Augmenting Markerless Complex 3D Objects By Combining Geometrical and Color Edge Information

Antoine Petit, Eric Marchand, Keyvan Kanani

▶ To cite this version:

HAL Id: hal-00851424
https://hal.inria.fr/hal-00851424

Augmenting Markerless Complex 3D Objects By Combining Geometrical and Color Edge Information

Antoine Petit*
INRIA Rennes

Eric Marchand†
IRISA

Keyvan Kanani‡
Astrium

ABSTRACT

This paper presents a method to address the issue of augmenting a markerless 3D object with a complex shape. It relies on a model-based tracker which takes advantage of GPU acceleration and 3D rendering in order to handle the complete 3D model, whose sharp edges are efficiently extracted. In the pose estimation step, we propose to robustly combine geometrical and color edge-based features in the nonlinear minimization process, and to integrate multiple-hypotheses in the geometrical edge-based registration phase. Our tracking method shows promising results for augmented reality applications, with a Kinect-based reconstructed 3D model.

Index Terms: 3D visual tracking, model-based tracking

1 INTRODUCTION

In Augmented Reality, dealing with a 3D object is a challenging task which requires to compute the full pose of the camera with respect to the object to be augmented. Based on the knowledge of the 3D model of the target, common approaches address this problem by using either texture [1] or edge features [2, 3, 5]. Edge features offer a good invariance to illumination changes or image noise, the pose computation being generally achieved by minimizing the distance between the projected edges of the 3D model and the corresponding edge features in the image. But though they have proven their efficiency, such methods require an image extraction process which can involve outliers and suffer from having similar appearances. It can result in ambiguities between different edges, leading, particularly in the case of complex objects, to tracking failures, which are harmful for augmented reality applications. In order to improve the tracking robustness and accuracy, we propose, following [4], to integrate classical geometrical and color edge-based features in the pose estimation phase. The general idea is to combine in the criterion to be optimized a geometrical information provided by the distances between model and image edges with a denser color-based information through object/background color separation statistics across the model edges. A low-level multiple hypotheses edge matching process is also embedded in our framework. In addition, like in our previous work [5], the model projection and model edge generation phase relies on the graphics process units (GPU) in order to handle complex 3D models, of any shape and to be reasonably time-consuming. This technique also allows to directly use 3D models which have been automatically reconstructed using sensors such as Kinect.

2 COMBINING GEOMETRICAL AND COLOR FEATURES IN 3D MODEL-BASED TRACKING

Our problem is restricted to model-based tracking, using a 3D model of the target. The goal is to estimate the camera pose ($\mathbf{r}$)

---

*e-mail: antoine.guillaume.petit@inria.fr
†e-mail:eric.marchand@irisa.fr
‡e-mail:keyvan.kanani@eads.astrium.net

by minimizing, with respect to $\mathbf{r}$, the error Δ between the observed data $\mathbf{s}^*$ and the current value $\mathbf{s}(\mathbf{r})$ of the same features projected in the image according to the current pose:

$$\Delta(\mathbf{r}) = \sum_i \rho(s_i(\mathbf{r}) - s_i^*) \qquad (1)$$

where ρ is a robust estimator, which reduces the sensitivity to outliers. This is a non-linear minimization process with respect to the pose parameters $\mathbf{r}$. We follow the Virtual Visual Servoing framework [2], similar to the Gauss-Newton approach. Our challenge is to combine geometrical edge-based features with a complementary type of features in order to overcome the limitations of classical edge-based approaches. Our idea is to avoid any image extraction or segmentation that would lead to outliers and mismatches. In this sense, we follow [4], by optimizing the separation of color statistics collected on both sides of the projected edges of the 3D model. Δ can then be rewritten as $\Delta = w^g \Delta^g + w^c \Delta^c$. Δ^g refers to the geometrical edge-based objective function and Δ^c stands for the color-based one. w^g and w^c are weighting parameters. Both kinds of features are based on the projection of the 3D model, in the vicinity of the projected model edges.

3 GEOMETRICAL EDGE-BASED FEATURES

3.1 Model projection and generation of model edge points

As in our previous work [5], we propose to automatically manage the projection of the model and to determine the visible and prominent edges from the rendered scene, by considering the direct use of a complete CAD model. By relying on the use of the graphics process units (GPU) and a 3D rendering engine, we avoid any manual pre-processing. For each acquired image I_{k+1}, the model is rendered with respect to the previous pose $\mathbf{r}_k$. A set of 3D points $\mathbf{X}_i$ which belong to the target sharp or texture edges can then be obtained. This is done by processing the rendered depth buffer through a Laplacian filter, allowing to determine the discontinuities which suit the geometrical appearance of the scene, resulting in a binary edge map of the visible scene.

