Combining edge and texture information for real-time accurate 3D camera tracking *

Luca Vacchetti Vincent Lepetit Pascal Fua
CVlab
Swiss Federal Institute of Technology
1015 Lausanne, Switzerland
{luca.vacchetti,vincent.lepetit,pascal.fua}@epfl.ch

Abstract

We present an effective way to combine the information provided by edges and by feature points for the purpose of robust real-time 3–D tracking. This lets our tracker handle both textured and untextured objects. As it can exploit more of the image information, it is more stable and less prone to drift that purely edge or feature-based ones.

We start with a feature-point based tracker we developed in earlier work and integrate the ability to take edge-information into account. Achieving optimal performance in the presence of cluttered or textured backgrounds, however, is far from trivial because of the many spurious edges that bedevil typical edge-detectors. We overcome this difficulty by proposing a method for handling multiple hypotheses for potential edge-locations that is similar in speed to approaches that consider only single hypotheses and therefore much faster than conventional multiple-hypothesis ones.

This results in a real-time 3-D tracking algorithm that exploits both texture and edge information without being sensitive to misleading background information and that does not drift over time.

1. Introduction

Most markerless tracking systems rely on either contours or interest points because they are both abundant on everyday objects and easy to extract. Both these features have advantages and inconvenients. Interest points, such as in [6] are very well adapted to textured objects and robust to geometrical distorsion and to light changes. Unfortunately, they become rare and unstable on poorly textured objects, and they are not invariant to scale changes. By contrast, contour points are informative for scenes with sharp edges and strong contrast changes, but less so in cluttered and textured scenes. In practice, there is no such sharp distinction between textured objects and objects with sharp edges. Therefore, the two information sources are complementary, and it is interesting to combine them for markerless camera tracking.

The integration of these two sources should be straightforward: Once the image primitives have been matched with their correspondences on the 3D model, the camera viewpoint can be estimated by minimizing the reprojection error of the different primitives. Integrating the edge information into the interest point-based tracker [20] tends to degrade the results instead of improving them. This is due to many errors made when matching contour primitives, mainly because of contour ambiguities. Such ambiguities can be due to strong texture on the object or background clutter. One may consider only the edges most likely to be stable. It is not a satisfying solution because it requires the user to manually select them. Furthermore, as shown in Fig. 5, even if the object is perfectly sharp on an uniform background, moving the camera results in aspect changes and in some edges projecting very close to one another. Possible solutions would be to predict which edges are going to get close and to disable them, or alternatively to assign lower weight to those edges, but this would remove an important source of information.

We propose an efficient and simple approach considering multiple hypotheses. These hypotheses are first established using a technique similar to the one retained by state of the art edge-based trackers [5, 15]. In our case we keep several hypotheses instead of only one. Then, instead of keeping only the best one, we

*This work was supported in part by the Swiss Federal Office for Education and Science.

