Real-Time Accurate Localization in a Partially Known Environment: Application to Augmented Reality on textureless 3D Objects.

Mohamed Tamaazousti* CEA, LIST
Vincent Gay-Bellile CEA, LIST
Sylvie Naudet Collette CEA, LIST
Steve Bourgeois CEA, LIST
Michel Dhome† LASMEA / CNRS

ABSTRACT

This paper addresses the challenging issue of real-time camera localization in a partially known environment, *i.e.* for which the geometric 3D model of one static object in the scene is available. We 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). In this paper a novel constrained bundle adjustment that deals with textureless 3D objects is introduced: model-based constraints provided by the sharp edges of the 3D model are used. We demonstrate the advantages of the resulting bundle adjustment with line constraints on both synthetic and real data and present very convincing augmentation of 3D objects in real-time.

Index Terms: H.5.1 [Multimedia Information Systems]: Artificial, augmented, and virtual realities—Life Cycle; I.4.8 [Scene Analysis]: Object Recognition—Tracking

1 INTRODUCTION

Real-time augmentation of a 3D object with a moving camera is a challenging research topic. It requires to accurately compute the relative pose of the camera *w.r.t.* this 3D object. Model-based tracking solutions exploit the *a priori* knowledge of the object geometry and/or appearance to estimate this pose in real-time by matching 3D features extracted from the model with their corresponding observations in the current image [7]. Nevertheless, this process implies that the object of interest is always visible and represents a large part of the images during the whole sequence for stable tracking.

On the other hand, keyframe-based SLAM algorithms, *e.g.* [5, 9, 10], estimate the motion of a camera without any prior on the scene geometry. They exploit the multi-view relationships to compute the camera poses and reconstruct a 3D map (*i.e.* usually a sparse 3D point cloud) of the environment that are simultaneously refined with a Bundle Adjustment (BA). While off-line solutions [10] use a global bundle adjustment that refines the whole reconstruction of a video sequence in a single optimization process, on-line solutions [5, 9] use a local bundle adjustment that optimizes sequentially only a limited number of camera poses and the 3D points they observe. Keyframe-based SLAM localization is very stable since the whole information present in the 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 initial scale factor of the scene is arbitrary set
- the localization suffers from error accumulation due to image noise, matching errors, etc.

Recent works tried to solve some of these drawbacks by combining SLAM with model-based tracking. Bleser et al. [1] introduce a solution to set the coordinate frame and the scale factor by using a model-based tracking algorithm at the first frame. Once the initial pose is estimated in the object coordinate frame, the scale factor is transmitted to the SLAM process by back-projecting point features extracted from the first image onto the model surface. It results in a 3D points cloud used by the SLAM process as an initial reconstruction of the environment. Nevertheless, this solution has different drawbacks. First, the coordinate frame and the scale factor are set from a single viewpoint registration which is subject to inaccuracies especially along the optical axis. The accuracy of the SLAM localization *w.r.t.* the 3D object is thus limited by this initialization. Second, the error accumulation problem is not handled. In our previous work [11], we proved that it is also possible to take benefit from the geometric model to improve the localization accuracy of keyframe-based SLAM. We introduced the constrained bundle adjustment framework that simultaneously includes the geometric constraints provided by the 3D model, the multi-view constraints relative to the known part of the environment (i.e. the object observations in the video sequence) and the multi-view constraints relative to the unknown part of the environment (i.e. the others observations). More specifically, the introduced constraint imposes that the 3D points of the object of interest, reconstructed by the SLAM process, lie on the 3D model through planar constraints in the bundle adjustment. This approach solves the drawbacks of keyframe-based SLAM solutions that prevented them to be used for 3D objects tracking. Nevertheless, the constraint proposed in [11] assumes that the object of interest is highly textured. Indeed, the planar constraints require that a 3D points cloud of the object of interest have been previously reconstructed by the SLAM process.

