Model-Based Object Tracking in Cluttered Scenes with Occlusions

Frédéric Jurie

Model-Based Object Tracking in Cluttered Scenes with Occlusions

Frederic Jurie
LASMEA - CNRS UMR 6602
Université Blaise-Pascal
Campus Scientifique des Cezeaux
63177 Aubière
France
email : Frederic.Jurie@lasmea.univ-bpclermont.fr

Abstract

In this paper, we propose an efficient method for tracking 3D modelled objects in cluttered scenes. Rather than tracking objects in the image, our approach relies on the object recognition aspect of tracking. Candidate matches between image and model features define volumes in the space of transformations. The volumes of the pose space satisfying the maximum number of correspondences are those that best align the model with the image. Object motion defines a trajectory in the pose space.

We give some results showing that the presented method allows to track objects even when they are totally occluded for a short while, without supposing any motion model and with a low computational cost (below 200 ms per frame on a basic workstation). Furthermore, this algorithm can also be used to initialize the tracking.

1 Introduction

One of the outstanding problems in visual perception for intelligent robots is the development of systems capable of tracking 3D objects in monocular sequences of image.

In this paper, 3D objects as well as images are supposed to be modelled by means of features. Then tracking objects involves to be able to find correspondences between model and image features from frame to frame. In realistic condition, the scene is cluttered and objects can be wholly occluded. A typical sequence is shown Fig.3.

The most common approach consists in computing the object pose from a previous images and in predicting the feature positions in the image by using a motion model. By using this approach, the matched image features are spatially close to the prediction but no one can be sure they are coherent regard to the object structure. This is specially true when the scene is cluttered and when objects can be occluded. The system may drift from the object and may track another part of the scene.

The approach developed here is different in the sense it is based on the track of the pose that best align the model with the image. This turns the problem of object tracking in a problem of dynamic object recognition. Candidate matches between image and model features define volumes in the space of transformations. The volumes of the pose space satisfying the maximum number of correspondences are those that best align the model with the image. A model feature is said to be aligned on an image one when they satisfy a geometric error model. Generally, a bounded error model is used. After discussing related papers, we will show in the section 3 how it is possible to efficiently find this best 3D pose.

One contribution of this paper is to use a probabilistic error model instead of the bounded one. It greatly improves the efficiency of the pose search algorithm. This error model and the way to use it will be described in the section 4. The experiments and results are illustrated in the section 5.

2 Previous works

3D object tracking with model has been intensively studied in the past years. Due to the lack of space we only cite most notable works. More references can be found in a paper of Koller, Daniilidis and Nagel (1993) [11].

Several techniques have been proposed in order to make the matching between features more reliable. Deriche and Faugeras (1990) [7] propose to measure the distance of line attributes by using the Mahanalobis distance. Crowley, Stelmaszyk, Skordas and Puget (1992) [5] combine this measure with a velocity constant motion model to estimate the 3D structure of a scene from 2D tracking.

Manjunath, Shekhar and Chellappa (1996) [12] describe an image feature detector using Gabor filters. The authors affirm that these features are robust and can be easily tracked in an image sequence.

Koller, Daniilidis and Nagel (1993) [11] propose a very efficient algorithm to track moving vehicles. They take into account the shadow edges of the vehicle by including an illumination model. The vehicle is modelled by 12 parameters enabling the instantiation of different vehicles.

Some other approaches concentrate only on the motion estimation. Zhang and Faugeras (1992) [15] established a constant angular velocity and constant translational acceleration-motion model.

3 Tracking objects in the pose space

As explained in the introduction, object tracking can be performed by tracking object poses in the 3D pose space. It is to say that there is a geometric transformation mapping model features onto their corresponding ones in the image. The problem is to identify a correspondence I that pairs model and image features. Each correspondence I specifies some transformation which maps one model feature to a corresponding image feature, given an error model.

Originally, the quality of the recognition has been measured by counting the size of the correspondence I. In that scheme, the aim is to find the set of transformations maximizing $|I|$.

The generalized Hough transform has been one of the first methods in that class. Poses are generally histogramed and each pose is represented by a single point in the pose space rather than considering the exact set of transformations. Error bounds are often taken into account by using overlapping bins.

