure from Motion, 3D localization and Augmented Reality.



Bassem Besbes graduated with the Master of Vision and Pattern Recognition (2008) and the Ph. D. degree in Computer science (2011) at the National Institute of Applied Science of Rouen. He was a postdoctoral researcher at LIST (CEA - French Atomic Energy Agency) in the Vision & Content Engineering Laboratory before joining Diotasoft company. This company develops upscale augmented reality for professionals in the fields of Sales Support, Industrial Process Support, and Education. His research interest includes computer vision, pattern recognition, and augmented reality.



Michel Dhome 57 year old, is a senior Researcher of the French National Council for Scientific Research (CNRS). From 2005 to 2011, he was the head of the Laboratory of the Sciences and Materials for Electronics, and for Automatic (LASMEA). Since January 2012, he is the head of the new laboratory <<Pascal Institut>> of Clermont-Ferrand (union of three laboratories - 300 people). He is also the head of the laboratory of excellence IMobS3 (research program on “Innovative Mobility : Smart and Sustainable Solutions” between seven laboratories) who includes all the academic people of Clermont-Ferrand working in the mechanics, automatics, electronics, computer sciences and process engineering. His main scientific contributions concern essentially the domain of the << Artificial Perception >>, with the << Robotics >> as main field of application. His activity can be classified according to the following four topics:

- Auto-calibration of video cameras;
- Complex 3D objects alignment on video sequences;
- Real time tracking of textured objects;
- Artificial vision approaches for the autonomous vehicles guidance.

Today, its production is 178 scientific publications including 1 book, 8 book chapters, 20 publications in international journals, 100 papers in international conferences and 6 patents.