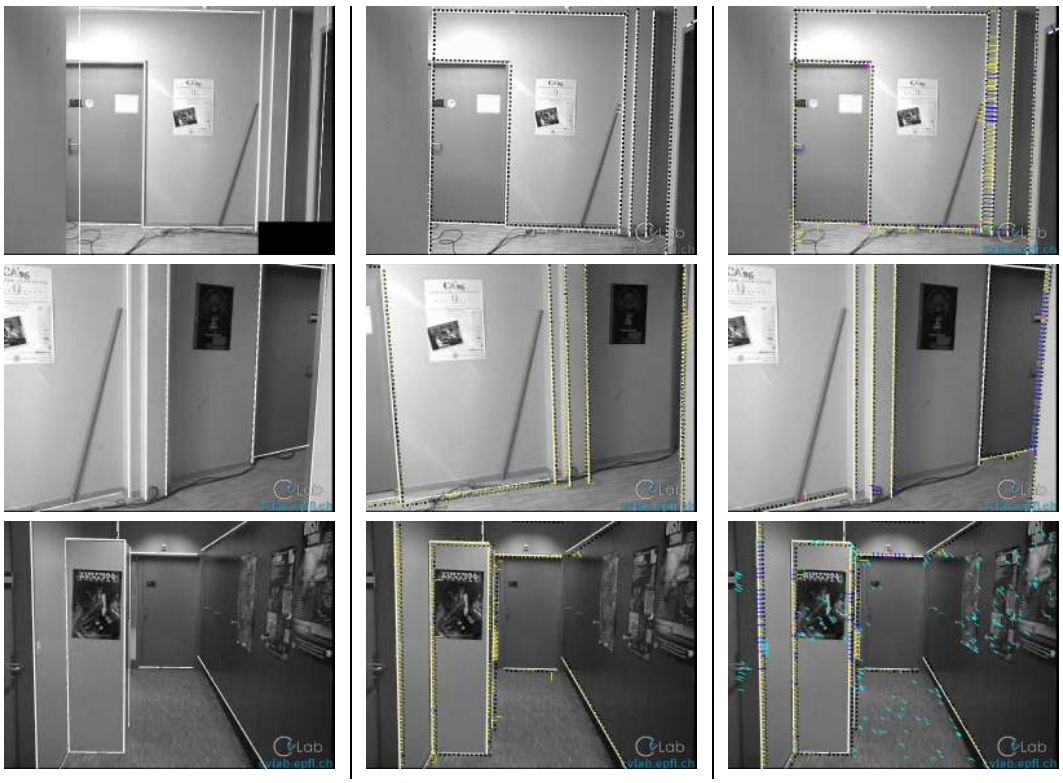


Figure 1. Corridor Sequence, combining edges and feature points. *First column:* **Using the earlier tracker based on reference frames and interest points only, the 3D model edges are not always reprojected at the right place because there is little texture. Four reference frames were used.** *Second column:* **Simple-minded integration of edge-based information does not improve accuracy and, sometimes, even degrades it.** *Third column:* **The integrated tracker we propose succesfully tracks the sequence. As can be seen in the submitted video sequence, there is no jitter, and the 3D model edges are always reprojected at the right places. No reference frame was used.**

retain several. The correct one is selected during the optimization of the pose parameters, using a robust estimator that we developed for this purpose.

Considering several hypotheses makes the tracking more robust because it is not perturbed by strong misleading contours, and more accurate because all the information is used. Our method is also fast: there is not much additional computation cost, and the tracking easily runs in real-time. Finally, this method lets us consider a much larger search-space, leading to improved handling of large and high speed displacements.

As a result, we were able to increase the range of applicability of an earlier feature-points based tracker [20] by allowing it to also use edge information. Not only is the improved tracker able to handle both textured and untextured objects but, unlike the earlier one, does not require the use of keyframes to avoid drift. These improvements are highlighted by Figs 2, 3 and 1.

The generality and robustness of this method make it suitable for a direct application into AR scenarios, where it can be a key feature for solving the registration problem in real-time.

In the remainder of the paper, we first discuss related work. Section 3 explicits our approach to multiple hypotheses handling. Section 4 describes the integration of the two sources of information, and our experiments and results are presented in Section 5.

2. Related Work

While interest points can be reliably characterized by the neighbouring texture, contours information is much more ambiguous, and it is necessary to consider several possibilities when matching models against image contours.

