Robust Fitting of 3D CAD Models to Video Streams

Christophe Meilhac and Chahab Nastar

INRIA B.P. 105, 78153 Le Chesnay Cedex, France
Contact E-mail: Christophe.Meilhac@inria.fr
http://www-syntim.inria.fr/syntim/recherche/meilhac/

Abstract. We present a robust and accurate semi-automatic algorithm for registering and tracking a 3D geometric model in a 2D video stream. The algorithm is a generalization of the "Iterative Closest Point" technique. Each iteration is composed of two steps: computation of camera parameters, and 3D/2D vertex matching. This last step is performed by polygon fitting in an edge image. To account for false matches, we use a robust M-estimation both for camera parameter estimation and 2D feature extraction. Experimental results show that accurate registration can be obtained even with very noisy outdoor images and incomplete data. Error analysis proves that the accuracy is obtained at the pixel level.

1 Introduction

Model-based vision leads to a vast improvement in performance, at the cost of *a priori* knowledge of the 3D geometry of one (or several) objects of the scene, and maybe of some user interactivity. In this framework, we show interest in fitting a geometrical CAD-model of an object to a 2D image of the object in a complex scene (figure 4 and 5). In other words, the projection of the model into the image should match image features. In mathematical terms, the goal is to determine a 3D translation/rotation aligning (registering) the model with the image data. Our goal is to perform a model-based tracking of the object in a sequence of 2D images, thus partially reconstructing significant 3D information from the video stream. Although this is a general and important vision problem *per se*, our application is augmented reality for entertainment, i.e. mixing virtual and real-world objects while ensuring visual and physical interactions [3]. For our application domain, *accuracy* is a major issue in the 3D information recovery, in order to avoid visual aberration in the augmented sequence. On the other hand, as we deal with real-world images, the object of interest is not restricted to lie on a uniform background, yielding essentially in a noisy pre-processing stage (e.g. edge extraction). Therefore, we need a *robust* algorithm for fitting and tracking. Indeed, our method needs to be accurate and robust for performing 3D/2D registration and tracking.

Model-based registration and tracking is a rather recent issue in computer vision. Lowe [8] performs the registration of a 3D model in a segmented (edge) image, and develops a robust tracking method based on Bayesian decision theory. Wunsch and Hirzinger [10] propose a matching algorithm between an image and a polyhedral model based on the inverse perspective in order to give a 3D-3D

match. Lavallée and Szeliski [6] compute a 2D-3D match of the occluding contour and use 3D distance maps and octrees to speed up the matching process. These methods are interesting but they assume a clean segmentation of the object of interest and lack the registration accuracy that we need in our applicative framework. Other methods of model-based pose estimation have been developed with emphasis on vehicle tracking [9,4].

Our method is mostly related to Kumar and Hanson [5] who use lines as 2D primitive and present a robust estimation of pose. We use the Iterative Closest Point (ICP) algorithm [1,11] combined with robust statistics [2,12].

2 A robust 3D/2D registration technique

The main idea of our iterative algorithm is to improve a prediction of camera parameters by (i) computing the best match between a vertex point of a 3D-polyhedral CAD model and a corner point of a 2D-image data, then (ii) running a calibration algorithm based on the previous match. For outlier rejection and stabilization of the iterative process, we use robust statistics both for line estimation and for calibration.

2.1 Statement of the problem

The goal is to find the camera parameters that make the projection of the 3D-model consistent with the 2D-image, i.e. to minimize the following objective function: $\sum_{\mathbf{x}\in V} ||\sigma(\mathbf{P},\mathbf{x}) - \Pi(\mathbf{P},\mathbf{x})||$ (1). w.r.t. camera parameters $\mathbf{P}$. $\sigma(\mathbf{P},.)$ is the matching operator, $\Pi(\mathbf{P},.)$ the projection operator and V the set of vertices of the 3D polyhedral model. Thus the problems are: (1) estimating the optimal $\mathbf{P}$, (2) computing operators $\sigma(\mathbf{P},.)$ and $\Pi(\mathbf{P},.)$, and (3) being robust to noise and spatial uncertainty. Formally, the operator of projection is defined as a perspective transformation:

$$\Pi(\mathbf{P},.) : \mathbb{R}^3 \rightarrow \mathbb{R}^2 \quad \mathbf{x} = (x\ y\ z)^t \mapsto \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} u_0 + \alpha_u \frac{x'}{z'} \\ v_0 + \alpha_v \frac{y'}{z'} \end{pmatrix} \tag{2}$$

with $(x'\ y'\ z')^t = \mathbf{R}(x\ y\ z)^t + \mathbf{T}$ where $\mathbf{R}$ is the rotation and $\mathbf{T}$ the translation. α_u, α_v, (u_0, v_0) are intrinsic camera parameters.