Those methods offer the advantage of avoiding exponential search. However, the quantized pose space is generally enormous (R^8 for a scaled orthographic projection).

Such methods suffer from other severe criticisms. They concern the presence of false peaks in the parameter space (which can be very high with noisy data), occlusion and tessellation effects (see Grimson and Huttenlocher (1990) [9]).

Several techniques have been proposed to reduce the search space size. We particularly note the *coarse-to-fine clustering* (see Stockman, Kopstein and Bennet (1982) [13] and the *recursive histograming* (Thompsom and Mundy (1987) [14]). With these techniques, the transformation space is recursively divided. Starting with a set including all transformations, the current space is divided into several subspaces. The subspace maximizing the number of correspondences is alternatively kept and divided.

Unfortunately these techniques present several problems. First, the error model is not respected, and above all, the quality of the match is evaluated by counting the size of the correspondences set. Huttenlocher and Cass (1992) [10], (as well as Gavila and Groen (1992) [8]) argued that this measure, although widely used, often greatly overestimates the quality of a match. Instead, they proposed the *maximum bipartite graph* or more simply to count the number of distinct features. If U distinct model features are in correspondence with V image features then the quality of the hypothesis is $min(U, V)$. In our experiments, we use a similar measure of quality (see section 4.5). Such techniques require keeping track which features are accounted for by a given set of feature pairs.

Cass [4, 3] was the first one to implement this strategy by introducing the CPS (Critical Point Sampling). It consists in a sweep of the arrangement generated by the feasible set given by all the correspondences between model and image features, in a polynomial time.

Breuel (1992) [2] combines both advantages of recursive search : respect of the error model and keeping track of pairing. By deriving Baird research [1], Breuel demonstrates that when the transformation is affine, convex polygonal errors bounds give rise to convex polyhedral sets. From these remarks, Breuel proposes the RAST algorithm under the conditions of 2D translations, rotations and scaling. The algorithm starts out with a box in the transformation space that contains all the transformations. Then the current box is subdivided into smaller ones. The same process is repeated recursively, by choosing the half-space giving the best quality of matching.

3.1 Pose search

However, the Breuel's algorithm is restricted to 2D matching. From what we know, DeMenthon (1993) [6] is the only one who resumed this method and applied it to 3D problems, under orthographic scaled projection. We use a similar strategy, but with a probabilist error-model instead of a bounded one.

(Vectors are written down with bold fonts.)

A full perspective projection can be represented by $\mathbf{P}$ such that :

$$\mathbf{P} = \begin{pmatrix} \mathbf{i} & tx \\ \mathbf{j} & ty \\ \mathbf{k} & tz \\ (0,0,0) & 1 \end{pmatrix} = \begin{pmatrix} \mathbf{P_1} \\ \mathbf{P_2} \\ \mathbf{P_3} \\ \mathbf{P_4} \end{pmatrix}$$

If M_0 is the origin of the object reference, and $\mathbf{M}_i$ the i-th model point projected onto $\mathbf{p}_i = (x_i, y_i)$ then the projection can be written (see [6]) :

$$\begin{cases} \mathbf{M_0 M_i P_1} \frac{f}{t_z} = x_i(1+\epsilon_i) \\ \mathbf{M_0 M_i P_2} \frac{f}{t_z} = y_i(1+\epsilon_i) \end{cases}$$

if $\epsilon_i = \mathbf{M_0 M_i}(\mathbf{P_3}/t_z - 1)$, and f, denotes the camera focal distance.

Weak perspective approximation assumes that objects lies in a plane parallel to the image plane passing

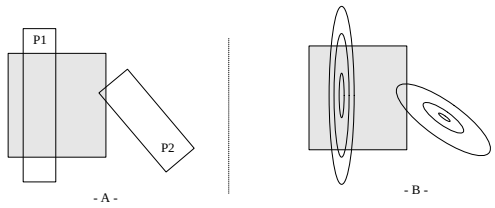

Figure 1: Bounded Error Model , Probabilistic Error Model

through the origin of the object frame. This is equivalent to the approximation : $\epsilon_i = 0$.