In this paper we introduced a novel constrained bundle adjustment that deals with textureless 3D objects (see Figure 8). We rather use model-based constraints provided by the sharp edges of the 3D model. Segments extracted from the 3D model are used in the bundle adjustment to constrain the camera trajectory by adding their reprojection errors in the bundle adjustment. The sparse structure of the resulting constrained local bundle adjustment is exploited in the implementation to ensure real-time performance.

Experimental results on synthetic data demonstrate that the proposed approach outperforms existing localization algorithms such as keyframe-based SLAM without model constraints in terms of accuracy. We also demonstrate on real data that the proposed approach outperforms model-based tracker [7] in terms of stability, since the unknown part of the environment is exploited. This also allows to maintain the tracking when the object of interest is not or partially visible. We present convincing augmentation on textureless 3D objects such as a toy car and a kitchen see Figures 8 and 9.

Plan. The Section 2, briefly describes keyframe-based SLAM algorithms and the associated local/global bundle adjustment pro-

*CEA LIST, Vision and Content Engineering Laboratory, Point Courrier 94, Gif-sur-Yvette, F-91191 France, e-mail: mohamed.tamaazousti@cea.fr
†Clermont Universite, Universite Blaise Pascal, LASMEA, BP 10448, Clermont-Ferrand / CNRS, UMR 6602, LASMEA, AUBIERE

cess. The Section 3 presents the proposed bundle adjustment with line constraints and demonstrate its sparse property in Section 4. The SLAM with the line constrained bundle adjustment is evaluated on synthetic and real data for textureless 3D objects tracking in Section 5. Finally, we give our conclusions and discuss future works, in Section 6.

Notation. Matrices are designated by sans-serif fonts such as M. Vectors are expressed in homogeneous coordinates, e.g. $\mathbf{q} \sim (x,y,w)^\mathsf{T}$ where $^\mathsf{T}$ is the transposition and $\sim$ the equality up to a non-zero scale factor. SLAM reconstruction is composed of N 3D points $\{\mathbf{Q}_i\}_{i=1}^N$ and m cameras $\{\mathsf{C}_k\}_{k=1}^m$. We note $\mathbf{q}_{i,k}$ the observation of the 3D point $\mathbf{Q}_i$ in the camera C_k and $\mathscr{A}_i$ the set of camera indexes observing $\mathbf{Q}_i$ with $n_i = \text{card}(\mathscr{A}_i)$. The projection matrix P_k associated with the camera C_k is given by $\mathsf{P}_k = \mathsf{K}\mathsf{R}_k^\mathsf{T}(\mathsf{I}_3|-\mathbf{t}_k)$, where K is the matrix of the intrinsic parameters and $(\mathsf{R}_k,\mathbf{t}_k)$ the extrinsic ones.

2 Keyframe-Based SLAM

Keyframe-based SLAM algorithms, *e.g.* [5, 9, 10], have proven to be relevant for camera localization in an unknown environment. They are based on the following steps to reconstruct the camera trajectory and a 3D points cloud of the observing scene. Keyframe-based SLAM begins by determining an initial reconstruction with two or three frames that will be used to arbitrary fix the global coordinate frame and the scale of the reconstruction. After that, a robust pose estimation is carried out for each frame of the video using features detection and matching. Some of the frames are selected and become keyframes that are used to 3D points triangulation. They are chosen such that there is enough motion between them but not to much to ensure matching. Then the camera poses and the 3D points cloud are simultaneously refined with a bundle adjustment. It 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:

$$\mathscr{E}_E\left(\{\mathsf{R}_j,\mathbf{t}_j\}_{j=1}^m,\{\mathbf{Q}_i\}_{i=1}^N\right) = \sum_{i=1}^N \sum_{j\in\mathscr{A}_i} d^2(\mathbf{q}_{i,j},\mathsf{P}_j\mathbf{Q}_i), \qquad (1)$$

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

For long sequences, BA rapidly becomes very time consuming and not adapted for real time localization even with an efficient sparse implementation (details are given below). To tackle this limitation, the local bundle adjustment framework has been proposed by [9]. The idea is to reduce the number of estimated parameters by optimizing only a subset of the 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 performance) and the reconstructed points they observe.

