ELSEVIER

Pattern Recognition Letters 19 (1998) 331–340

Pattern Recognition Letters

Tracking objects with a recognition algorithm [1]

Frederic Jurie [2]

LASMEA - CNRS UMR 6602, Université Blaise-Pascal, Campus Scientifique des Cezeaux, 63177 Aubière, France

Received 10 March 1997; revised 10 November 1997

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. Possible matches between image and model features define volumes within a transformation space. The model is best aligned with the image in volumes satisfying the greatest number of correspondences. Object motion defines a trajectory in the transformation space. We propose an efficient algorithm to compute these transformations. © 1998 Elsevier Science B.V. All rights reserved.

Keywords: Tracking; Object recognition

1. Introduction

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

This paper focuses on the tracking of rigid and modelled 3D objects. The task consists in finding correspondences between model features and image ones from frame to frame. In realistic conditions the scene is cluttered and objects can be occluded. A typical sequence is shown in Fig. 4.

Tracking is usually performed by predicting features' positions in the image, by using their positions in previous images (see Fig. 1(b) for an illustration). Predictions generally rely on a motion model. Within this framework, matched features are spatially close to predictions, but nothing prove their coherence with regard to the model geometry. In clutter, the system is disrupted and may drift away from the object and track background features (see Fig. 1(c) for an illustration). This is specially true in cluttered scenes or when objects are occluded. In this paper, we propose to use a recognition algorithm to overcome this difficulty.

1.1. Previous works

The 3D object tracking with model has been intensively studied in the past years. Due to the lack of space we only cite the most notable works. More references can be found in the paper of Koller et al. (1993).

Several techniques have been proposed to make the matching process more reliable. Some works

[1] Electronic Annexes available. See http://www.elsevier.nl/locate/patrec.

[2] Email: frederic.jurie@lasmea.univ-bpclermont.fr.

PII S0167-8655(97)00175-X

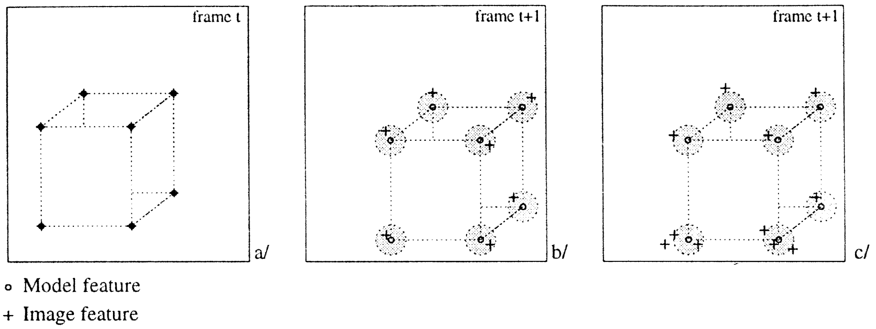

Fig. 1. (a) Frame t: model features and image features are aligned. (b) Frame $t+1$: correspondences are obtained by predicting features positions. (c) Frame $t+1$ (cluttered case): wrong matches will make the transformation inaccurate, and this will have consequences on next predictions.

concern the distance function used to pair features. Deriche and Faugeras (1990) propose to measure the distance of line attributes by using the Mahanalobis distance. Crowley et al. (1992) combine this measure with a velocity constant motion model to estimate the 3D structure of a scene from 2D tracking.

An other way is to use more discriminant features. Manjunath et al. (1996) describe an image feature detector using Gabor filters. The authors affirm that these features are robust and they can easily be tracked in an image sequence. More robustness can also be obtained by tracking textured patterns instead of simple features (like corners or edges). Black and Jepson (1996) describe for example an approach for tracking objects using a view-based representation.

The precision of object models can be of great importance. Koller et al. (1993) propose a very efficient algorithm to track moving vehicles, taking 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 on the motion estimation, in general using Kalman filtering (Gennery, 1992; Harris, 1992; Zhang and Faugeras, 1992).