3.2 Feature computation

In order to compute the edge-based function Δ^g, we rely on our previous work [5], as briefly described hereafter. From the model edge points we perform a 1D search along the normal of the underlying edge of each $\mathbf{x}_i(\mathbf{r}_k)$, projected from the 3D point $\mathbf{X}_i$ with respect to the pose $\mathbf{r}_k$. A common approach is to choose the pixel with the maximum gradient as the matching edge point $\mathbf{x}_i'$ in the image. Once correspondences are established, we consider the distance between the projected 3D line $l_i(\mathbf{r})$ underlying the projected model edge point $\mathbf{x}_i(\mathbf{r})$ and the selected matching point $\mathbf{x}_i'$ in the image. Δ^g can be written as:

$$\Delta^g = \sum_i \rho^g(d_\perp(l_i(\mathbf{r}), \mathbf{x}_i'))) \qquad (2)$$

where ρ^g is a Tukey robust estimator.

In order to improve the robustness of the pose estimation and to avoid problems due to ambiguities between edges, it is possible to consider and register for each $\mathbf{x}_i$ different hypotheses corresponding to potential edges. They correspond to different local extrema of

the gradient $\mathbf{x}'_{i,j}$ along the scan line. As in [5, 6], we choose the hypothesis which has the closest distance to the projected 3D line l_i during the minimization process and Δ^g becomes:

$$\Delta^g = \sum_i \rho^g (\min_j d_\perp (l_i(\mathbf{r}), \mathbf{x}'_{i,j})) \qquad (3)$$

4 COLOR FEATURES

The color-based function Δ^c is designed to characterize the separation between both sides of the projected model edges, by relying on color information. In order to compute Δ^c, as in [4], we restrict ourself to silhouette edges. Given the set of projected silhouette model edge points $\mathbf{x}_i(\mathbf{r})$, determined from points $\mathbf{X}_i$, we compute color statistics, so to say RGB means $\bar{\mathbf{I}}_i^O$ and $\bar{\mathbf{I}}_i^B$ and covariances $\bar{\mathbf{R}}_i^O$ and $\bar{\mathbf{R}}_i^B$, on both side of the edge (object O and background B) using $2 \times D + 1$ pixels along the edge normal $\mathbf{n}_i$, up to a distance L.

These statistics are then mixed according to a fuzzy membership rule, giving a mean $\hat{\mathbf{I}}_i(\mathbf{r})$ and a covariance $\hat{\mathbf{R}}_i(\mathbf{r})$. $\hat{\mathbf{I}}_i(\mathbf{r})$ represents a desired color value for the jth pixel $\mathbf{y}_{i,j}$ on the normal $\mathbf{n}_i$, wether it is on the object O or background side B. With the error $\mathbf{e}_{i,j}^c(\mathbf{r}) = \hat{\mathbf{I}}_i(\mathbf{r}) - \mathbf{I}(\mathbf{y}_{i,j})$ and using a M-estimator (Tukey) to cope with outliers, Δ^c can be written as:

$$\Delta^c = \sum_i \sum_j \rho^c (\sqrt{\mathbf{e}_{i,j}^c(\mathbf{r})^T \hat{\mathbf{R}}_i^{-1} \mathbf{e}_{i,j}^c(\mathbf{r})}) \qquad (4)$$

For more accuracy and temporal smoothness, we propose to introduce temporal consistency, by integrating the color statistics computed on the previous frame $^P\mathbf{I}$ for the silhouette edge points $\mathbf{x}_i(\mathbf{r}_k)$ at the first iteration of the minimization process. With $\alpha + \beta = 1$, it gives $\mathbf{e}_{i,j}^c(\mathbf{r}) = \alpha \hat{\mathbf{I}}_i(\mathbf{r}) + \beta(^P\hat{\mathbf{I}}_i(\mathbf{r})) - \mathbf{I}(\mathbf{y}_{i,j})$.
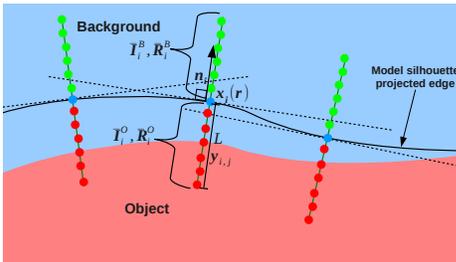

Figure 1: Collection of local color statistics on background (B) and object (O) sides.