Normal equations and sparsity. Eq. 1 is usually minimized by the Levenberg-Marquardt (LM) algorithm [8] that iterates a linearized solution around the current parameters. The linear solution is found by second-order Taylor expansion and the second-order derivative matrix, *i.e.* the Hessian, is approximated by Jacobian products. The system to be solved for each Levenberg-Marquardt iteration is:

$$\begin{pmatrix} \mathsf{U} & \mathsf{W} \\ \mathsf{W}^\mathsf{T} & \mathsf{V} \end{pmatrix}\begin{pmatrix} \Delta_c \\ \Delta_p \end{pmatrix} = \begin{pmatrix} \mathsf{J}_c^\mathsf{T}\varepsilon \\ \mathsf{J}_p^\mathsf{T}\varepsilon \end{pmatrix} \qquad (2)$$

where $\mathsf{U} = {\mathsf{J}_c}^\mathsf{T}\mathsf{J}_c$, $\mathsf{V} = {\mathsf{J}_p}^\mathsf{T}\mathsf{J}_p$, $\mathsf{W} = {\mathsf{J}_c}^\mathsf{T}\mathsf{J}_p$ are sub-matrices composing the approximation of the Hessian matrix, $\mathsf{J} = \left[\mathsf{J}_c,\mathsf{J}_p\right]$ is the Jacobian matrix, Δ_c, Δ_p are the parameter increments to be calculated and ε is a vector concatenating all the residual errors[1].

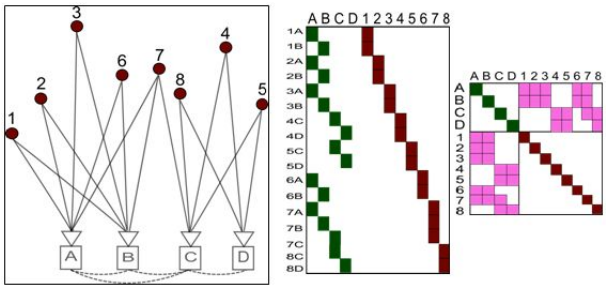

Figure 1: Left : a toy bundle problem. In dark red, the reconstructed 3D points cloud $(1-8)$. The square (A-D) represent the camera trajectory. Black lines indicate the observability of the 3D points for each position of the moving camera. Right : the associated Jacobian and Hessian matrices.

A useful property of BA is the sparse block structure of the normal equations due to the lack of interaction among parameters for different 3D points and cameras: the matrices U and V are block-diagonal and the sizes of the associated blocks are 6×6 and 3×3 respectively. The matrix W is composed of 6×3 sub-matrices that express the inter correlations between the structure and the motion parameters. By taking advantage of this specific sparse structure, BA can be efficiently implemented as described in [12].

An illustration of the block structures of the Jacobian and Hessian matrices for a toy bundle problem is represented in Figure 1.

3 Bundle Adjustment with Line Constraints

3.1 Problem Formulation

The major limitations of keyframe-based SLAM algorithms is that the reconstruction is estimated up to a scale factor and in an arbitrary coordinate frame. They also may suffer from error accumulations due to image noises, matching errors, etc. To solve these problems, we introduce in [11] an original solution for camera localization in a partially known environment. The geometric constraints, provided by the 3D model of one object in the scene, are incorporated in the bundle adjustment.

The resulting constrained bundle adjustment is formulated as a compound cost functions. It is composed of a known part (*i.e.* with model-based constraints) and an unknown part (*i.e.* without model-based constraints) of the environment:

$$\mathscr{E} = \rho\left(\mathscr{E}_E,c_1\right) + \lambda\rho\left(\mathscr{E}_M,c_2\right), \qquad (3)$$

where $\mathscr{E}_E$,$\mathscr{E}_M$ are the cost functions associated to the known and the unknown parts of the environment respectively. Adding the model based constraints in the bundle adjustment introduced an additional step in the minimization since data extracted from the images have to be associated to the model. A robust estimator ρ is thus required to deals with wrong data-to-model associations. The weight λ that control the influence of each term is usually compute through cross validation [3] or fixed experimentally. We use the simplest alternative described in [11]: the influence of each term is directly controlled through a particular choice of the rejecting thresholds c_1 and c_2 of the M-estimator.