The perspective projection is therefore approximated with an affine transformation. The methodology described before is directly re-usable : a correspondence between a model segment and an image segment under the bounded error constraint produces a polyhedron in the 8-D transformation space. Then a recursive search can be applied.

However, DeMenthon [6] reports that matching 30 model features (corners in his case) with 200 images features requires several hours of computation. We observed that these poor results are mainly due to the bounded error model.

This is illustrated in Fig. 1. In part A, both polyhedra P1 and P2 intersect the box. But intuitively, the best transform is more likely to be in P1 rather that in P2 because its intersection with the box is larger. With the bounded error model, the two polyhedra have the same weight in the evaluation of the box.

That is why we propose to substitute a probability function for the bounded error, the probability of a match subject to a box of transformations, defined in section 4.1. In that case, the evaluation of a box will be more complicated than simply computing intersecting polyhedrons, as with the bounded error model.

3.2 Searching the space of feasible 3D affine transformations

The algorithm starts out with a box containing all possible transformations. Recursive subdivisions are then performed, alternating the axis used to divide the box. This process can be seen as a tree search. The root node corresponds to the entire subspace. Each node represents a subspace. Leaves are the smallest regions taken into account.

Breuel (1992) [2] proposed to explore the "best" branch (dividing on each level the box with the best evaluation) and then to backtrack the search, looking for other possible solutions. During the backtracking stage, remaining boxes are subdivided if their score are higher than the best score obtained on a "leaf" box.

The maximal number of boxes explored and consequently the run time cannot be guaranteed. In the worst case the whole space has to be explored.

That's why we recommend a N-search algorithm. N branches are explored at the same time and no backtracking is required. The maximum number of boxes evaluated is below Nh where h denotes the number of levels. Furthermore, N directly represents the efficacy of the algorithm.

3.3 From scaled orthographic projection to full perspective projection

By searching the best scaled orthographic projection, we look for $\mathbf{I}$ and $\mathbf{J}$ (with $\mathbf{I} = \mathbf{i}(f/t_z)$, $\mathbf{J} = \mathbf{j}(f/t_z)$) such that :

$$\begin{cases} \mathbf{M_0 M_i I} = x_i \\ \mathbf{M_0 M_i J} = y_i \end{cases}$$

for a maximum number of features. In fact, the exact perspective projection is :

$$\begin{cases} \mathbf{M_0 M_i P_1} f/t_z) = x_i(1+\epsilon_i) \\ \mathbf{M_0 M_i P_2} f/t_z) = y_i(1+\epsilon_i) \end{cases}$$

That is to say we have to calculate ϵ_i previously omitted, given by

$$\epsilon_i = \frac{\mathbf{M_0 M_i} \cdot \mathbf{P_3}}{t_z}$$

$\mathbf{M_0 M_i}$, the coordinates of point i in the object reference, are known. We have to determine t_z and $\mathbf{k}$. From that, we deduce $t_z = f/\|\mathbf{I}\| = f/\|\mathbf{J}\|$, and we have :

$$\begin{cases} (\mathbf{i}, tx) = \mathbf{P_1} \frac{t_z}{f} \\ (\mathbf{j}, ty) = \mathbf{P_2} \frac{t_z}{f} \end{cases}$$

This permits to compute $\mathbf{k} = \mathbf{i} \times \mathbf{j}$. and finally ϵ_i.

When ϵ_i is obtained, we modify the coordinates of image features, by multiplying them by $(1+\epsilon_i)$. Several iterations are necessary for the first image of a sequence to recover the full perspective. During dynamic recognition, corrections ϵ_i are periodically refined, when new images occur.

4 Probabilistic error model

In this section, we try to answer the following questions :

1. What is the probability of a given image segment to be matched with a given model segment, subject to a given affine transformation ?
2. What is the probability of a given image segment to be matched with a given model segment, subject to a given box of possible affine transformations ? (We call "box" a rectangular volume of the pose space.)
3. What is the probability of an object to be matched given a set of correspondences ?

Transformations are first supposed to be scaled orthographic projections. We will see later how it can be extended to perspective transformations.

