

# HYBRID TRACKING APPROACH USING OPTICAL FLOW AND POSE ESTIMATION

Muriel Pressigout\*, Éric Marchand<sup>+</sup> and Étienne Mémin<sup>°</sup>

\*IETR Image group/INSA de Rennes, Rennes, France

<sup>+</sup>INRIA, IRISA, Rennes, France

<sup>°</sup>Université de Rennes 1, IRISA, Rennes, France

## ABSTRACT

This paper proposes a hybrid approach to estimate the 3D pose of an object. The integration of texture information based on image intensities in a more classical non-linear edge-based pose estimation computation has proven to highly increase the reliability of the tracker. We propose in this work to exploit the data provided by an optical flow algorithm for a similar purpose. The advantage of using the optical flow is that it does not require any *a priori* knowledge on the object appearance. The registration of 2D and 3D cues for monocular tracking is performed by a non linear minimization. Results obtained show that using optical flow enables to perform robust 3D hybrid tracking even without any texture model.

*Index Terms*— Hybrid tracking, robust tracking

## 1. INTRODUCTION

This paper addresses the problem of robust model-based tracking of 3D objects using a monocular vision system. It proposes to integrate texture information based on the optical flow estimation in an edge-based process to obtain a spatio-temporal tracker. The aim is to be accurate and more robust to textured environments than classical 3D trackers without adding any knowledge about the object texture.

Among the “classical tracker”, both contour-based trackers (eg [1, 2, 3]) and textured-based trackers (eg [4, 5]) have complementary advantages and drawbacks. The idea is then to integrate both approaches in the same process. Among approaches to cue integration one can find:

- a sequential use of the available information (mainly motion and edges) [6, 7, 8]. In these approaches, motion estimation (dominant motion or optical flow) provides a prediction of the edge (*ie*, of the 2D object location) which is helpful for the contour-based registration step and improves tracking reliability.
- probabilistic approaches. Most of these approaches rely on the well known Kalman filter, its non-linear version the Extended Kalman Filter (EKF) or particle filter. [9] integrates the outputs from two trackers (a 3D model-based tracker [2] and a point of interest tracker) using an EKF. [10] fuses contour-based tracking and optical-flow estimation within an Iterated Extended Kalman Filter to update object position. Let note that many approaches rely on a particle filtering or

Probabilistic Multiple Hypothesis Tracker (PMHT) but are usually very slow.

- In [3] the proposed model-based approach considers both 2D-3D matching against a key-frame that represents a single pose as in a classical model-based approach but considering multiple hypotheses for the edge tracking and 2D-2D temporal matching (which introduces multiple view spatio-temporal constraints in the tracking process). A nice extension is proposed in [11] to integrate contribution of a contour-based tracker similar to [1, 2].

However, adding texture information requires some more knowledge about the object: keyframe or reference images for example. Our aim is then to use hybrid technics such that the only 3D information needed is the 3D model required for the contour-based 3D tracker while keeping accuracy tracking. Texture information based on the optical flow provides texture-based features by using only the images to be processed. Considering optical flow [12, 13, 14, 15], the dense estimation of the motion fields gives precise point correspondences without need to pay attention to the feature selection. Using optical flow or dominant motion estimation in 3D tracking has been already studied but mainly for edge-based pose estimation initialization [7, 8]. This paper presents a hybrid 3D tracker that integrates more closely a camera displacement computation based on the optical flow estimation in a classical edge-based tracker. Camera pose and displacement estimation are both computed thanks to a unified full scale non-linear minimization that consider edge-based information and motion field. To improve robustness, M-estimator are considered at each level of the algorithm.

Section 2 describes the general framework of our tracker. The texture-based features are then described more precisely in Section 3. Results on a video sequence are shown in Section 4.

## 2. HYBRID TRACKER: GENERAL FRAMEWORK

The basic principle of the proposed approach is an adaptation of the one presented in [16]. The general overview will be briefly summed up in this section. This paper will focus on the fact that the withdrawal of the texture model does not lead to a less accurate tracker.