3.2 The line constraint

The SLAM with the constrained bundle adjustment provides many benefits. Indeed, it resolves the drawbacks of the "classic" SLAM,

[1] where index $_c$ and $_p$ indicate the dependence on the camera and the points parameters respectively.

mentioned before and also manages to improve a rough initialization of the coordinate frame and scale factor of the scene. Nevertheless, the algorithm proposed in [11] assumes that the object of interest is highly textured. Indeed, the planar constraints require that a 3D points cloud of the object of interest have been previously reconstructed by the SLAM process.

The constraint presented here rather uses 3D segments extracted from the 3D model to directly constrain the camera poses. The constraint consists in aligning the sharp edges of the 3D model with the edges of the object extracted in the image. The sharp edges of the 3D model (a mesh model) are edges formed by two triangles and whose dihedral angle is superior to a certain threshold. These edges are then sampled in a set of short 3D segments $\{\mathbf{L}_i\}_{i=1}^{s}$; each one is parametrized by a center point $\mathbf{M}_i$ and a direction $\mathbf{D}_i$. These segments constitute the known part of the environment.

3.3 The Cost Function

The resulting line constraint minimizes the orthogonal distance between the projection of the segment's center point $\mathsf{P}_j\mathbf{M}_i$ and its associated edge feature $\mathbf{m}_{i,j}$ in the image, see *e.g.* [14] for details. Let's assume 2D/3D associations between the 3D segments $\mathbf{L}_i$ of the model and edges $\mathbf{m}_{i,j}$ extracted at keyframes $j \in \mathscr{S}_i$ (see section 3.4), the line constraints are given by:

$$\mathscr{E}_M\left(\{\mathsf{R}_j,\mathbf{t}_j\}_{j=1}^{m}\right) = \sum_{i=1}^{s}\sum_{j\in\mathscr{S}_i} \left|\mathbf{n}_{i,j}.\left(\mathbf{m}_{i,j} - \mathsf{P}_j\mathbf{M}_i\right)\right|, \qquad (4)$$

where $\mathbf{n}_{i,j}$ is the normal of the projected direction $\mathsf{P}_j\mathbf{D}_i$.

Comparatively to tracking methods based on edges of the model [2, 4, 14], this cost function uses roughly the same criterion but with an extension to the multiple view case: all the keyframes included in the bundle adjustment are used for the line constraint.

Note that segments have previously been used by Klein *et al.* in [6] 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, edges triangulation and a dedicated Bundle Adjustment to combine point and edge features. In our case, 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. The resulting compound cost function of our bundle adjustment with line constraints is then given by Eq. (5).

3.4 2D/3D associations between edges and 3D segments

The bundle adjustment with line constraints requires an additional step compared to the "classical" one. In fact, to achieve line constraints, 3D segments extracted from the model have to be associated to edges in the image. A visibility test is first used to get only a subset of visible 3D segments for each keyframes of the BA. The visible 3D segments are then projected in each keyframes 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 Bundle Adjustment) estimated by the SLAM process at each keyframe are good estimates. The steps of the bundle adjustment with line constraints are recapitulated in Table 1.

4 Sparse Bundle Adjustment with Line Constraints

This section demonstrates that the sparse property of the matrices (Hessian and Jacobian), associated to the bundle adjustment is preserved when the line constraint is added.