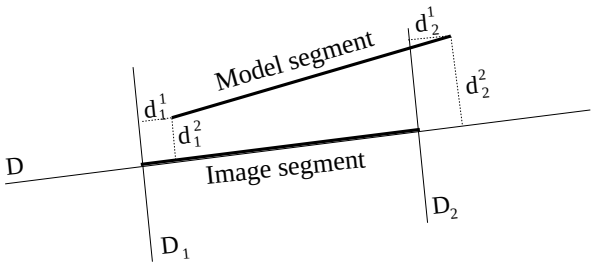

Figure 2: Error model for line segments

4.1 Probability of two segments to be matched

Segments are represented by the positions of their extremities. A segment is considered as being matched with (or in correspondence with) an other segment when their extremities are matched. Rather than modelling this probability of correspondence by means of Euclidean distances between correspondent extremities, we use a measure that is less sensible to segment fragmentation.

D is the support line of the segment (see Fig. 2 for details). D1 (and respectively D2) is orthogonal to D and cuts D at the first (respectively second) extremity of the segment. These lines can be defined by the equation $\mathbf{n_i^j} \cdot \mathbf{p} = 0, (i,j) \in \{1 \ldots 2\}^2$, where $\mathbf{n_i^j}$ is the unit vector orthogonal to it, and where $\mathbf{p} = (x, y, 1)$ denotes a point of the image plane.

Let $\mathbf{d} = (d_1^1, d_1^2, d_2^1, d_2^2)$ the vector made of the four distances between model segment extremities and the four lines. As the distances $\mathbf{d_i^j}$ take discrete values, we can define the conditional probability of a segment m of being matched with a segment s knowing $\mathbf{d}$ as the function $P(m \rightarrow s|d) = f(D(m,s))$ (f has been learnt from a set of examples), with $D(m,s) = \sum_{i=1}^{2}\sum_{j=1}^{2}(\frac{d_i^{j\,2}}{2(\sigma_i^j)^2})$. σ_i^j are constant values estimated from a set of examples.

In the sequel, we will only study $D(m,s)$ knowing that the conditional probability $P(m \rightarrow s|d)$ is directly derived from it.

The distance from point $\mathbf{p}$ to line D_i^j is $d_i^j = |\mathbf{n}_i^j \cdot \mathbf{p}|$ If p_i, $(i \in \{1,2\})$ denotes the model segment extremities, then $D(m,s)$ can be written :

$$D(m,s) = \sum_{i=1}^{2}\sum_{j=1}^{2} \frac{\left|\mathbf{n_i^j} \cdot \mathbf{p}_i\right|^2}{2(\sigma_i^j)^2}$$

4.2 Segments distance given a scaled orthographic projection

Let $\mathbf{t} = (I_x, I_y, I_z, T_x, J_x, J_y, J_z, T_y)^t$ be a scaled orthographic projection. This transformation can also be written with the following homogeneous matrix :

$$\mathbf{T}(t) = \begin{pmatrix} \mathbf{I} & tx \\ \mathbf{J} & ty \\ (0,0,0) & 1 \end{pmatrix}$$

With $\mathbf{I} = (I_x, I_y, I_z)$ and $\mathbf{J} = (J_x, J_y, J_z)$.

Let us denote $\mathbf{P} = (X, Y, Z, 1)^t$ a point in the 3D object reference, $\mathbf{s} = (p_1, p_2)$ an image segment and $\mathbf{m} = (P_1, P_2)$ the corresponding model segment. $\mathbf{m}$ is mapped (in the image plane) on segments $(\mathbf{T(t)} \cdot \mathbf{P_1}, \mathbf{T(t)} \cdot \mathbf{P_2})$ by the projection $\mathbf{T(t)}$, where points $\mathbf{P}_i$ are segment extremities.

The product $\mathbf{n_i^j} \cdot \mathbf{T(t)} \cdot \mathbf{P}_i$ can be re-written as follows : $\mathbf{n_i^j} \cdot \mathbf{T(t)} \cdot \mathbf{P_i} = (n_{ix}^j, n_{iy}^j, 1)\mathbf{T(t)}(X,Y,Z,1)^t = (n_{ix}^j X, n_{ix}^j Y, n_{ix}^j Z, n_{ix}^j, n_{iy}^j X. n_{iy}^j Y, n_{iy}^j Z, n_{iy}^j)\mathbf{t} = \mathbf{hp_i^j} \cdot \mathbf{t}$

This product can be geometrically interpreted as the distance from the transformation $\mathbf{t}$ to the hyper plane $\mathbf{hp_i^j}$, in the scaled orthographic pose space.