The approach consists of estimating the real camera pose  ${}^t\mathbf{M}_o$  by minimizing the error  $\Delta$  between the observed data  $\mathbf{s}^*$  and the current value  $\mathbf{s}$  of the same features computed using the model according to the current pose:

$$\Delta = \sum_{i=1}^N \rho(s_i(\mathbf{r}) - s_i^*)^2, \quad (1)$$

\*The first author performed the work while her PhD at Université de Rennes 1, IRISA, Rennes, France

where  $\rho(u)$  is a robust function introduced in the objective function in order to reduce the sensitivity to outliers (M-estimation) and  $\mathbf{r}$  is a vector-based representation of the pose  ${}^t\mathbf{M}_o$ . A virtual camera, defined by its position  $\mathbf{r}$  in the object frame, can be virtually moved in order to minimize this error. At convergence the position of the virtual camera will be aligned with the real camera pose. This can be achieved by considering a simple control law given by  $\mathbf{v} = -\lambda(\widehat{\mathbf{D}}\widehat{\mathbf{L}}_s)^+\mathbf{D}(\mathbf{s}(\mathbf{r}) - \mathbf{s}^*)$  where  $\mathbf{v}$  is the velocity screw of the virtual camera,  $\mathbf{L}_s$  is the interaction matrix or image Jacobian related to  $\mathbf{s}$  and defined such as  $\dot{\mathbf{s}} = \mathbf{L}_s\mathbf{v}$  and  $\mathbf{D}$  is a diagonal weighting matrix given by  $\mathbf{D} = \text{diag}(w_1, \dots, w_k)$ . The weights  $w_i$  reflect the confidence in each feature and their computation is based on M-estimators and is described in [1, 2].

$\mathbf{s}$  can be a contour-based feature or a texture-based one. Depending on the nature of the features, this approach can solve a pose computation problem or a camera displacement one. As presented in [16], combining in this framework both approaches allows to introduce a spatio-temporal constraint in the pose estimation by considering information in the current and past images and the underlying multi-view geometrical constraints. Equation (1) can be rewritten as:

$$\Delta = \Delta_1({}^t\mathbf{M}_o) + \Delta_2({}^t\mathbf{M}_o) \quad \text{with} \quad \Delta_1 = \sum_{i=1}^M \rho_1(d_{\perp}(\mathbf{p}_i, \mathcal{C}({}^t\mathbf{M}_o))) \quad (2)$$

that have to be optimized for the pose  ${}^t\mathbf{M}_o$ .

$\Delta_1$  accounts for the contour-based part of the tracker and represents the distance in the image between point extracted using a low-level IP algorithm (local edge tracker) and the projection of the contour  $\mathcal{C}(\cdot)$  for a given pose.

$\Delta_2$  accounts for texture-based part of the tracker and makes use of two consecutive frames. Rather than estimating directly the pose it allows to compute the displacement between two successive frames.

In [16], the texture-based features used in  $\Delta_2$  are intensity values, which implies to acquire off-line some key images of the object representing its texture and to compute the camera pose for each of them. It may be too restrictive in some situations, as for example outdoors environments or scenes with many planes. Our goal is therefore to perform robust registration using a hybrid tracking with less *a priori* knowledge or off-line computation as possible. In this paper, we propose a hybrid tracker without a texture model. The objective is to obtain a tracker that is based only on a classical 3D model of the object but that fuses both contour-based and texture-based features in order to keep the advantages of a hybrid tracker.

Since the contour-based features are similar to the ones used in our previous work [16] or in a classical model-based tracker [2, 1], the contour-based features related to  $\Delta_1$  will not be described in this paper. In our case we have used the approach fully described in [1]. We will focus in the next section on the texture-based ones.

### 3. FEATURES BASED ON OPTICAL FLOW ESTIMATION

The basic idea is to use the optical flow to provide information related to the camera displacement thanks to the relation that links the position of point in two images acquired by a moving camera. Let note  ${}^t\mathbf{x}$  the image points in the image  $\mathbf{I}_t$  and  ${}^{t'}\mathbf{x}$  their correspondent

in the next image  $\mathbf{I}_{t'}$ . The optical flow estimation between these successive images is performed as described in the next paragraph. The resulting point correspondence  $({}^t\mathbf{x}, {}^{t'}\mathbf{x})$  is then used in the tracking framework to estimate the camera displacement as explained in paragraph 3.2.

#### 3.1. Optical flow estimation

The most accurate techniques to address the generic problem of estimating the apparent motion from image sequences are based on the seminal work of Horn and Schunck [12]. These techniques are formalized as the minimization of a global cost function  $\mathcal{H}$  composed of two terms ( $\mathcal{H} = \mathcal{H}_{obs} + \mathcal{H}_{reg}$ ). The first one is derived from a *brightness constancy* assumption and assumes that a given point keeps the same intensity along its trajectory. It is expressed through the well known optical flow constraint equation (OFCE):

$$\mathcal{H}_{obs}(\mathbf{I}, \mathbf{v}) = \iint_{\Omega} \rho_2 \left[ \nabla \mathbf{I}(\mathbf{p}, t) \cdot \mathbf{v}(\mathbf{p}, t) + \frac{\partial \mathbf{I}(\mathbf{p}, t)}{\partial t} \right]^2 d\mathbf{p}, \quad (3)$$