In the context of contour-based object recognition,

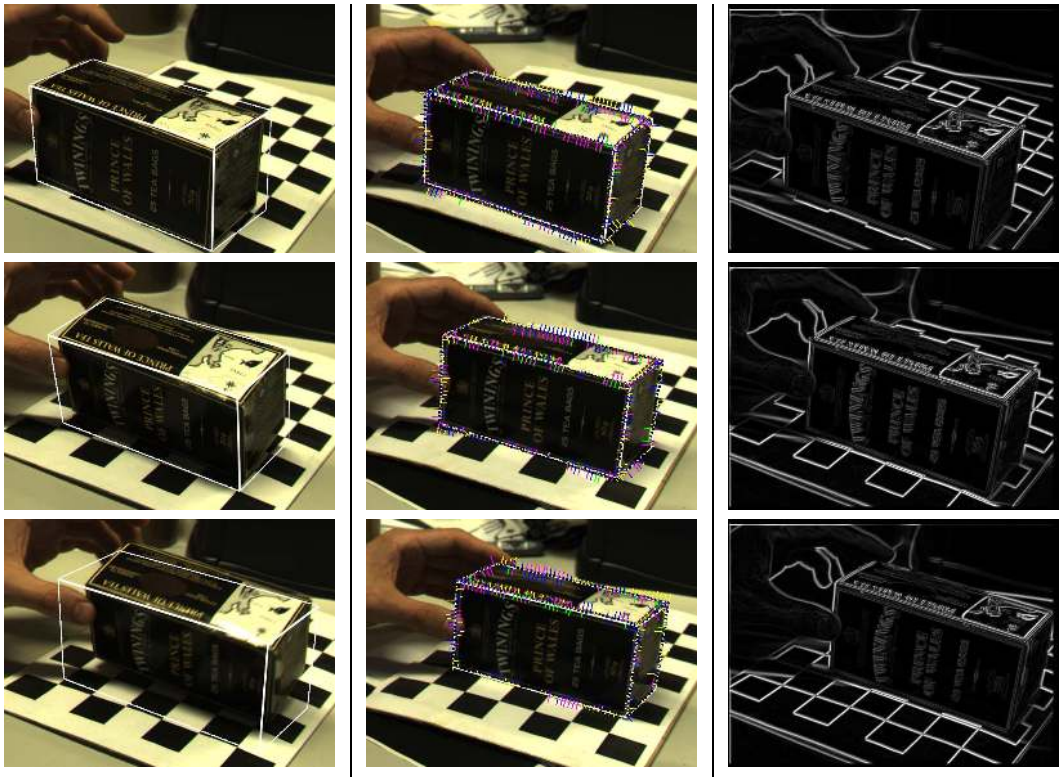


Figure 2. BOX SEQUENCE. *First column:* **When tracking using interest points and considering only one hypothesis per edge point, the top edge of the box 3D model is attracted by the strong edges on the checkerboard.** *Second column:* **Considering multiple hypotheses as we propose allows to get the 3D model correctly reprojected.** *Third column:* **The gradient maps show the strong edges present on the checkerboard.**

multiple hypotheses for such correspondences are always considered, for example by using a Generalized Hough transform [11], a stochastic optimisation [3], or a robust graph matching [13]. Unfortunately, the combinatorics can quickly become very large and make such approaches unpractical for real-time applications.

Condensation is a more efficient way to maintain multiple hypothesis over time while tracking, where particles represent the probability distribution of the target position. It has been used to successfully track poorly textured objects such as hands [9] or human bodies [19] in dense visual clutter. Unfortunately, the Condensation approach would be too slow in our context, the observation process being too costly to be applied to each particle. Another drawback of Condensation is its known tendancy to recover jittering trajectories [10], that makes this solution not suitable for Augmented Reality applications. A post-processing can be applied, but obviously not for real-time tracking.

For efficiency reasons, edge-based camera trackers search the edge correspondents in a restricted area of the image, around their predicted positions. This search can be done for curves [18], segments [14] or points sampled on the model edges [5, 15]. Then the pose is estimated by minimizing the reprojection error, using a robust estimator to remove the spurious matches. In these works, the correspondent is usually chosen as the point with the highest gradient value. Nevertheless there is no actual reason to justify this choice, and this can result in a failure when the tracking is "attracted" by an incorrect contour with a strong gradient, even if robust estimation is used to reject outliers. The solution we propose makes the tracking more robust because it is not perturbed by strong misleading contours, and more accurate because more information are used.

To our knowledge, there are relatively few published approaches to the integration of the texture and contour information. [4] combine optical flow and edge

Figure 3. CORRIDOR SEQUENCE, USING EDGE INFORMATION ONLY. *First column:* **When considering only one hypothesis per edge point and no interest points, the right side of the door is attracted by the right wall (pictures 4 and 5) and the tracking eventually fails.** *Second column:* **Considering multiple hypotheses as we propose makes the tracking more robust. Nevertheless the reprojected 3D model sometimes jitters. Integrating the texture information suppresses this problem.**

information but consider faces. [16] propose a nice extension of [12] to integrate contour information but is limited to planar objects. By contrast, our method

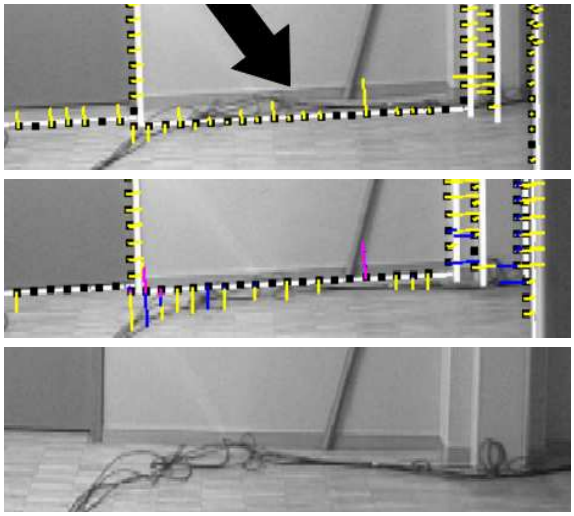


Figure 4. DETAIL OF THE CORRIDOR SEQUENCE. **Top picture: When considering only one hypothesis, the tracking can be perturbed by misleading strong contours, like the black cable that attracts the edge corresponding to the wall base. Middle picture: Our method avoids these errors. Bottom picture: The cable alone.**

is generic. It extracts interest points from the texture, which we believe to be more reliable than optical flow.