Coefficients of $\mathbf{hp_i^j}$ are functions of the 3D coordinates of the segment and of the coordinates of the corresponding 2D image segment. The distance between an image segment and a model segment subject to a given transformation is the sum of the squared distances from this transformation to these four hyper planes.

Then finally, with these notations, the distance from $\mathbf{m}$ to $\mathbf{s}$ given the transformation $\mathbf{T(t)}$ is :

$$D(m,s) = \sum_{i=1}^{2}\sum_{j=1}^{2} \frac{|\mathbf{hp_i^j} \cdot \mathbf{t}|^2}{2(\sigma_i^j)^2}$$

4.3 Segments distance subject to a given box of transformations

We define the distance from an image segment to a model segment subject to a box of transformation as $D(m,s,Box) = min_{t \in Box} D(m,s,t)$

The computation of this distance requires the minimization of a quadratic function, subject to linear inequality constraints (the box of transformations).

One common approach uses the *Lagrange Multipliers* combined with active set methods. But active sets method are highly time consuming.

Accordingly, we propose a more efficient algorithm computing an approximation of this minimal distance.

Let V the affine manifold generated by the 4 hyperplanes produced by a pair of matched segments. We first compute the position of the point $\mathbf{t}_0 \in V$ such that $\forall \mathbf{t} \in V, d(c, \mathbf{t}_0) \leq d(\mathbf{c}, \mathbf{t})$ (where $\mathbf{c}$ is the centre of the box and $d()$ the Euclidean distance), by means of the Lagrangian method. If $\mathbf{t}_0$ is not in the box then $\mathbf{t}_{min}$ is taken as the intersection of the line $(\mathbf{c}, \mathbf{t}_0)$ with the convex hull of the box ($\mathbf{t}_{min} = \mathbf{t}_0$ if $\mathbf{t}_0$ is included in the box). If $\mathbf{t}_{min}$ is not in the box, its position is iteratively adjusted to minimize the distance $D(\mathbf{m}, \mathbf{s}, \mathbf{t})$.

For that purpose, we use the *alternating variable strategy* in which at iteration $k (k \in [1..8])$ only variable t_k is changed in attempt to reduce the objective function value. In our experiments, we measure that this algorithm is more than 100 times faster than the active set method.

4.4 Probability of object match

We suppose that the probability of having the model M in an image subject to a transformation (or a box of transformations) T only depends on individual probabilities of model segments to be matched with image segments. If the model size is $\mathcal{M}$ (number of features of M), there are $2^{\mathcal{M}}$ possible configurations denoted γ; this makes the estimation of $P(M|T)$ difficult. The fact that a model segment m is matched is denoted $m \rightarrow$ (respectively $m \not\rightarrow$ if the segment is not matched). Configurations can be grouped according to their number of matches. The set $E^k, k \leq \mathcal{M}$ includes configurations that match k model segments. We denote $E^k = \bigcup_{j=1}^{j<\subset_{\mathcal{M}}^k} \gamma_j^k$, and $\Gamma = \bigcup_{i=1}^{i\leq\mathcal{M}} E^i$, the set of all possible exhaustive and mutually exclusive configurations.

$$\begin{aligned} P(M|T) &= \sum_{\gamma\in\Gamma} P(M|T,\gamma)P(\gamma|T) \\ &= \sum_{k=1}^{k\leq\mathcal{M}} \sum_{j=1}^{j\leq\subset_{\mathcal{M}}^k} P(M|T,\gamma_j^k)P(\gamma_j^k|T) \end{aligned}$$

We can simplify this formula, as M and T are conditionally independent given γ :

$$P(M|T) = \sum_{k=1}^{k\leq\mathcal{M}} \sum_{j=1}^{j\leq\subset_{\mathcal{M}}^k} P(M|\gamma_j^k)P(\gamma_j^k|T)$$

The size of Γ is generally large, and $P(M|\gamma)$ would be difficult to learn. We simplify this expression considering that the most significant parameter for computing this probability is the number of image features matched.