Let us now introduce a robust version of the objective function and one that can be iteratively minimized (see section 2.2):

$$f(\mathbf{P},\mathbf{Q}) = \sum_{\mathbf{x}\in V} \rho(\sigma(\mathbf{P},\mathbf{x}) - \Pi(\mathbf{Q},\mathbf{x})) \tag{3}$$

where ρ is a M-estimator[1] (the classical least squares problem is obtained for $\rho(x) = \frac{x^2}{2}$). We have implemented many M-estimators (Welsh, Cauchy,...) ; for simplicity, we use the robust Geman-McClure M-estimator: $\rho(x) = \frac{x^2}{1+x^2}$.

[1] ρ is a generalization of classical M-estimators to $\mathbb{R}^2$: $\rho(x,y) = \rho(x) + \rho(y)$.

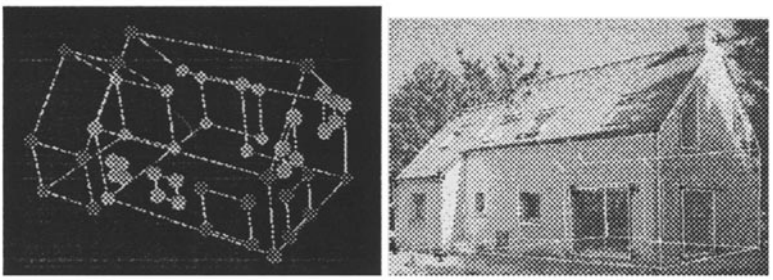
2.2 Description of the algorithm

Our purpose is to minimize the objective function $\mathbf{P} \mapsto f(\mathbf{P}, \mathbf{P})$ w.r.t. $\mathbf{P}$. The main idea is to adapt the Iterative Closest Point (ICP) algorithm [1] as follows:

Initialization: interactive initialization (see section 2.3) of $\mathbf{P}^0$.
Repeat
Matching: computation of $\sigma(\mathbf{P}^{(m)}, \mathbf{x})$ for each 3D vertex $\mathbf{x}$.
Sampling: each projected line $s \in$ Model is sampled, giving $M_s = \{(u_{i,s}, v_{i,s})\}$.
Searching: $(x_{i,s}, y_{i,s}) :=$ `ClosestPoint`$(u_{i,s}, v_{i,s})$[2] in the contour image.
Regression: estimation of the best polygon fitting $(x_{i,s}, y_{i,s})$, (see section 2.4)
Computation of corners: for all 3D vertex x, $\sigma(\mathbf{P}^{(m)}, \mathbf{x})$ = polygon corner.
Calibration: computation of $\mathbf{P}^{(m+1)}$ that minimizes $\mathbf{P} \mapsto f(\mathbf{P}, \mathbf{P}^{(m)})$. This is a classical non-linear calibration problem. To obtain a fasterconvergence we use a quasi-Newton technique. This is a robust version of [7].
Until stabilization

2.3 Interactive Initialization

For initializing the process, we have to compute a camera estimation that provides approximate projection of the model onto the image. The user can intuitively and easily match a few vertices of the 3D model with corresponding image features (figure 1). The problem is then a classical calibration problem. This step provides the visible part of the 3D model in the 2D image.



3D Model — interactive image matching

Fig. 1. Initialization is computed using interactive matching followed by calibration algorithm.

2.4 Robust polygon estimation

The 2D polygon is defined by the graph (vertices and edges) of the visible part of the 3D model. It is parameterized by the line equation of each edge. A polygon corner is defined as a line intersection. To find the polygon in edge image, we minimize the following objective funtion w.r.t. $(a_s, b_s, c_s)_s : \sum_{i,s} \rho(r_{i,s})$ (4). where $r_{i,s} = a_s x_{i,s} + b_s y_{i,s} + c_s$ is the residual of line. With no other information, finding this minimum is equivalent to finding each line minimum independently. There are some constraints that could be added to estimate a coherent polygon,

[2] Note that this operator is not differentiable.

e.g. the polygon should not be null and the intersection point is unique when more than two lines intersect (this is a graph constraint). We want to find a coherent polygon with respect to graph knowledge. A necessary condition that expresses that lines l_i, l_j, l_k intersect on a same point is $\det(l_i, l_j, l_k) = 0$ with $l = (a, b, c)$. This means that if a point is defined as intersection of $n > 2$ lines, ($n - 2$ independent constraints).