3. Considering Multiple Hypotheses when Tracking Edges

In this section we outline our edge matching approach that handles multiple hypotheses. We first describe how we generate these hypotheses, then we discuss how to select the correct one by means of our robust estimator.

3.1. Establishing Hypotheses

We rely on a similar approach than the one used in [15, 5], and introduced earlier in the Moving Edges algorithm [1] and in the RAPID tracker [7]. The only difference at this stage is that we retain several hypotheses, instead of only one.

As described by Fig. 6.a, during tracking, the CAD model of the scene is reprojected in the image at time t from the camera predicted position. To be general, we do not use any motion model and this predicted position is simply the viewpoint estimated for the image at time $t-1$. Points $e_{i,j}$ are first sampled along

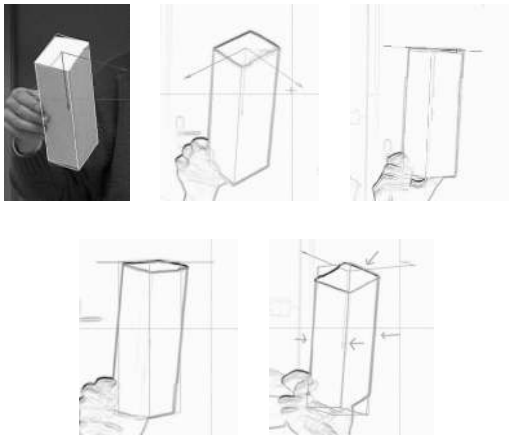


Figure 5. CONSIDERING SINGLE HYPOTHESES. **Even in the case of a simple object like this white box, and in absence of misleading contours, the tracking can get stuck in a wrong position because of the ambiguities. It is not simply due to a local minimum problem: The wrong and the correct positions approximately give the same value for the minimized objective function.**

the reprojection of the edges E_i in the CAD model. Then, for each point $e_{i,j}$, a local search is performed on a scan line in the direction of the reprojected contour normal. Previous methods attribute to $e_{i,j}$ one correspondent $e'_{i,j}$ located at the strongest discontinuity along the scan line. By contrast, we attribute to $e_{i,j}$ all the local extrema of the gradient along the scanline as potential correspondents $e'_{i,j,k}$ as shown in Fig. 6.b.

The search is fast because it is limited to a mono dimensional search path and it does not require any prior edge extraction. As in [2], we use a precomputed convolution kernel function of the contour orientation to find only edges with an orientation similar to the reprojected contour orientation, not all edges in the scanline.

In the single hypothesis case, this approach allows to estimate the camera viewpoint P_t that minimizes v_t given by:

$$v_t = \frac{1}{N_e} \sum_i \sum_j \rho \left(\Delta_t(E_i, e'_{i,j}) \right) \tag{1}$$

where

- $\Delta_t(E, e)$ is the squared distance between the 2D point e' and the 3D contour E reprojected on the image plane according to the projection P_t;
- ρ is an robust estimator used for reducing the influence of wrong matches;
- N_e is the number of sampled points $e'_{i,j}$ along the reprojected contours.

We applied this technique to try and retrieve the camera trajectory for the corridor sequence of Fig. 3. The camera internal parameters were known and fixed, and we used the Tukey estimator for the ρ estimator [8]. The results are presented in the first column of Fig. 3. The tracking is corrupted by misleading contours and quickly fails. In the following, we show how to consider multiple $e'_{i,j,k}$ hypotheses to avoid such problems.

3.2. A Multiple Hypotheses Robust Estimator

In order to efficiently consider multiple hypotheses, we introduce a new robust estimator built on the Tukey estimator. The Tukey estimator ρ_{Tuk} is computed as:

$$\rho_{\text{Tuk}}(x) = \begin{cases} \frac{c^2}{6} \left[1 - \left(1 - \left(\frac{x}{c} \right)^2 \right)^3 \right] & \text{if } |x| \leq c \\ \frac{c^2}{6} & \text{if } |x| > c \end{cases}$$

where c is a threshold chosen with respect to the standard deviation of the data.