In the cluttered case, combinatorial data association methods can be effective (Bar-Shalom and Fortmann, 1988). Isard and Blake (1996) notice that data association methods do not apply to moving curves, and propose a stochastic algorithm to propagate an entire probability distribution for object position and shape over time.

1.2. Overview of our approach

The lack of robustness in case of cluttered scenes with unknown model motion is the main criticism that can be levelled at the approaches mentioned above. As explained before, this is mainly due to the fact that in matching process the criterion of difference from prediction is more important that the object structure criterion. In our approach, the priority is given to the spatial coherence of matched features. This is possible by using a recognition algorithm. But the combinatoric aspect of recognition algorithm slows it down and therefore makes it unsuitable for tracking applications.

Recognition strategies have already been used for tracking 3D objects. For example, a paper of Tan et al. (1994) concerns the pose determination of vehicles in traffic scenes. A kind of the generalized Hough transform is used to find consistent matches. Under normal conditions, vehicles stand on the ground. Their number of degree of freedom is reduced from 6 to 3.

The approach developed here is based on the same kind of strategy: it consists in tracking the transformation that best align the model and the image. This turns the problem of object tracking in a problem of dynamic object recognition. The method of Tan et al. (1994) and our differ about the way to compute the transformation.

A model feature is aligned on an image feature if their relative positions in the image satisfy a *model error*. A bounded error model is generally used. In that case, each pair of features (image, model) defines a volume V of the transformation space: transformations included in these volumes align the model feature onto its corresponding one in the image. The greatest number of correspondences is obtained in regions of the transformation space intersecting the maximal number of volumes V. We propose an efficient algorithm to determine these regions.

A contribution of this paper is to use a probabilistic error model instead of a bounded one. It greatly improves the efficiency of the transformation search algorithm. This error model and the way to use it will be described in Section 3. Experiments and results are illustrated in Section 4.

2. Tracking objects in the transformation space

As explained in the introduction, object tracking can be performed by tracking object transformations in 3D pose space. That means 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, recognition quality has been measured by counting the size of the correspondence I. In the present scheme, the aim is to find the set of transformations maximising $|I|$.

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

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

Such methods can be criticized from other points of views, particularly 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)).

Several techniques have been proposed to reduce the search space size. In particular, we note the *coarse-to-fine clustering* (see (Stockman et al., 1982) and the *recursive histogramming* (Thompsom and Mundy, 1987)). These techniques divide the transformation space recursively. Starting with a set including all transformations, the current space is divided into several subspaces. The subspace with the greatest number of correspondences is alternatively kept and divided.

Unfortunately, these techniques pose different problems. At first, the error model is not respected, and above all, the match evaluation relies on counting the size of the correspondence set. Huttenlocher and Cass (1992) (as well as Gavila and Groen (1992)) argued that this measure, although widely used, often greatly overestimates the match quality. They propose instead the *maximum bipartite graph*, or in a simpler way, to count the number of distinct features. If U distinct model features correspond to V image features, the quality of the hypothesis is $\min(U,V)$. In our experiments, we use a similar measure of quality (see Section 3.5). Such techniques require keeping the track of feature pairs.

Cass (1992, 1990) 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 set of all correspondences between model and image features, in a polynomial time.

Breuel (1992) combines both advantages of recursive search that are the respect of the error model and keeping the track of pairing. By deriving Baird research (Baird, 1985), 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 with a *box* [3] in the transformation space containing all transformations. The current box is then subdivided into smaller ones. The same process is repeated recursively, by choosing the half-space giving the best quality of matching.

2.1. 3D pose search

However, Breuel's algorithm is restricted to 2D matching. From what we know, DeMenthon (1993)

[3] A box is a hypercube of the transformation space

is the only one who resumed this method and applied it to 3D problems. We use a similar strategy, but with a probabilistic error model instead of a bounded one.

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

DeMenthon (1993) reports that matching 30 model features (corners in his case) with 200 images features requires several hours of computation. We observed that such poor results are mainly due to the bounded error model, as illustrated in Fig. 2. 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.