The Figure 2 (Left) represents a toy bundle problem with line constraints and the Jacobian and Hessian matrices associated to

- Visibility test to find a subset of visible 3D segments for each keyframes $j \in \mathscr{S}_i$.
- Projection of the visible 3D segments $\mathbf{L}_i$ in each keyframes $j \in \mathscr{S}_i$.
- Association of 3D segments $\mathbf{L}_i$ to image edges $\mathbf{m}_{i,j}$.
- Computation of the rejecting thresholds c_1 and c_2 [11].
- Minimization of Eq. (5) with the Levenberg Marquardt (LM) algorithm [8].

Table 1: bundle adjustment with line constraints.

Eq. 5. The Hessian matrix has exactly the same structure as for a classic bundle adjustment, see Figure 2 (Right). Indeed, the line constraint add residuals depending only on the camera parameters in the Jacobian matrix (one residual for each segment observation). Moreover, the line constraints described above do not increase the number of parameters since the 3D segments are not optimized.

Thus, an efficient sparse implementation of the bundle adjustments with line constraints is possible, as described in [12].

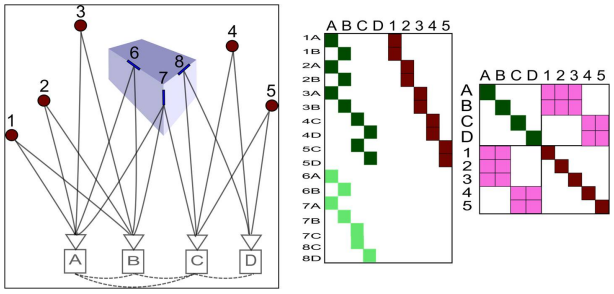

Figure 2: Left: a toy bundle problem with line constraints. In dark red, 3D points of the unknown part of the environment $(1-5)$; in dark blue, 3D segments that constitute the known part of the environment $(6-8)$. The square (A-D) represents the camera trajectory. Black lines indicate the observability of the 3D features (points and segments) for each position of the moving camera. Right : the associated Jacobian and Hessian matrices.

5 Experimental results

This section describes the evaluation of the bundle adjustment with line constraints on synthetic and real data. We use the keyframe-based SLAM described in [9]. It is an on-line localization algorithm that achieves real-time performance through local bundle adjustment applied on a sliding window of keyframes triplets. At each keyframe, only the poses associated to the three last keyframes and the 3D points they observed are optimized.

An evaluation of the SLAM algorithm performances, for 3D object tracking, is realized on a synthetic sequence. This is done with the two following local bundle adjustment algorithms:

- The original one described in [9] called LBA_E in the rest of the paper. It minimizes Eq. (1) by the Levenberg-Marquardt algorithm.
- The proposed LBA_LC&E algorithm[2]. It minimizes Eq.(5) that takes line constraints into account with the procedure described in Table 1.

[2]LC means Lines 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.

$$\mathscr{E}\left(\{\mathsf{R}_j,\mathbf{t}_j\}_{j=1}^{m},\{\mathbf{Q}_i\}_{i=1}^{N}\right)=\underbrace{\sum_{i=1}^{N}\sum_{j\in\mathscr{A}_i}\rho\left(d^2(\mathbf{q}_{i,j},\mathsf{P}_j\mathbf{Q}_i),c_1\right)}_{\text{Unknown part of the environment (E)}}+\underbrace{\sum_{i=1}^{s}\sum_{j\in\mathscr{S}_i}\rho\left(\left|\mathbf{n}_{i,j}.\left(\mathbf{m}_{i,j}-\mathsf{P}_j\mathbf{M}_i\right)\right|,c_2\right)}_{\text{Known part of the environment (LC)}} \tag{5}$$

Then a comparison is realized, on real data, between the SLAM algorithm with the line constrained bundle adjustment (SLAM + LBA_LC&E) and state of the art 3D object tracker.

Note that the initialization step of the SLAM process is modified. In its original implementation [9], it is realized by the five points algorithm, on the three first frames. We rather use here the initialization presented in [1]: an approximate pose, that roughly register the world and the 3D model coordinate frames, is estimated on the first frame of the video sequence, then an initial 3D points cloud is obtained through back-projection of the observations extracted on this frame.