Our new estimator denoted ρ^*_{Tuk} onwards is a multivariate function that can be expressed as:

$$\rho^*_{\text{Tuk}}(x_1, \ldots, x_n) = \min_i \rho_{\text{Tuk}}(x_i)$$

For example, Fig. 7 depicts the function $\rho^*_{\text{Tuk}}(u - u_1, u - u_2, u - u_3)$. Intuitively, this estimator takes several residuals, but only the residual closest to 0 has an influence on the final objective function. When all the values are too high, none of them has an influence.

We can now rewrite the term v_t of Equ. 1 in order to take into account for each point $e_{i,j}$ the $K_{i,j}$ hypotheses $e'_{i,j,k}$ established as described in the previous section. This term is now noted v^*_t to show that it uses our multiple hypotheses robust estimator:

$$v^*_t = \frac{1}{N_e} \sum_i \sum_j \rho^*_{\text{Tuk}} \left(\Delta_t(E_i, e'_{i,j,1}), \ldots, \Delta_t(E_i, e'_{i,j,K_{i,j}}) \right) \tag{2}$$

We use a numerical non-linear optimization to estimate P_t. This method has been used alone on the corridor sequence, the results are presented in the second column of Fig. 3. Compared to the single hypothesis tracking, it helps to improve the tracking robustness, but still lacks accuracy. In Section 4 we will show how to solve this problem by adding the texture information.

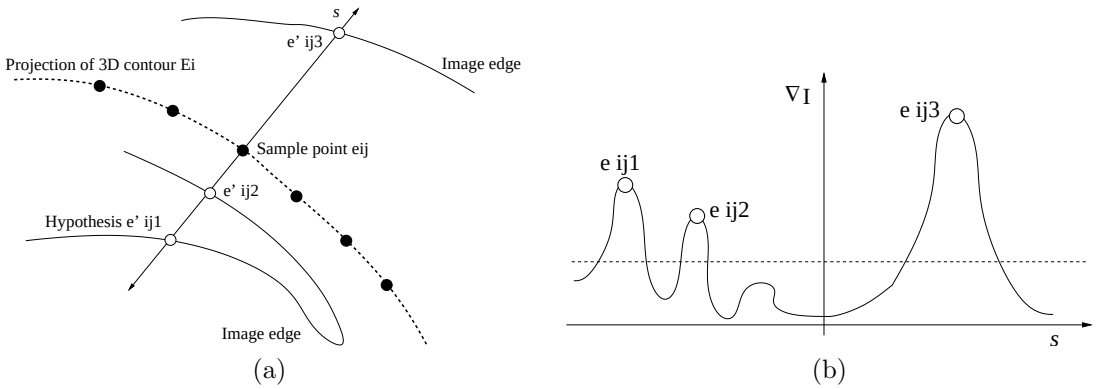


Figure 6. SEARCHING FOR MULTIPLE HYPOTHESES. **(a): Like previous methods, we search for the correspondent points along a scan line orthogonal to the reprojected contours. The difference is that we consider all the local extrema of the intensity gradient ∇I as potential correspondents (b).**

3.3. Relation with Gaussian Mixtures

Another approach to estimate the camera viewpoint using multiple hypotheses would be to maximize the likelihood:

$$P_t = \underset{P}{\operatorname{argmax}} \prod_i \prod_j p\Big(e'_{i,j,1}, \ldots, e'_{i,j,K_{i,j}} | P.E_i\Big)$$

where $P.E_i$ represents the reprojection of the contour E_i with respect to P. The term $p_{i,j} = p\Big(e'_{i,j,1}, \ldots, e'_{i,j,K_{i,j}} | P.E_i\Big)$ is an observation density that is usually expressed as a mixture of Gaussian distributions. More precisely, in our case the expression of $p_{i,j}$ would be:

$$p_{i,j} = \lambda + \sum_k G\left(\Delta_t\left(E_i, e'_{i,j,k}\right)\right)$$

This approach has a strong inconvenient in our case: when two hypotheses are too close to each other, the related peaks in the observation density tend to fuse. To illustrate this phenomenon, we have represented the graph of such a density in Fig. 7.c, streched so that it can be compared to our estimator ρ^*_{Tuk}.

That could result in an inaccuracy in the recovered pose because the minimum is not at the expected place, or a wrong hypothesis selection because the merged hypotheses have a larger weight. The second advantage of our robust estimator is that it relies on the Tukey estimator, that is known to be suitable to numerical optimization for camera registration [18, 2, 20].

4. Integration

To coherently merge the information from edges and texture, we will rely on the same approach we implemented in [20] to combine reference frames and previous frame information.

The texture information is handled by detecting Harris interest points (denoted m^i_t onwards) in the source image at every time step t. The 2D points m^i_{t-1} lying on the projected 3D model in the previous frame are matched with points m^i_t in the current frame.