Therefore we propose to substitute a probability function for the bounded error. The probability of a match subject to a box of transformations is defined in Section 3.1. In that case, the evaluation of a box is more complicated than computing intersecting polyhedrons, as by using the bounded error model.

The algorithm starts 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 and each node represents a subspace. The leave are the smallest regions taken into account.

Breuel (1992) 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, the remaining boxes are subdivided, if their score is higher than the best score obtained on a ‘‘leaf’’ box.

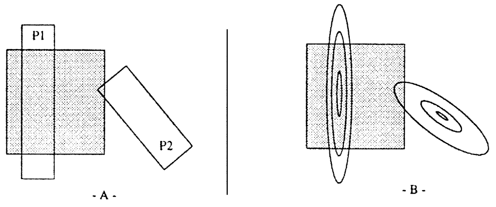

Fig. 2. Bounded error model, probabilistic error model.

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 is why we recommend an 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 shows the efficacy of the algorithm.

2.2. From scaled orthographic transformation to full perspective transformation

Vectors are written down in bold font. A full perspective transformation can be represented by $\boldsymbol{P}$ such that:

$$\boldsymbol{P} = \begin{pmatrix} \boldsymbol{i} & tx \\ \boldsymbol{j} & ty \\ \boldsymbol{k} & tz \\ (0,0,0) & 1 \end{pmatrix} = \begin{pmatrix} \boldsymbol{P}_1 \\ \boldsymbol{P}_2 \\ \boldsymbol{P}_3 \\ \boldsymbol{P}_4 \end{pmatrix}.$$

$\boldsymbol{M}_0 = (X_0,Y_0,Z_0,1)$ is the origin of the object reference, and $\boldsymbol{M}_i = (X_i,Y_i,Z_i,1)$ the ith model point projected onto $\boldsymbol{p}_i = (x_i,y_i)$. With these notations, the perspective transformation can be written (see (DeMenthon and Davis, 1992)):

$$\begin{cases} \boldsymbol{M}_0\boldsymbol{M}_i\boldsymbol{P}_1 \dfrac{f}{tz} = x_i(1+\varepsilon_i), \\ \boldsymbol{M}_0\boldsymbol{M}_i\boldsymbol{P}_2 \dfrac{f}{tz} = y_i(1+\varepsilon_i), \end{cases}$$

with $\varepsilon_i = \boldsymbol{M}_0\boldsymbol{M}_i\boldsymbol{P}_3/t_z - 1$. f denotes the camera focal.

Scaled orthographic projection assumes that objects lie in a plane parallel to the image plane passing through the origin of the object frame. This is equivalent to the approximation: $\varepsilon_i = 0$.

The perspective transformation can be approximated with a scaled orthographic transformation, and conversely a perspective transformation can be obtained from a scaled orhtographic transformation by computing the ε_i.

To compute the ε_i e have to determine t_z and $\boldsymbol{k}$. $t_z = f/\|\boldsymbol{I}\| = f/\|\boldsymbol{J}\|$, and we have:

$$\begin{cases} (\boldsymbol{i},tx) = \boldsymbol{P}_1 \dfrac{t_z}{f}, \\ (\boldsymbol{j},ty) = \boldsymbol{P}_2 \dfrac{t_z}{f}. \end{cases}$$

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

When ε_i is obtained, we modify the coordinates of image features, by multiplying them by $(1+\varepsilon_i)$. 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.

2.3. Motion model

The tracking of transformations over time consists in predicting the box of transformations to be explored. Any motion model can be used. When using a Kalman filter, the probability density propagation process allows to determine the box size.

To prove the efficacy of the algorithm, the results given in this paper have been obtained without any motion model: the initial box on the first image represent the entire transformation space. During the sequence, the initial box has a constant size (computed from the maximal acceleration of objects in the scene) and is centered on the last transformation obtained. If the object is not detected, the size of the initial box is doubled.

3. 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? (A ‘‘box’’ an is hypercube of the transformation 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 transformations. We show in Section 2.2 how it can be extended to perspective transformations.