The main interest of this global minimization (instead of independent estimation of each line) is that we take into account "good" lines versus "bad" lines (partially occluded lines or "short" lines) with respect to constraints (see fig. 2).

Note that the error distribution over the polygon has to be normalized for efficient M-estimation.

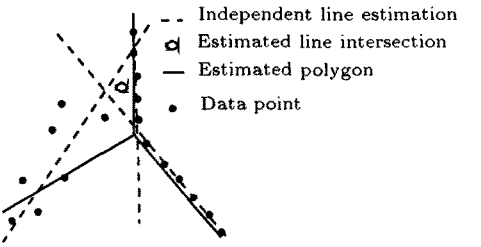


Fig. 2. Stability of intersection computation

2.5 Error analysis

There are many sources of error in the general registration problem. The model-based approach allows to compute the errors w.r.t. the model. There are many errors involving the errors of computation of the transformation: the matching error, the regression error, the projection error, the camera parameters error and the vertex projection error (see figure 3 for a geometric interpretation).

The **matching error** is the error between the image data and the 3D projection. Let $M = \bigcup_{s \in Seg} M_s$ the set of all the sample points computed from section 2.2, and where Seg is the set of all edges of the 3D-model. The error is defined as: $m_x = \|\texttt{ClosestPoint}(\mathbf{x}) - \mathbf{x}\|$, $\forall \mathbf{x} \in M$ (5). The matching error gives an idea of the presence of data. It is very sensitive to occlusion and noise.

The **regression error** is the error between image data and polygon estimation. It is defined as: $r_x = d(\texttt{ClosestPoint}(\mathbf{x}), s)$, $\forall \mathbf{x} \in \bigcup_{s \in Seg} M_s$ (6). where $d(\mathbf{x}, s)$ is the distance of point $\mathbf{x}$ to segment s. This is a pure 2D error. The regression error describes the corelation between the data and the estimated polygon.

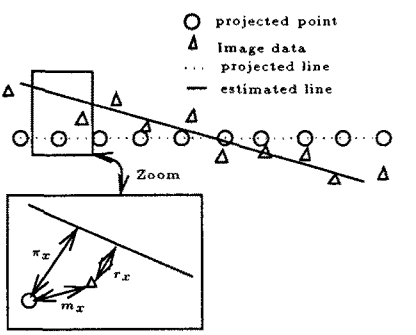


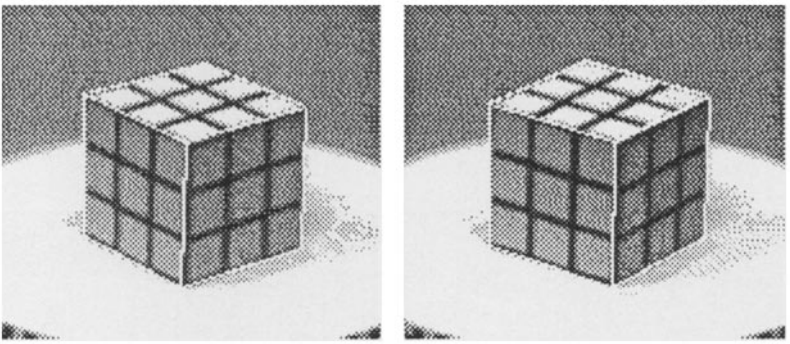
Fig. 3. Matching, regression and projection errors

The **projection error** is the error between the estimated 2D-polygon of the 3D projection of the model: $\pi_x = d(\mathbf{x}, s)$, $\forall \mathbf{x} \in M$ (7). The projection error

gives an idea of the quality of the reprojection. It is not sensitive to occlusion or noise since they have been dealt with by polygon regression.

Computing the vertex projection error v_x is more complex and has to integrate the following covariance matrix of camera parameters, polygon corners and line parameters. See [13] for a full explanation.

3 Results and experiments



(a) 1st image (b) 20th image

Fig. 4. Tracking of 3D CAD model "Rubik cube" (in wireframe).

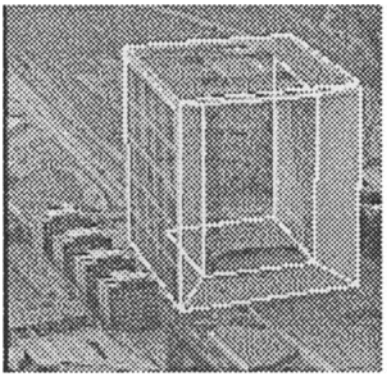
First, we test our algorithm on an easy case. The Rubik cube sequence contains twenty images of a Rubik cube in pure rotation with a fixed camera. Rotation is sampled very regularly. For all these reasons, the input data (edge image) is very clean and produces a large stability of view point.