These points are the projection of 3D points lying on the 3D model. Therefore, we have to simultaneously optimize the reprojection errors in these frames over the 3D position of these points, and over the viewpoints related to the previous and the current frames. The problem becomes:

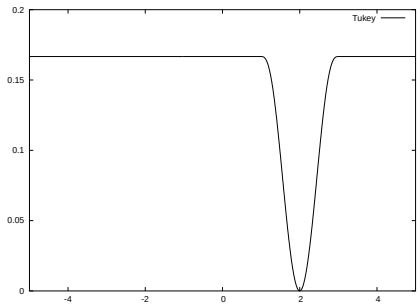
$$\min_{\substack{P_t, P_{t-1} \\ M_i}} \quad v^*_t + v^*_{t-1} + \sum_i s^i_t \tag{3}$$

with

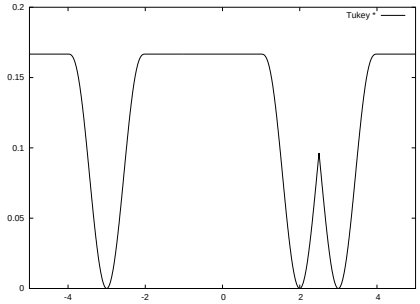
$$s^i_t = \rho_{\text{TUK}}\left(\phi_t(M_i, m^i_t) + \phi_{t-1}(M_i, m^{\nu(i)}_{t-1})\right),$$

where the interest point m^i_t detected in the current frame is matched with the point $m^{\nu(i)}_{t-1}$ detected in the previous frame. The term v^*_t has been introduced in Equ. 2 and corresponds to the edge contribution.

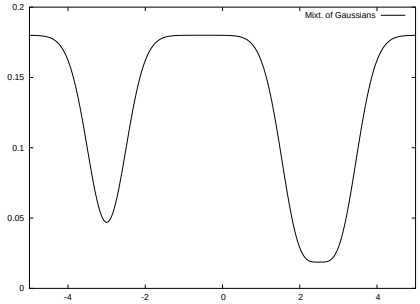
The important point here is that the 3D position M_i of the tracked points are also optimized, but constrained to stay on the 3D model surface. The formulation of this objective function allows us to satisfy



a. $\rho_{\text{Tuk}}(x-2)$



b. $\rho^*_{\text{Tuk}}\,(x-2, x-3, x-(-3))$



c. MixtureOfGaussians $(x-2, x-3, x-(-3))$

Figure 7. ADVANTAGE OF OUR ROBUST ESTIMATOR ρ^*_{TUK}, FOR THE 1D CASE. **a: Classical estimators consider a single hypothesis, contrary to the robust estimator we propose (b). c: Mixture of Gaussians can handle multiple hypotheses, but tend to merge hypothesis close to each other. They are also less suitable to numerical optimization.**

both the constraints from interest points matching between the successive frames and the contour information, without assumption of the accuracy of viewpoints

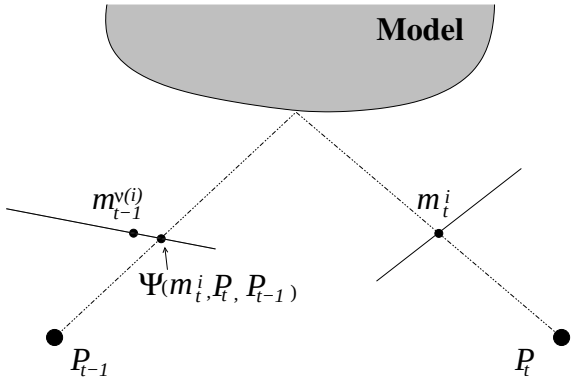


Figure 8. COMPUTING s^i_t. **The camera positions P_{t-1} and P_t are simultaneously optimized online.**

previously estimated. Equ. 3 can be rewritten as:

$$\min_{P_t, P_{t-1}} \left(v^*_t + v^*_{t-1} + \min_{M_i} \sum_i s^i_t \right) \tag{4}$$

since v^*_t and v^*_{t-1} are independent of the tracked points M_i.