5.1 Evaluation on Synthetic Data

In this section we compare the two local bundle adjustment algorithms described above, LBA_E and LBA_LC&E, on a synthetic sequence. We evaluate the contribution of the LBA_LC&E refinement algorithm in the SLAM, 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.

5.1.1 The "Cubes" Sequence

In this sequence, illustrated in Figure 3, 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.

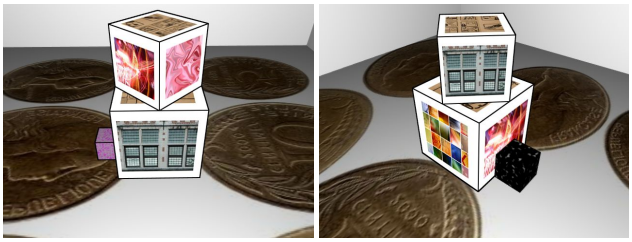

Figure 3: Two images of the "cubes" sequence.

5.1.2 Comparison of the two LBA Algorithms.

Two kinds of experiments are performed on the "cubes" sequence to compare the LBA_E and LBA_LC&E refinement algorithms.

The first one evaluates the accuracy of the resulting localization for a perfect initialization given by the ground truth. Results in terms of accuracy are shown on Figure 4. The SLAM with the refinement algorithm LBA_E is subject to error accumulation whereas, with the line constrained local bundle adjustment LBA_LC&E, the resulting localization is drift free. Adding the line constraints improve drastically the accuracy of the localization.

The second experiment, evaluates the robustness of the SLAM + LBA_LC&E to inaccurate initialization. Perturbations with increasing magnitudes are performed on the camera pose at first frame. Their amplitude fluctuates between 1% and 6% of the circle radius formed by the camera trajectory. The results are averaged over 10 random trials. The Figure 5 shows that the SLAM + LBA_LC&E deals with inaccurate initialization: after a certain amount of camera poses, the 3D errors stabilize to small values. We do not present results against initialization inaccuracy for the LBA_E refinement algorithm since it is obvious that it has no reason to reduce the starting error.

5.2 Evaluation on Real Data

It is shown above, that with the refinement algorithm LBA_LC&E, the SLAM approach can be used for texturless 3D object tracking in a partially known environment. In this section, the SLAM + LBA_LC&E is then compared to state of the art model-based 3D object tracker. For this, sequences have been acquired with a low-cost IEEE1394 GUPPY camera providing (640×480) images at 30 frames per second.

5.2.1 Tracking of a Textureless 3D Object

The object of interest is a textureless toy car representing a Lamborghini Gallardo. The 3D model used in our experiments is composed of 14000 triangles. We extract from this model 1186 3D segments as seen in Figure 6. The car is placed in a desk context composed by a computer screen, keyboard, books, etc. They constitute the unknown part of the environment. We compare sequential SLAM that includes the local bundle adjustment with line constraints LBA_LC&E with a classic model-based tracker which, by definition, uses only the known part of the environment. The latter is an improved version of the model-based tracker of [2] that includes multiple hypotheses for data associations as in [13, 15]. The comparison is done on a challenging sequence that presents large variation in scale, fast motion, partial occlusions of the toy car, *etc*. The initialization (*i.e.* coordinate frames registration) on the first frame is performed by positioning approximatively a coded marker near the car.

Figure 6: The 3D segments extracted from the Lamborghini model.

5.2.2 Results.

The 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 registration on the first frame is not accurate since the coded marker is roughly positioned near to the car. Both methods manage to correct this registration error after a few frames: the front and the back of the car are well projected on the images as seen in Figure 7 (Left).