3.1. Probability of two segments to be matched

Supposing that the quality of correspondence between the transformation of a model segment m and an image segment s is a function of their distance in the image. We evaluate this quality through the conditional probability of correspondence knowing the distance. The conditional probability denoted $P(m \rightarrow s|d)$ is learnt from examples.

Segments are represented by the position of their extremities. The probability of a correspondence therefore depends on the distance $D(m,S)$ between extremities. Rather than using the Euclidean distance, we use a measure which is less sensible to segment fragmentation.

D_{12} (D_{22}) is the support line of the segment (see Fig. 3 for details). D_{11} (respectively D_{21}) is orthogonal to D_{12} and cuts D_{12} at the first (respectively second) extremity of the segment. These lines are defined by equations $\boldsymbol{n}_i^j \cdot \boldsymbol{p} = 0$, $(i,j) \in \{1,\dots,2\}^2$, where $\boldsymbol{p} = (x,y,1)$ denotes a point of the image plane and $\boldsymbol{n}_i^j$ a vector defined as follows: if equation of D_{ij} is $ax+by+c=0$, then

$$n_{ij} = \left(\frac{a}{\sqrt{a^2+b^2}}, \frac{b}{\sqrt{a^2+b^2}}, \frac{c}{\sqrt{a^2+b^2}} \right).$$

Let $\boldsymbol{d} = (d_1^1, d_1^2, d_2^1, d_2^2)$ the vector made of the four distances between model segment extremities and the four lines. As the distances $\boldsymbol{d}_i^j$ take discrete values, we can define the conditional probability of a segment m of being matched with a segment s

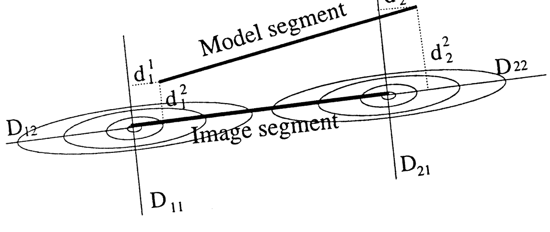

Fig. 3. Error model for line segments.

knowing $\boldsymbol{d}$ as the function $P(m \rightarrow s|d) = f(D(m,s))$ (f has been learnt from a set of examples), with

$$D(m,s) = \sum_{i=1}^{2} \sum_{j=1}^{2} \left(\frac{d_i^{j2}}{2\left(\sigma_i^{\,j}\right)^2} \right).$$

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

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

3.2. Segments distance given a scaled orthographic transformation

Let $\boldsymbol{t} = (I_x, I_y, I_z, T_x, J_x, J_y, J_z, T_y)^{\mathrm{t}}$ be a scaled orthographic transformation $(\|\boldsymbol{I}\| = \|\boldsymbol{J}\|,\ \boldsymbol{I} \cdot \boldsymbol{J} = 0)$. This transformation can also be written with the following homogeneous matrix:

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

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

Let us denote $\boldsymbol{P} = (X,Y,Z,1)^{\mathrm{t}}$ a point in the 3D object reference, $\boldsymbol{s} = (p_1, p_2)$ an image segment and $\boldsymbol{m} = (P_1, P_2)$ the corresponding model segment. $\boldsymbol{m}$ is mapped (in the image plane) on segments $(\boldsymbol{T}(\boldsymbol{t}) \cdot \boldsymbol{P}_1, \boldsymbol{T}(\boldsymbol{t}) \cdot \boldsymbol{P}_2)$ by the projection $\boldsymbol{T}(\boldsymbol{t})$, where points $\boldsymbol{P}_i$ are segment extremities.