Our algorithm gives very good results even with non robust estimation. Only the first and the last frames of our tracking are displayed on figure 4.

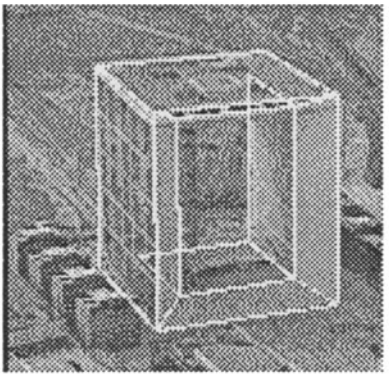
Our second data set is a video stream of 88 images of a complex aerial view of the famous "Arche de la Défense" monument in Paris. This is a real-world application where the object is still and the camera has an unstable trajectory due to the helicopter. This is a complex example of tracking because the input data is very noisy (low resolution, noise on video, missing data) and because the camera motion is not smooth.

Figure 5 displays the tracking result. Our algorithm with full robust estimation is able to track the arch through the whole sequence. The algorithm provides a good reprojection of the 3D model into the video stream. Note the wide variations of point of view (rotation and translation).

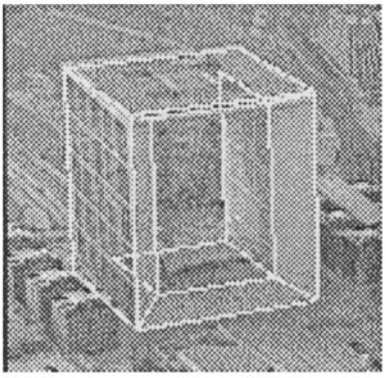
We now detail the performance of the algorithm on the complex arch sequence. Table 1 shows typical numerical results of errors in the 3D/2D registration process. We sample each line at one point per pixel. Estimated segments have an average length of 40 pixels (maximum length is 70 pixels and minimum length is 8 pixels). In this sequence, each corner moves over 40 pixels.



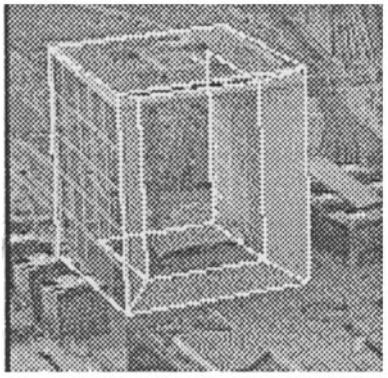
(a) 1st image



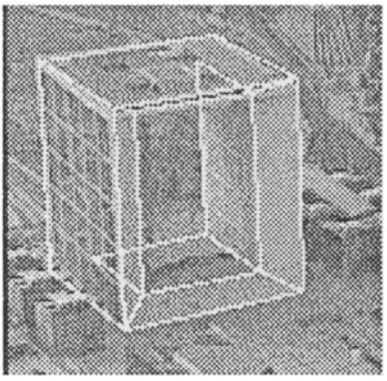
(b) 15th image



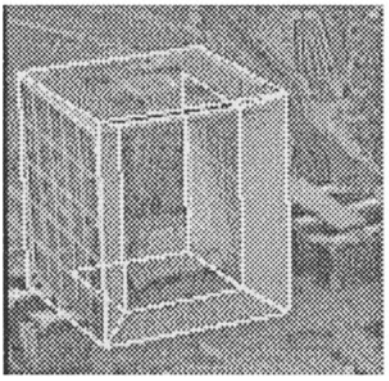
(c) 30th image



(d) 45th image



(e) 60th image



(f) 75th image

Fig. 5. Tracking of the arch. In wireframe, reprojection of the 3D model. Note the wide pose variations, and the correct estimation of occluded edges of the arch.

error type	v_x	m_x (equation (2.5))	r_x (equation (2.5))	π_x (equation (2.5))
mean error	0.5	1.23	0.63	0.16
deviation	0.2	1.23	0.75	0.14
max error	0.9	8.64	6.66	0.5
min error	0.2	0.05	7.6e-3	2e-2
# of points	14	892	892	892

camera parameter	a_x	a_y	a_z	t_x	t_y	t_z	α_u	α_v	u_0	v_0
value v	153	140	-0.71	-0.18	-0.21	-179	-90	909	536	594
deviation σ	1.2e-3	5.1e-4	3.5e-4	1.6	2.1	4.8	2.8	3.3	0.9	1.3

Table 1. Error analysis of projection and of parameters of camera. $(a_x\ a_y\ a_z) = \mathbf{r}\tan\frac{\theta}{2}$ where $(\mathbf{r}, \theta)$ is the rotation, $(t_x\ t_y\ t_z)$ is the translation, $\alpha_u, \alpha_v, u_0, v_0$ the intrinsic camera parameters (see equation (2)).