On the other hand the sequential SLAM with the LBA_LC&E refinement algorithm outperforms the model-based tracker. In fact,

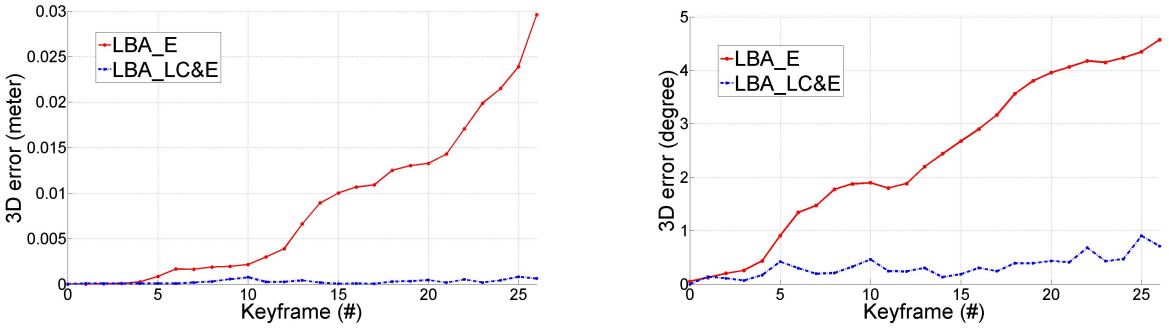

Figure 4: (a) (resp. (b)): Errors in position (resp. orientation) expressed in meter (resp. in degree) for the different LBA algorithms.

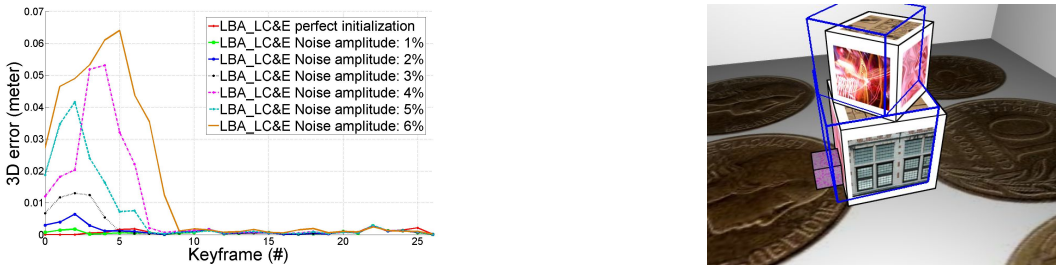

Figure 5: (a) Results obtained by the LBA_LC&E refinement algorithm for different magnitudes of perturbation ($\in [1\% \cdots 6\%]$ of the radius of the camera trajectory) applied on the first camera pose. (b) Illustration of the discrepancy error in the image between the reprojected model and the object of interest for a perturbation magnitude of 6%.

the latter failed when fast motion occurs due to bad 2D/3D associations of the features as seen in Figure 7 (Right). Moreover the reprojection of the model in the images is subject to jitters. Our proposed localization algorithms use both the informations provided by the known and the unknown parts of the environment. It yields an accurate, stable and robust localization. The SLAM with the LBA_LC&E refinement algorithm is then perfectly designed for Augmented Reality applications.

Figure 7: Localization in a partially known environment composed of a textureless 3D object. Top: results obtained with a model-based tracker similar to the one proposed in [2]. Bottom: results obtained with a sequential SLAM that use the proposed LBA_LC&E algorithms.

5.3 Application to Augmented Reality

We present two applications of Augmented Reality using the SLAM method with the proposed refinement algorithm LBA_LC&E. The first is the virtual tuning of a vehicle as shown in Figure 8. The personalization of the car is done by changing the color of one or more elements of its bodywork. It is thus possible separately to change, very realistically, the color of the hood, trunk and the whole body of the car. Note that, some elements like wheels, windows and windshields are not augmented.

We also apply the proposed localization algorithm to kitchen designing through Augmented Reality. The goal is to enable the user to see the different designs combinations of a kitchen *i.e.* colors, materials, shapes of the handles, etc while moving around it. Each part of the real kitchen, *e.g.* splashback, furnitures, worktops, can be modified separately or together. A menu appears when the user clicks on one kitchen component in the images. It contains the different designs available for the selected element. Once the design is chosen, the kitchen is automatically augmented with it in real-time. A realistic rendering has been obtained by modeling the light sources of the room including the external light. Note that it is also possible to remove a furniture and eventually replace it. The Figure 9 shows the different functionalities of the kitchen design software.