where  $\mathbf{v}(\mathbf{p}, t) = (u, v)^T$  is the unknown velocity field at time  $t$  and location  $\mathbf{p} = (x, y)$  in the image plane  $\Omega$ ,  $\mathbf{I}(\mathbf{p}, t)$  being the image brightness, viewed for a while as a continuous function. Function  $\Phi$  is a penalty function that can be quadratic or a so called robust penalty function to limit the impact of locations where the brightness consistency assumption is violated [13, 14]. The single (scalar) brightness consistency equation does not allow to estimate the velocity vectors. In order to solve this ill-posed problem, it is common to employ the additional smoothness constraint  $\mathcal{H}_{reg}$ . Usually, this second term enforces a spatial smoothness coherence of the flow field. It relies on a contextual assumption which enforces a spatial smoothness of the solution. This term usually reads:

$$\mathcal{H}_{reg}(\mathbf{v}) = \alpha \iint_{\Omega} \rho_3 [|\nabla u(\mathbf{p}, t)| + |\nabla v(\mathbf{p}, t)|^2], \quad (4)$$

The penalty function  $f_2$  is a usually a non quadratic robust penalizer in order not to smooth out the natural discontinuities of the velocity field [13, 14, 15].  $\alpha > 0$  is a parameter controlling the balance between the smoothness constraint and the global adequacy to the observation assumption.

To handle large displacements and for a better computational efficiency, the associated successive minimization are also usually performed using efficient multigrid iterative methods [14, 15].

#### 3.2. Computing the camera displacement

The displacement vectors  $\mathbf{v}^i$  obtained from the optical flow estimation process enable to get the point correspondences  $(\mathbf{p}_t^i, \mathbf{p}_{t'}^i)$  between two successive images by selecting the  $\mathbf{p}_t^i$  in the image  $\mathbf{I}_t$  and computing their correspondent  $\mathbf{p}_{t'}^i$  following:

$$\mathbf{p}_{t'}^i = \mathbf{p}_t^i + \mathbf{v}^i \quad (5)$$

These pixel correspondences give the texture-based part of the objective function :

$$\Delta_2 = \sum_{i=1}^N \rho_2 ({}^t tr_t ({}^{t'}\mathbf{x}^i) - {}^t\mathbf{x}^i), \quad (6)$$

where  ${}^t tr_t(\cdot)$  is the transfer function between image  $\mathbf{I}_t$  and  $\mathbf{I}_{t'}$ .

The next paragraphs describe the transfer function  ${}^t tr_t(\cdot)$  and the interaction matrix needed in the non-linear minimization step.

### 3.2.1. Point transfer

Let us note that in the general case (that is a non-planar scene viewed by a camera which rotates and translates), the point transfer can be achieved, using multiple images, considering the epipolar geometry and the essential or fundamental matrices. In order to handle any kind of camera displacement, this section will be restricted to the less general case where point transfer can be achieved using an homography. Indeed, some particular cases (planar scene, pure rotational camera motion) lead the  $2D$  transformation between two images to be a homography. In that case, a point  ${}^1\mathbf{x}$  in image 1 expressed in homogeneous coordinates  ${}^1\mathbf{x} = ({}^1x, {}^1y, {}^1w)$ , is transferred in image 2 as a point  ${}^2\mathbf{x}$ , considering the following relation:

$${}^2\mathbf{x} = {}^2tr_1({}^1\mathbf{x}) = \alpha {}^2\mathbf{H}_1 {}^1\mathbf{x} = \alpha ({}^2\mathbf{R}_1 + \frac{{}^2\mathbf{t}_1}{{}^1d} {}^1\mathbf{n}^\top) {}^1\mathbf{x} \quad (7)$$

where  ${}^2\mathbf{H}_1$  is an homography (defined up to scale factor  $\alpha$ ) that defines the transformation between the images acquired by the camera at pose 1 and 2.  ${}^1\mathbf{n}$  and  ${}^1d$  are the normal and distance to the origin of the reference plane expressed in camera 1 frame.  ${}^2\mathbf{R}_1$  and  ${}^2\mathbf{t}_1$  are the rotation matrix and the translation vector between the two camera poses.

Let note that this transfer function based on the homography can be generalized to the general case of a non-planar structure as in [16] by adding a parallax term that represents the relative depth of the point with respect to a virtual plane.