The vertex location is obtained with an accuracy of 0.5 pixel (table 1, v_x). This is the main error we wish to minimize, and the obtained precision is satisfactory. The regression error is about 0.6 pixel (table 1, regression error r_x). It gives an idea of a presence of polygon in edge image. It is sensitive to corrupted data. The matching error is about 1.2 pixels (table 1, matching error m_x), and basically reflects missing image data (occlusion etc.). Finally the projection error is 0.16 pixels and reflect the 3D/2D coherence of the polygon. Our robust approach allows to deal with this rather large error. The estimation of camera parameters is very good for all parameters.

We observe that through the video stream, each corner moves more than 40 pixels, and still, our estimation is accurate. On Figure 6 we display the distribution of vertex projection error. Note that it is less than 1 pixel.

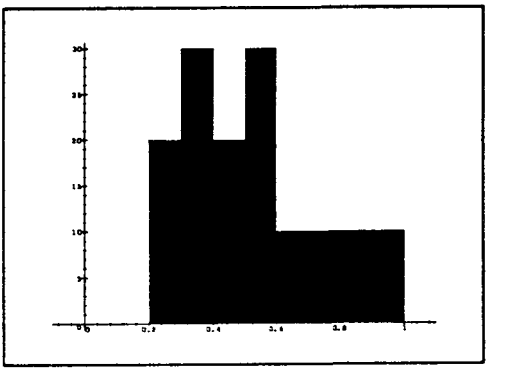


Fig. 6. Distribution of the vertex projection error. Note that the error is always less than 1 pixel.

4 Conclusion and future work

We presented an algorithm for robust and accurate registration and tracking of a 3D-model in video streams. The algorithm uses information about edge location. It is based on an ICP minimization technique. All estimations (2D-extraction, camera parameter computation) are robust. We prove experimentally the robustness of the approach to very noisy data and important occlusion.

The experimental results are very encouraging. We are currently improving the method by performing temporal stabilization, e.g. Kalman filtering, post-computation regularization. The 2D-feature extraction technique can also be extended to more generic 2D-models like ellipses, or any parametric description of 2D-curves.

Finally, the subpixel error obtained by our method should allow a straightforward application to mixing virtual objects into video streams while ensuring 3D coherence.

References

1. P. J. Besl and N. D. McKay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, February 1992.
2. P.J. Huber. *Robust Statistics.* Wiley series in probability and mathematical statistics, 1981.
3. P. Jancène, C. Meilhac, F. Neyret, X. Provot, J.-P. Tarel, J.-M. Vezien, and A. Verroust. RES: computing the interactions between real and virtual objects in video sequences. In *the second IEEE workshop on networked realities*, Boston Mass. (USA), October 26-28 1995.
4. H. Kollnig and H.H. Nagel. 3D pose estimation by fitting image gradients directly to polyhedral models. In *International Conference on Computer Vision*, pages 569–574, 1995.
5. R. Kumar and A. R. Hanson. Robust methods for estimating pose and sensitivity analysis. *Computer Vision, Graphics, and Image Processing: Image Understanding*, 60(3):313–342, November 1994.
6. S. Lavallée and R. Szelisky. Recovering the position and orientation of free-form objects from image contours using 3D distance maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(4):378–390, April 1995.
7. R.K. Lenz and R.Y. Tsai. Techniques for calibration of the scale factor and image center for high accuracy 3-D machine vision metrology. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(5):713–720, 1988.
8. D.G. Lowe. Fitting parameterized three-dimensional models to images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(5):441–450, May 1991.
9. A.D. Worrall, G.D. Sullivan, and K.D. Baker. Pose Refinement of Active Models Using Forces in 3D. In *European Conference on Computer Vision*, pages A:341–350, 1994.
10. P. Wunsh and G. Hirzinger. Registration of CAD-Models to Images by Iterative Inverse Perspective Matching. In *International Conference on Pattern Recognition*, 1996.
11. Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, 13(2):119–152, 1994.
12. Z. Zhang. Parameter estimation techniques: A tutorial with application to conic fitting. Technical Report 2676, INRIA, October 1995.
13. Z. Zhang. Determining the epipolar geometry and its uncertainty: A review. *International Journal of Computer Vision*, 1997.