That is to say :

$$\forall k \in \{1 \ldots \mathcal{M}\}, \forall i \in [1 \ldots \subset_{\mathcal{M}}^k]$$

$$P(M|\gamma_i^k) = P(M|\bigcup_{i=1}^{i\leq\subset_{\mathcal{M}}^k} \gamma_i^k) = P(M|E^k)$$

The probability $P(M|T)$ can therefore be written : $P(M|T) = \sum_{k=1}^{k\leq m} P(M|E^k) \sum_{j=1}^{j\leq\subset_{\mathcal{M}}^k} P(\gamma_j^k|T)$. $P(M|E^k)$ is the probability of having model M knowing that k of its features are matched. It has been learned from our image basis.

The computation of $P(E_i|T) = \sum_{j=1}^{j\leq\subset_{\mathcal{M}}^k} P(\gamma_j^k|T)$ is more tedious. Event E_i is the union of $\subset_m^k$ different combinations. $P(\gamma_j^k|T)$ is the probability of that combination, given a set of correspondences. This probability can be written $P(\gamma_j^k|T) = \prod_{i=1}^{i\leq\mathcal{M}} P(m_i \overset{b(i)}{\rightarrow})$, where $b(i)$ is a Boolean variable, meaning the model segment m_i should be (or not) matched in that combination $(m \overset{0}{\rightarrow}= m \not\rightarrow, m \overset{1}{\rightarrow}= m \rightarrow)$. If we suppose that $m_i \overset{b(i)}{\rightarrow}, i \in \{1, \ldots, \mathcal{M}\}$ are independent events, we have $P(\gamma_j^k|T) = \prod_{i=1}^{i\leq\mathcal{M}} P(m_i \overset{b(i)}{\rightarrow}|T)$.

In that case $P(E_i|T)$ is a sum of products taking a long time to be computed. We propose to use an approximation of that sum, by only considering its maximal terms. As each term is a product of positive values, the maximal product is obtained with maximal values. This simplification is very easy to implement : first we sort probabilities $P(m_i \overset{b(i)}{\rightarrow}|T)$, and affect the k highest probabilities to the k segments that should be matched. Other probabilities are affected to unmatched segments.

If we suppose that I is an index function such that

$$\begin{aligned} &\forall (k,l) \in \{0, \ldots, \mathcal{M}\}^2 \\ &P(m_{I(l)} \rightarrow |T) > P(m_{I(k)} \rightarrow |T) \Rightarrow k < l \end{aligned}$$

Then

$$\begin{aligned} P(E_i|T) &= P(m_0 \overset{b(0)}{\rightarrow}, \ldots, m_i \overset{b(i)}{\rightarrow}, \ldots, m_{\mathcal{M}} \overset{b(\mathcal{M})}{\rightarrow}) \\ &>= \prod_{j=1}^{j<=i} P(m_{I(j)} \rightarrow) \prod_{j=i+1}^{j<=\mathcal{M}} (1 - P(m_{I(j)} \not\rightarrow)) \end{aligned}$$

4.5 Distinct correspondences

A model segment may be associated with more than one scene segment ; conversely a scene segment may be associated with more than one model segment. The risk is an overestimation of the quality of a match. This problem has been treated by several authors like Gavila and Groen (1992) [8] or Huttenlocher and Cass (1992) [10] in case of bounded error models.

In our case, the use of probabilities allows a straightforward solution to this problem. If we note $P(m \rightarrow)$ the probability of an image segment to be matched, and $P(m \rightarrow s)$ the probability that "model segment m is matched with image segment s", then we define $P(m \rightarrow) = P(\cup_{j=1}^{j\leq\mathcal{S}} m \rightarrow s_j)$ where $\mathcal{S}$ is the number of image segments.