As in [17], we eliminate the M_i to simplify the minimization problem: Instead of estimating the M_i, it is equivalent to estimate its projection in the two images. Then, according to [17], the terms s^i_t can be approximated using a transfer function that involves only the point reprojection. As depicted Fig. 8, such a transfer function $\Psi(m_1, P_1, P_2)$ returns the point m_2 so that it exists a 3D point M belonging to the model surface $m_1 = P_1 M$ and $m_2 = P_2 M$. Finally s^i_t is approximated by:

$$s^i_t = \rho_{\text{TUK}} \left(\left\| \Psi(m^{\nu(i)}_{t-1}, P_{t-1}, P_t) - m^i_t \right\|^2 + \left\| \Psi(m^i_t, P_t, P_{t-1}) - m^{\nu(i)}_{t-1} \right\|^2 \right). \tag{5}$$

Efficient computation The computation of the transfer function Ψ can be theoretically prohibitive, but since estimated 3D points are then close to their actual position, we reasonably know the facet on which the point actually lies, and Ψ can be approximated by the homography induced by the facet. The robust estimator handles errors on facet assignments and false matches. Since we start from a good initial estimate provided by the previous frame, the optimization is very fast and converges in a couple of iterations.

Figure 9. AUGMENTED VERSION OF THE CORRIDOR SEQUENCE. **The corresponding video sequence can be found at the following address: http://cvlab.epfl.ch/research/augm/augmented.html. Thankful to the edge information, occlusions between the real scene and the virtual objects are precisely handled. When watching the sequence, the reader can see that the virtual objects do not jitter or drift.**

5. Results

In this section we show how our technique improves tracking robustness by reducing the ambiguity of the edge information and integrating the edge and texture information.

In order to verify the improvement brought by the multiple hypotheses method, we conduced the following experiment. We considered a corridor sequence, which is a 475 frames long PAL video in which the camera is displaced through a corridor undergoing aspect changes. We know the model of the corridor, which is composed by 3100 triangles; we just used some of the edges present in this complex model.

We first tracked the sequence using the edge information only, and compared the results obtained with the single hypothesis and with the multiple hypotheses. In one hypothesis case, the camera viewpoint was estimated using Eq. 1, and in the other case using Eq. 2. The results on the corridor sequence are shown in Fig. 3 and Fig. 1. The precision of the first method is quite low, the tracking shakes and eventually fails at about frame 250. Considering multiple hypotheses as we propose makes the tracking more robust. Nevertheless the reprojected 3D model sometimes jitters. Integrating the texture information suppresses this problem.

Using the same sequence, we tested different ways of using the interest points information. As shown in Fig. 1, using our previous tracker based on reference frames and interest points only, the 3D model edges are not always reprojected at the right place, even though four reference frames were used.

As mentioned in Section 1, integrating a single hypothesis per edge point in this previous tracker does not improve the accuracy, but often even degrades it. For example, as shown in Fig. 4, the tracker gets confused by the edge of a cable lying on the floor. Estimating the camera trajectory using Eq. 5 gives much more precise results, and reduces the amount of user interaction since no reference frame was used.

In a second experiment, we tested the advantages of considering multiple hypotheses with the complex background of Fig. 2. In this sequence, a textured box was moving on a checkerboard. Obviously, the checkerboard creates numerous strong misleading edges. When the tracker uses interest points and considers only one hypothesis per edge point, the top edge of the 3D model is attracted by the strong edges on the checkerboard. Considering multiple hypotheses as we propose allows the 3D model to be correctly reprojected.

The tracking performances vary depending on which combination of information and how many features are adopted. The multiple hypothesis edge tracking alone runs at 30 frames per second on a PIV 2.6 GHz machine. When we add the feature points matching, the frame rate falls down to about 20, the interest-point matching being more demanding.

We used the method we just presented for building some Augmented Reality scenes. As shown in Fig. 9, we exploited the tracking information and the model for adding virtual plants and the ISMAR logo to the corridor floor. Thanks to the accurate model and a correct registration over the whole sequence, we were able to handle the occlusions in a proper way.

Another augmented scene is presented in Fig. 10. We track the position of the toy shuttle and we superimpose a virtual jet. We can also use the model infor-

mation to find the silhouette of the real object and to turn the background into a space like, virtual one.

6. Conclusion

This paper presents a real-time 3D tracking approach that combines edges and interest point features. These two sources of information make it both robust and precise for textured and non textured objects. Our multiple hypotheses technique, by contrast to conventional approaches, allows to use the edges information even when the features to track are much weaker than the misleading features in the background.

We tested this method on difficult scenes and showed that our method brings major improvments with respect to state-of-the-art methods. We finally showed that the stability and robustness of our system make it suitable for practical AR applications by building some test augmented scenes.

References

[1] P. Bouthemy. A maximum likelihood framework for determining moving edges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(5):499–511, May 1989.