### 3.2.2. Interaction matrix

The interaction matrix  $\mathbf{L}_s$  is the interaction matrix that links the variation of the point position  ${}^2\mathbf{x} = {}^2tr_1({}^1\mathbf{x})$  to the camera motion [17]:

$$\mathbf{L}_s = \begin{pmatrix} -1/Z & 0 & x/Z & xy & -(1+x^2) & y \\ 0 & -1/Z & y/Z & (1+y^2) & -xy & -x \end{pmatrix} \quad (8)$$

The depth information  $Z$  is computed at each iteration from the coordinates  $(x, y)$  and from the equation of the reference plane updated from the current camera displacement:

$$1/Z = \frac{{}^1d - {}^2\mathbf{t}_1 {}^1\mathbf{n}}{{}^2\mathbf{R}_1 {}^1\mathbf{n} {}^1\mathbf{x}}$$

## 4. RESULTS

This Section presents some tracking results to highlight the accuracy of this hybrid tracker with respect to single cue trackers. Moreover, we use as ground-truth the hybrid tracker described in [16] that uses a texture model to show that no loss of accuracy is observed.

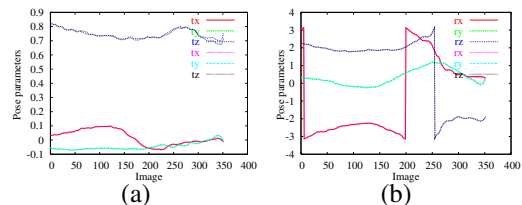
The edge locations and texture points used in the minimization process are displayed in the first image (blue crosses for the texture sample locations and red crosses for the edge locations). In the next images, only the forward-projection of the model for a given pose is displayed in green. For the texture location selection, a regular sample is done from the center of each face. The number of texture points tracked by the optical flow and taken into account in the

minimization process by face is proportionnal to its visibility in the image.

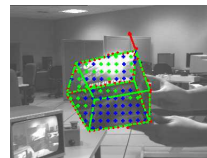
In the video sequence, the object to track is a box. The difficulty is to deal with the specularities, changes in illumination and faces apparition/disappearance that occur during the sequence.

Drift is observed when only the optical flow is used to estimate the object pose. Indeed, the errors due to the spatial constraints in the optical flow computation and displacement integration are not corrected since there is no permanent reference image. As far as the edge-based tracker is concerned, the main reason of failure is the rotation of the object around itself. When faces appear or disappear, the geometrical features to be tracked also change which may skew the low level process that extracts the contours in the image. The two hybrid trackers, the one presented in this paper using the optical flow for the texture part and the one presented in [16] using a texture model, succeed to deal with this problem, despite the fact there are also strong specularities as shown in Figure 2.

Camera position and orientation parameters are displayed in Figure 1 for the two hybrid trackers. One can see that the pose estimation is similar in both case though the hybrid tracking based on the optical flow exploits less *a priori* knowledge about the object (only the *CAD* model). Let us note that the smoothness of the estimated parameter curves is directly due to the simultaneous fusion of two complementary cues since no Kalman filter is used as in [10, 9]. This means that this kind of hybrid tracker can be used for applications where no object motion model is known, augmented reality for example.

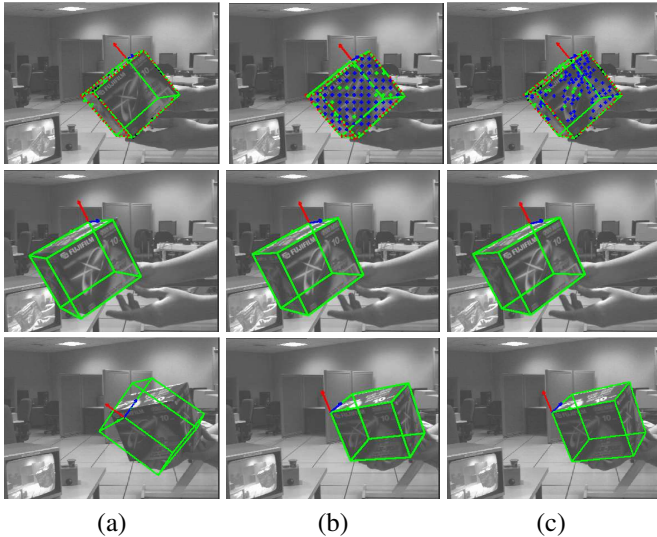


**Fig. 1.** Pose parameters evolution for the two hybrid trackers. (a) position parameters, (b) orientation parameters. Even though no texture model is used with the presented hybrid tracker, the result remains as accurate as the hybrid tracker based on reference images.



**Fig. 2.** An example of specularities observed during the sequence. M-estimators reject the outlier data (green points), the remaining points being considered as inliers (blue points).