The product $\boldsymbol{n}_i^j \cdot \boldsymbol{T}(\boldsymbol{t}) \cdot \boldsymbol{P}_i$ can be re-written as follows:

$$\begin{aligned} &\boldsymbol{n}_i^j \cdot \boldsymbol{T}(\boldsymbol{t}) \cdot \boldsymbol{P}_i \\ &\quad = \left(n_{ix}^j, n_{iy}^j, 1\right) \boldsymbol{T}(\boldsymbol{t}) (X,Y,Z,1)^{\mathrm{t}} \\ &\quad = \left(n_{ix}^j X, n_{ix}^j Y, n_{ix}^j Z, n_{ix}^j, n_{iy}^j X, n_{iy}^j Y, n_{iy}^j Z, n_{iy}^j\right) \boldsymbol{t} \\ &\quad = \mathbf{hp}_i^j \cdot \boldsymbol{t}. \end{aligned}$$

This product can be geometrically interpreted as the distance from the transformation $\boldsymbol{t}$ to the hyperplane $\mathbf{hp}_i^j$, in the scaled orthographic transformation 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 hyperplanes.

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

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

3.3. 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 transformations as

$$D(m,s,Box) = \min_{t \in Box} D(m,s,t).$$

(A box has been defined to be a hypercube of the transformation space.)

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 be the affine manifold generated by the four hyperplanes produced by a pair of matched segments. We first compute the position of the point $\boldsymbol{t}_0 \in V$ such that $\forall \boldsymbol{t} \in V,\ d(c,\boldsymbol{t}_0) \leqslant d(\boldsymbol{c},\boldsymbol{t})$ (where $\boldsymbol{c}$ is the center of the box and $d()$ the Euclidean distance), by means of the Lagrangian method. If $\boldsymbol{t}_0$ is not in the box then $\boldsymbol{t}_{min}$ is taken as the intersection of the line $(\boldsymbol{c},\boldsymbol{t}_0)$ with the convex hull of the box ($\boldsymbol{t}_{\min} = \boldsymbol{t}_0$ if $\boldsymbol{t}_0$ is included in the box). If $\boldsymbol{t}_{\min}$ is not in the box, its position is iteratively adjusted to minimize the distance $D(\boldsymbol{m},\boldsymbol{s},\boldsymbol{t})$. For that purpose, we use the *alternating variable strategy* in which at iteration k $(k \in [1,\dots,8])$ 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.

3.4. 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 $\mathscr{M}$ (number of features of M), there are $2^{\mathscr{M}}$ 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 $m \nrightarrow$ if the segment is not matched). Configurations can be grouped according to their number of matches. The set E^k, $k \leqslant \mathscr{M}$, includes configurations that match k model segments. We denote by $E^k = \bigcup_{j=1}^{j < \subset^k_{\mathscr{M}}} \gamma_j^k$, and $\Gamma = \bigcup_{i=1}^{i \leqslant \mathscr{M}} E^i$, the set of all possible exhaustive and mutually exclusive configurations.

$$P(M|T) = \sum_{\gamma \in \Gamma} P(M|T,\gamma) P(\gamma|T)$$

$$= \sum_{k=1}^{k \leqslant \mathscr{M}} \sum_{j=1}^{j \leqslant \subset^k_{\mathscr{M}}} P\left(M|T,\gamma_j^k\right) P\left(\gamma_j^k|T\right).$$

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

$$P(M|T) = \sum_{k=1}^{k \leqslant \mathscr{M}} \sum_{j=1}^{j \leqslant \subset^k_{\mathscr{M}}} P\left(M|\gamma_j^k\right) P\left(\gamma_j^k|T\right).$$

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:

$$\forall k \in \{1,\ldots,\mathscr{M}\}, \forall i \in \left[1,\ldots,\subset^k_{\mathscr{M}}\right]$$

$$P\left(M|\gamma_i^k\right) = P\left(M| \bigcup_{i=1}^{i \leqslant \subset^k_{\mathscr{M}}} \gamma_i^k\right) = P(M|E^k).$$

The probability $P(M|T)$ can therefore be written:

$$P(M|T) = \sum_{k=1}^{k \leqslant m} P(M|E^k) \sum_{j=1}^{j \leqslant \subset^k_{\mathscr{M}}} P\left(\gamma_j^k|T\right).$$

$P(M|E^k)$ 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

$$P(E_i|T) = \sum_{j=1}^{j \leqslant \subset^k_{\mathscr{M}}} P\left(\gamma_j^k|T\right)$$