5 Experiments and Results

5.1 Tracking experiments

Fig. 3 presents some of the results obtained with a hundred images of the "mouse sequence". In that se-

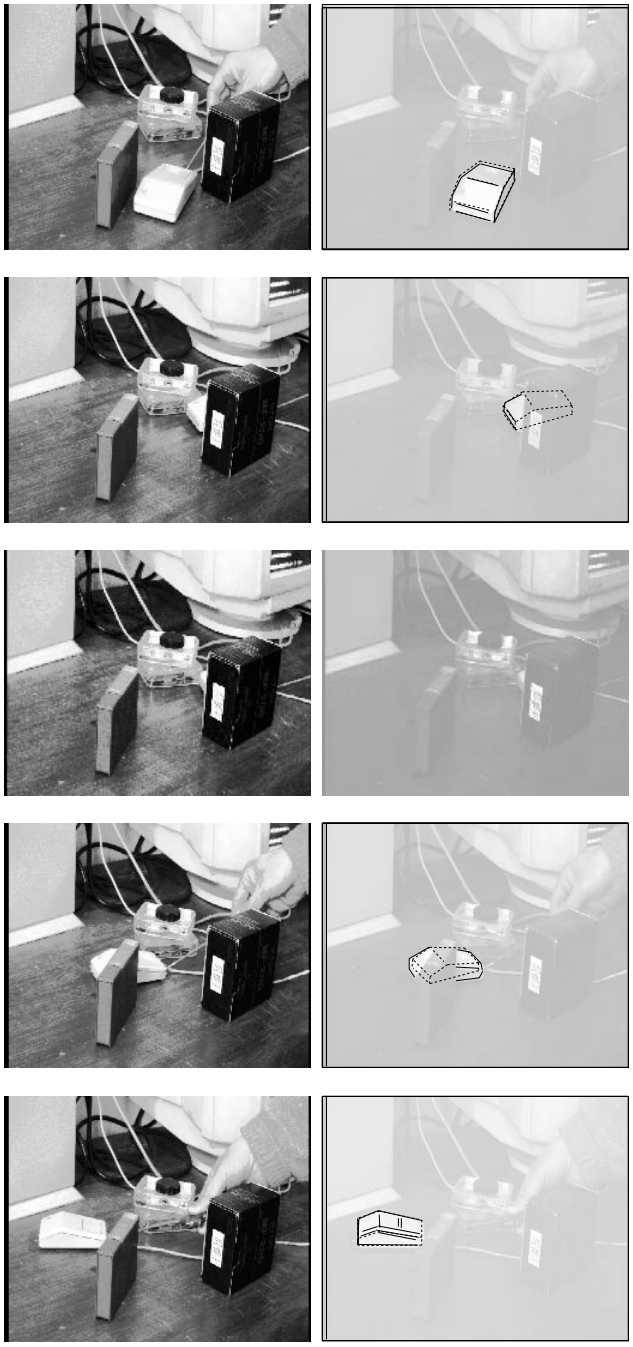

Figure 3: Model (dotted lines) and correspondences obtained on the "mouse" sequence

quence, the mouse moves on the right until it is completely hided behind the box, then starts to cross the screen in the left direction. The motion is then unpredictable.

To guarantee the achievement of the right solution, the constant N (defined in the section 3.2) has to be set to about 5.

No motion analysis is performed. The 3D pose space is searched in a box centred on the previous pose. The size of the box depends on the quality of the previous match.

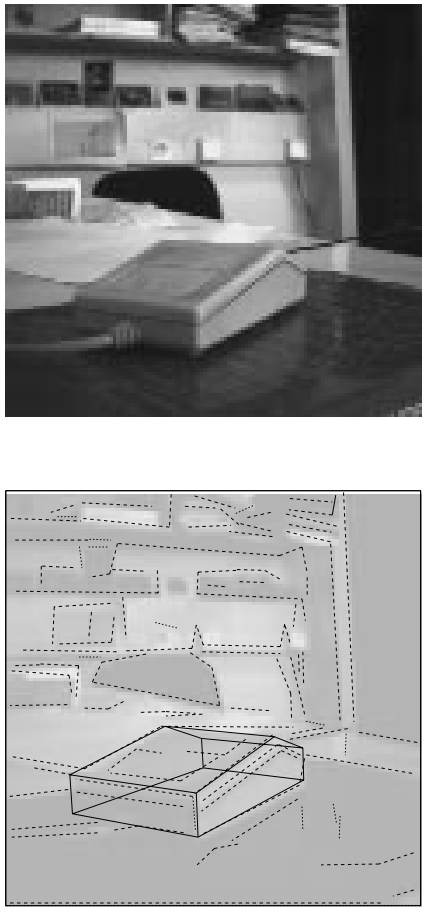

Figure 4: Recognition stage : grey level image, line segments extracted (dotted lines) and pose computed (solid lines).