[2] A. I. Comport, E. Marchand, and F. Chaumette. A Real-Time Tracker for Markerless Augmented Reality. In *International Symposium on Mixed and Augmented Reality*, Tokyo, Japan, September 2003.

[3] P. David, D. DeMenthon, R. Duraiswami, and H. Samet. Simultaneous pose and correspondence determination using line features. In *Conference on Computer Vision and Pattern Recognition*, volume 2, pages 424–431, Madison, WI, June 2003.

[4] D. DeCarlo and D. Metaxas. The Integration of Optical Flow and Deformable Models with Applications to Human Face Shape and Motion Estimation. In *Conference on Computer Vision and Pattern Recognition*, pages 231–238, 1996.

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

[6] Y. Genc, S. Riedel, F. Souvannavong, C. Akinlar, and N. Navab. Marker-less tracking for ar: a learning-based approach. In *International Symposium of Mixed and Augmented Reality*, pages 295–304, 2002.

[7] C. Harris. *Tracking with Rigid Objects*. MIT Press, 1992.

[8] P. Huber. *Robust Statistics*. Wiley, New York, 1981.

[9] M. Isard. and A. Blake. CONDENSATION - conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, August 1998.

[10] M. Isard and A. Blake. A smoothing filter for condensation. In *European Conference on Computer Vision*, pages 767–781, 1998.

[11] F. Jurie. Solution of the Simultaneous Pose and Correspondence Problem Using Gaussian Error Model. *Computer Vision and Image Understanding*, 73(3):357–373, 1999.

[12] F. Jurie and M. Dhome. Hyperplane approximation for template matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):996–100, july 2002.

[13] T. Lourens and R. Wurtz. Object recognition by matching symbolic edge graphs. In *Asian Conference on Computer Vision*, 1998.

[14] M. Armstrong and A. Zisserman. Robust Object Tracking. In *Proceedings of Asian Conference on Computer Vision*, pages 58–62, 1995.

[15] E. Marchand, P. Bouthemy, and F. Chaumette. A 2d-3d model-based approach to real-time visual tracking. *Image and Vision Computing*, 19(13):941–955, 2001.

[16] L. Masson, F. Jurie, and M. Dhome. Contour/texture approach for visual tracking. In *Scandinavian Conference on Image Analysis*, pages 661–668, 2003.

[17] Y. Shan, Z. Liu, and Z. Zhang. Model-Based Bundle Adjustment with Application to Face Modeling. In *International Conference on Computer Vision*, Vancouver, Canada, July 2001.

[18] G. Simon and M.-O. Berger. A two-stage robust statistical method for temporal registration from features of various type. In *International Conference on Computer Vision*, pages 261–266, Bombay, India, Jan. 1998.

[19] C. Sminchisescu and B. Triggs. Covariance Scaled Sampling for Monocular 3D Body Tracking. In *Conference on Computer Vision and Pattern Recognition*, Hawaii, 2001.

[20] L. Vacchetti, V. Lepetit, and P. Fua. Fusing Online and Offline Information for Stable 3–D Tracking in Real-Time. In *Conference on Computer Vision and Pattern Recognition*, Madison, WI, June 2003.

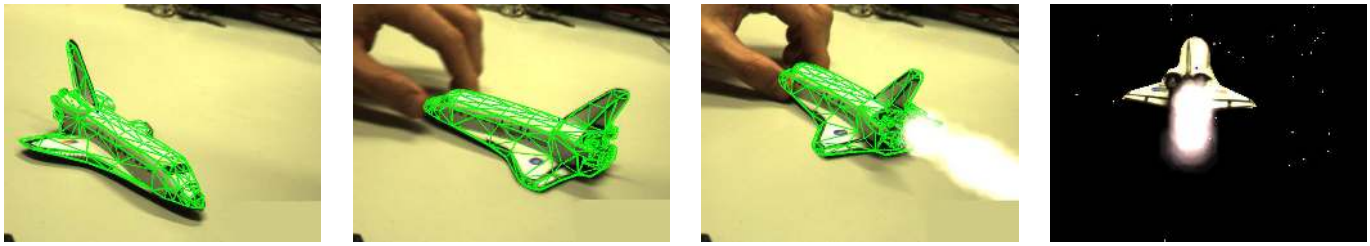


Figure 10. AUGMENTING A TOY SHUTTLE. **Tracking and augmentation of a model of spacecraft using our method. The corresponding video sequence can be found at the following address: http://cvlab.epfl.ch/research/augm/augmented.html.**