6 Conclusion

This paper presents a bundle adjustment with line constraints for keyframe-based SLAM algorithms. A compound cost function that includes both information provided by the sharp edges of the geometric model and the multi-view relationships of the unknown parts of the environment, has been proposed.

Experimental results on synthetic data demonstrate that the proposed approach outperforms ”classic” keyframe-based SLAM

Figure 8: Augmented Reality on a textureless toy car. The car with its original color (Left) and the same car with different augmented colors. Note that only its bodywork has been changed, the wheels, the windows and the windshield are real.

Figure 9: Kitchen Designing through Augmented Reality. The original kitchen (Left) is modified with an augmented worktops, with augmented furnitures and furniture replacement. A menu is used to select the design to apply at each kitchen component.

without model constraints. The localization of the SLAM with the LBA_LC&E refinement is more accurate and also robust to inaccurate initialization, in an object tracking context.

A comparison is then performed on real data between the proposed approach SLAM with LBA_LC&E and the state of the art model-based 3D object tracker. This comparison shows that the proposed approach provides better results than the model-based tracker in terms of stability. Indeed, the unknown part of the environment stabilizes the localization and allows to maintain the tracking when the object of interest is just partially or even not visible.

For future work we study the possibility of integrating segments of the 3D model in the different steps of the SLAM *e.g.* tracking at each frame.

REFERENCES

[1] Gabriele Bleser, Harald Wuest, and Didier Stricker. Online camera pose estimation in partially known and dynamic scenes. In *ISMAR*, 2006.

[2] Tom Drummond and Roberto Cipolla. Real-time visual tracking of complex structures. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(7):932–946, 2002.

[3] Michela Farenzena, Adrien Bartoli, and Youcef Mezouar. Efficient camera smoothing in sequential structure-from-motion using approximate cross-validationt. In *ECCV*, 2008.

[4] C. Harris. Tracking with rigid objects. In *MIT Press*, 1992.

[5] Georg Klein and David Murray. Parallel tracking and mapping for small AR workspaces. In *ISMAR*, 2007.

[6] Georg Klein and David Murray. Improving the agility of keyframe-based slam. In *ECCV*, 2008.

[7] Vincent Lepetit and Pascal Fua. Monocular model-based 3d tracking of rigid objects: A survey. In *FTCGV*, 2005.

[8] D. Marquardt. An algorithm for least-squares estimation of non linear parameters. *J. Soc. Industr. Appl. Math.*, 11(1):431–444, 1963.

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

[10] Eric Royer, Maxime Lhuillier, Michel Dhome, and Thierry Chateau. Localization in urban environments: Monocular vision compared to a differential gps sensor. In *CVPR*, 2005.

[11] Mohamed Tamaazousti, Vincent Gay-Bellile, Sylvie Naudet-Collette, Steve Bourgeois, and Michel Dhome. Nonlinear refinement of structure from motion reconstruction by taking advantage of a partial knowledge of the environment. In *CVPR*, 2011.

[12] Bill Triggs, Philip F. McLauchlan, Richard I. Hartley, and Andrew W. Fitzgibbon. Bundle adjustment - a modern synthesis. In *ICCVW: International Workshop on Vision Algorithms Theory and Practice*, 2000.

[13] Luca Vacchetti, Vincent Lepetit, and Pascal Fua. Combining edge and texture information for real-time accurate 3d camera tracking. In *ISMAR*, 2004.

[14] Harald Wuest and Didier Stricker. Tracking of industrial objects by using cad models. *Journal of Virtual Reality and Broadcasting*, 4(1), 2007.

[15] Harald Wuest, Florent Vial, and Didier Stricker. Adaptive line tracking with multiple hypotheses for augmented reality. In *ISMAR*, 2005.