For both hybrid trackers, occlusions or errors in the contour point tracking due to a textured environment are handled correctly thanks to the M-estimators. Figure 2 shows an example of specularities that occur during the tracking.



**Fig. 3.** Pose estimation. Comparison of the different trackers. The hybrid tracker (b) presented in this paper, that integrates the optical flow in the classical edge-based tracker, enables to improve the pose computation as in the hybrid tracker (c) with the same 3D knowledge on the object as in the classical edge-based tracker (a) (while the hybrid tracker(c) uses a texture model).

## 5. CONCLUSION

We are interested in hybrid trackers that enable to robustify a classical edge-based tracker by adding some information based on the object texture. The complementarity of the two types of features helps to deal with a wider range of situations. In this paper, the choice of the texture-based features relying on the optical flow estimation enables to perform this integration without adding a texture model (image templates) contrarily to previous works. We have chosen to estimate motion of the object using a very precise optical flow estimator. Nevertheless other approaches can be considered within the same scheme such as dominant motion estimation, point tracker (eg, KLT) or image-based template registration. A problem of drift may appear, requiring to handle template update properly, however this should allow a better computational efficiency.

## 6. REFERENCES

- [1] A.I. 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*, vol. 12, no. 4, pp. 615–628, July 2006.
- [2] T. Drummond and R. Cipolla, "Real-time visual tracking of complex structures," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 932–946, July 2002.
- [3] L. Vacchetti, V. Lepetit, and P. Fua, "Stable real-time 3d tracking using online and offline information," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 26, no. 10, pp. 1385–1391, October 2004.
- [4] G. Hager and P. Belhumeur, "Efficient region tracking with parametric models of geometry and illumination," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 20, no. 10, pp. 1025–1039, Oct. 1998.
- [5] S. Benhimane and E. Malis, "Homography-based 2d visual tracking and servoing," *Int. Journal of Computer Vision*, 2007, Special IJCV/IJRR issue on vision for robots.
- [6] B. Bascle, P. Bouthemy, N. Deriche, and F. Meyer, "Tracking complex primitives in an image sequence," in *Int. Conf. on Pattern Recognition, ICPR '94*, Jerusalem, Oct. 1994, pp. 426–431.
- [7] E. Marchand, P. Bouthemy, F. Chaumette, and V. Moreau, "Robust real-time visual tracking using a 2D-3D model-based approach," in *IEEE Int. Conf. on Computer Vision, ICCV'99*, Kerkira, Greece, Sept. 1999, vol. 1, pp. 262–268.
- [8] T. Brox, B. Rosenhahn, D. Cremers, and H.-P. Seidel, "High accuracy optical flow serves 3-D pose tracking: exploiting contour and flow based constraints," in *European Conf. on Computer Vision, ECCV'06*, A. Leonardis, H. Bischof, and A. Pinz, Eds., Graz, Austria, May 2006, vol. 3952 of *LNCS*, pp. 98–111, Springer.
- [9] V. Kyrki and D. Kragic, "Integration of model-based and model-free cues for visual object tracking in 3d," in *IEEE Int. Conf. on Robotics and Automation, ICRA'05*, Barcelona, Spain, Apr. 2005, pp. 1566–1572.
- [10] M. Haag and H.H. Nagel, "Combination of edge element and optical flow estimates for 3D-model-based vehicle tracking in traffic image sequences," *Int. Journal of Computer Vision*, vol. 35, no. 3, pp. 295–319, Dec. 1999.
- [11] 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'2004*, Arlington, Va, Nov. 2004, vol. 2, pp. 48–57.
- [12] B. Horn and B. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, pp. 185–203, 1981.
- [13] M. Black and P. Anandan, "The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields," *Computer Vision and Image Understanding*, vol. 63, no. 1, pp. 75–104, 1996.
- [14] E. Mémin and P. Pérez, "Hierarchical estimation and segmentation of dense motion fields," *Int. J. of Computer Vision*, vol. 46, no. 2, pp. 129–155, February 2002.
- [15] A. Bruhn, J. Weickert, T. Kohlberger, and C. Schnrr, "A multigrid platform for real-time motion computation with discontinuity-preserving variational methods," *IJCV*, vol. 70, no. 3, pp. 257–277, 2006.
- [16] M. Pressigout and E. Marchand, "Real-time hybrid tracking using edge and texture information," *Int. Journal of Robotics Research, IJRR*, vol. 26, no. 7, July 2007.
- [17] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *IEEE Trans. on Robotics and Automation*, vol. 8, no. 3, pp. 313–326, June 1992.