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

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 $P(m_i b(i)|T)$, 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

$$\forall(k,l) \in \{0,\ldots,\mathscr{M}\}^2$$

$$P\left(m_{I(l)} \rightarrow |T\right) > P\left(m_{I(k)} \rightarrow |T\right) \Rightarrow k < l,$$

then

$$P(E_i|T) = P\left(m_0 \overset{b(0)}{\rightarrow}, \ldots, m_i \overset{b(i)}{\rightarrow}, \ldots, m_{\mathscr{M}} \overset{b(\mathscr{M})}{\rightarrow}\right)$$

$$\geqslant \prod_{j=1}^{j \leqslant i} P(m_{I(j)} \rightarrow) \prod_{j=i+1}^{j \leqslant \mathscr{M}} \left(1 - P\left(m_{I(j)} \nrightarrow\right)\right).$$

3.5. Distinct correspondences

A model segment may be associated with more than one scene segment; conversely a scene segment

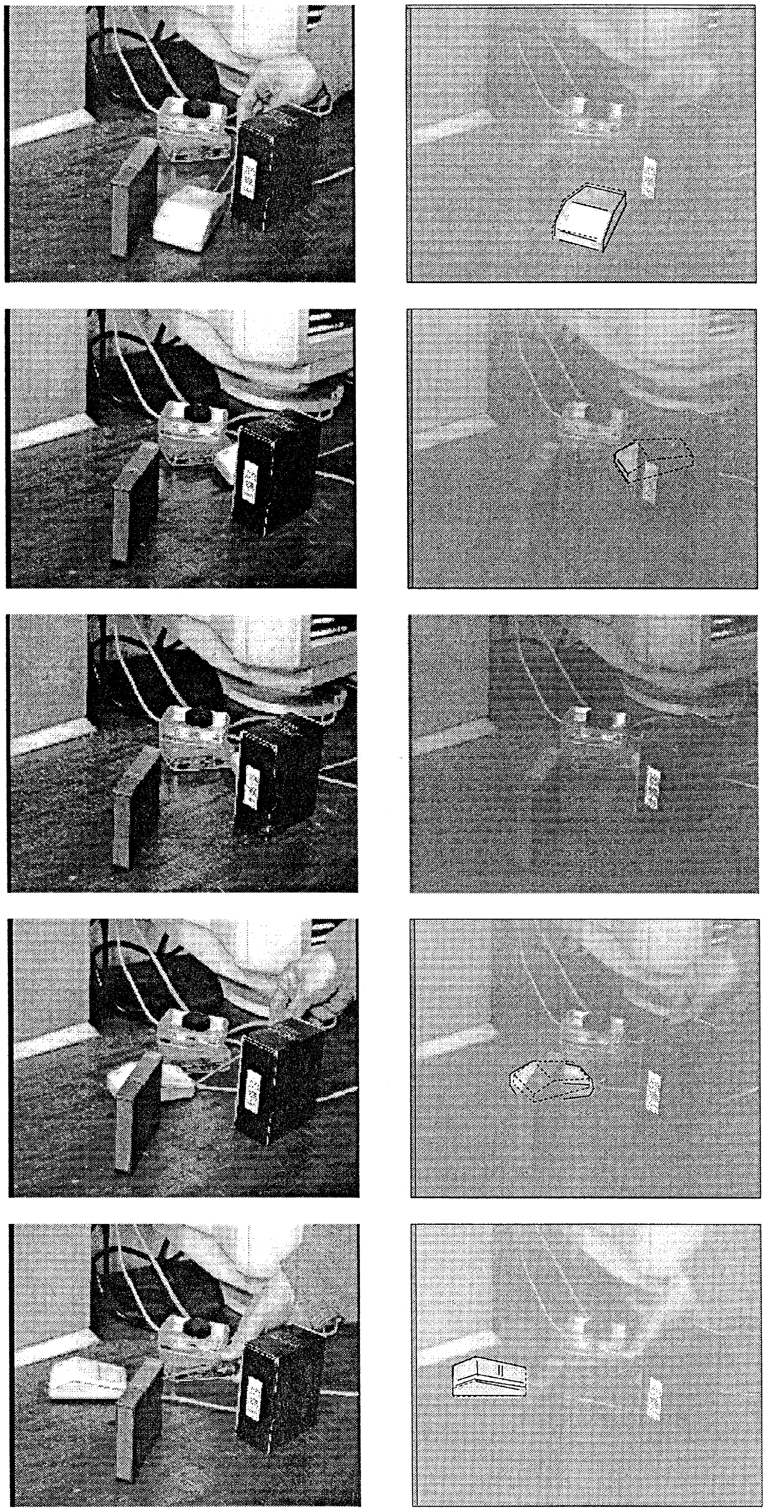

Fig. 4. Model (dotted lines) and correspondences obtained on the ‘‘mouse’’ sequence.

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) or Huttenlocher and Cass (1992) 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 $P(m \rightarrow) = P(\bigcup_{j=1}^{j \leq \mathscr{S}} m \rightarrow s_j)$ where $\mathscr{S}$ is the number of image segments.

4. Experiments and results

4.1. Tracking experiments

Fig. 4 presents some of the results obtained with a hundred images of the ''mouse sequence''. In that sequence, the mouse moves to the right until it completely disappears behind the box, then it start again to cross the screen to the left. The motion is unpredictable.

To guarantee the achievement of the right solution, the constant N (defined in Section 2.1) has to be set to about 5.

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