5 RESULTS

Our tracking method has been tested for augmented reality applications on a figurine of "Mario". The target, which is 35cm high, is made of curved and complex shapes. We have reconstructed it automatically using a Kinect sensor and the ReconstructMe software, which is an easy to use real-time 3D reconstruction system. Despite rough modelization of some parts (the cap, the hands for instance), this model, which is made of 15000 vertices, for a 5.5MB size, has been directly used and processed in our tracking algorithm, showing the convenience of the proposed method. We have augmented this object with a virtual "Yoshi". The rendering process of the 3D models of both the tracked and the augmenting objects relies on OpenSceneGraph, a flexible 3D rendering engine. Figures 2 shows the results of the tracking and the augmenting task on two different sequences, with fixed and hand held cameras. Despite some image blur, cluttered background or shaky displacements the object is correctly augmented throughout the sequences. Though not being real-time, our algorithm runs at a 10 fps framerate on these sequences, with 640×480 images.

Using a synthetic image sequence featuring a Spot satellite and provided with ground truth, we have also quantitatively verified the effectiveness and benefit of some of our contributions by evaluating the approach of [5], denoted by $C0$, the integration of the color features with robust estimators ($C1$), and the addition of temporal

Figure 2: Tracking and augmenting Mario.

Table 1: RMS errors: t_x, t_y, t_z (in meters) and R_x, R_y, R_z (in radians) respectively refer to translation and rotation (Euler angles) parameters.

Mode	t_x	t_y	t_z	R_x	R_y	R_z
C0	0.118	0.238	1.771	0.158	0.069	0.016
C0, C1	0.082	0.114	0.517	0.066	0.037	0.017
C0, C1, C2	0.076	0.090	0.486	0.055	0.038	0.014

consistency for the color-based function ($C2$). With the approach proposed in [4], which is equivalent to $C0$ with $C1$ but without the multiple hypotheses framework and the robust estimators, tracking gets lost on this sequence. Tracking results can be seen on Figure 3 and are represented on Table 1 in terms of root mean square errors on the pose parameters over the sequence.
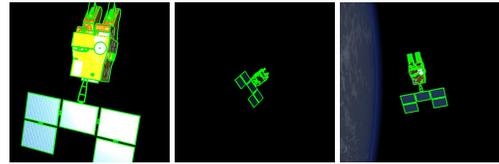

Figure 3: Tracking for the sequence with a Spot satellite using the proposed method.

6 CONCLUSION

In this paper, we present a robust model-based tracking algorithm for augmented reality applications. By efficiently managing the complete given or reconstructed 3D model and the extraction of its sharp edges, we are able to conveniently deal with any kind of 3D object. The robust combination of geometrical and color edge-based features makes of a criterion whose minimization provides a robust and accurate estimate of the complete pose, as validated by carried out AR experiments. Future works would aim at improving computational aspects for real-time concerns.

REFERENCES

[1] G. Bleser, Y. Pastarmov, and D. Stricker. Real-time 3d camera tracking for industrial augmented reality applications. *Journal of WSCG*, pages 47–54, 2005.

[2] A. Comport, E. Marchand, M. Pressigout, and F. Chaumette. Real-time markerless tracking for augmented reality: the virtual visual servoing framework. *IEEE Trans. on Visualization and Computer Graphics*, 12(4):615–628, July 2006.

[3] T. Drummond and R. Cipolla. Real-time visual tracking of complex structures. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(7):932–946, July 2002.

[4] G. Panin, E. Roth, and A. Knoll. Robust contour-based object tracking integrating color and edge likelihoods. In *Proc. of the Vision, Modeling, and Visualization Conference 2008, VMV 2008*, pages 227–234, Konstanz, Germany, Oct. 2008.

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

[6] L. Vacchetti, V. Lepetit, and P. Fua. Combining edge and texture information for real-time accurate 3d camera tracking. In *ACM/IEEE Int. Symp. on Mixed and Augmented Reality, ISMAR'04*, pages 48–57, Arlington, VA, Nov. 2004.