The first image is processed several times (4 times in our case) in order to compute the perspective pose. The processing time is about 200 ms on our HP-700 workstation (this doesn't include the line segment extraction processing time).

The pose found on the fourth image presented is not very accurate. This is because the mouse rotates while being hidden by the box. This movement is difficult to detect because of occlusions. Furthermore, we do not use the fact the motion is planar in that sequence.

5.2 Recognition experiments

At the beginning of a sequence, the 3D pose is unknown. This algorithm can be used to compute this 3D initial pose. This is possible, because it is in fact a recognition algorithm. Rather than searching a small box of the pose space, the full 3D pose space is searched. It takes from 10 to 30 seconds to find the pose, depending on the complexity of the image.

This recognition stage is illustrated in the Fig. 4.

Conclusion

Our opinion is that the presented recognition algorithm can be used to perform a robust tracking. Correspondences are computed in the pose space rather than in the image space. By that way, the spatial coherence of the matched features is guaranteed.

As it involves a recognition scheme, the tracked object can disappear for a while, and the object motion doesn't have to be known.

References

[1] H.S Baird. Model-based image matching using location. In *MIT Press, Cambridge, MA*, 1985.

[2] T.M. Breuel. Fast recognition using adaptive subdivisions of transformation space. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 445–451, Champain, Illinois, 1992.

[3] T.A. Cass. Feature matching for object localization in the presence of uncertainty. In *Proc. IEEE International Conference on Computer vision*, pages 360–364, Osaka, Japan, 1990.

[4] T.A. Cass. Polynomial-time object recognition in the presence of clutter, occlusions, and uncertainty. In *Proc. European Conference on Computer Vision*, pages 834–842, Santa Margherita Ligure, Italy, 1992.

[5] J.L. Crowley, P. Stelmaszyk, T. Skordas, and P. Puget. Measurement and integration of 3-d structures by tracking edge lines. *International Journal of Computer Vision*, 8:29–52, 1992.

[6] D. DeMenthon. De la vision artificielle a la realite synthetique : systeme d'interaction avec un ordinateur utilisant l'analyse d'images video. In *These Univ. Grenoble I - Laboratoire TIMC/IMAG*, 1993.

[7] R. Deriche and O. Faugeras. Tracking line segments. *Image and Vision Computing*, 8(4):261–270, 1990.

[8] D.M. Gavila and F.C.A. Groen. 3d object recognition from 2d images using geometric hashing. *Pattern Recognition Letters*, 13:263–278, 1992.

[9] W.E.L. Grimson and D.P. Huttenlocher. On the sensitivity of the hough transform for object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(3):255–274, March 1990.

[10] D.P. Huttenlocher and T.A. Cass. Measuring the quality of hypotheses in model-based recognition. In *Proc. European Conference on Computer Vision*, pages 773–777, Santa Margherita Ligure, Italy, 1992.

[11] D. Koller, K. Daniilidis, and H.H. Nagel. Model-based object tracking in monocular image sequences of road traffic scene. *International Journal of Computer Vision*, 10(3):257–281, 1993.

[12] B.S. Manjunath, C. Shekhar, and R. Chellappa. A new approach to image feature detection with applications. *Pattern Recognition*, 29(4):627–640, 1996.

[13] G. Stockman, S. Kopstein, and S. Bennet. Matching images to model for registration and object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 4(3):229–241, 1982.

[14] D.W. Thompsom and J.L Mundy. Three dimentional model matching from an unconstrained viewpoint. In *Proc. Robotics and Automation*, pages 208–220, Raleigh, North Carolina, U.S.A., 1987.

[15] Z. Zhang and O.D. Faugeras. Three-dimensional motion computation and object segmentation in a long sequence of stereo frames. *International Journal of Computer Vision*, 7:211–241, 1992.