The first image is processed several times (4 times in our case) in order to compute the perspective transformation. The processing time is about 200 ms on our HP-700 workstation (without including processing time for extracting the line segments).

The transformation found on the fourth image is not very accurate, because the mouse rotates behind the box. This movement is difficult to detect because there are occlusions. Furthermore, we do not use the fact that the motion is planar in the sequence.

4.2. Recognition experiments

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

This recognition stage is illustrated in Fig. 5.

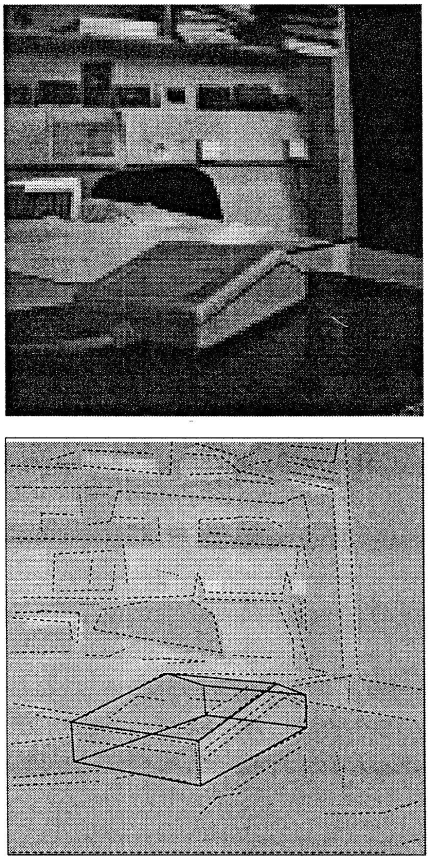

Fig. 5. Recognition stage: grey level image, line segments extracted (dotted lines) and pose computed (solid lines).

4.3. Improvement obtained by using the proposed probabilistic framework

In Section 2.1 we explained that only the N more probable boxes are explored simultaneously. In the ideal case $N = 1$, the correct solution can be reached by exploring only one branch of the search tree. In the sequence given in Fig. 4 and with the bounded error model, the constant N has to be set to about

100 to guarantee the achievement of the right solution. By using the probabilistic framework, $N = 5$ is enough. The recognition time is 20 times shorter than by using the probabilistic framework.

Furthermore, the probabilistic error model tolerates incorrect endpoint detection improving the alignment of the model on the image in comparison to the bounded error model.

5. Conclusion

In our opinion, is that the presented recognition algorithm can be used as a robust tracking algorithm. Correspondences are computed in the transformation space rather than in the image space. This 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 does not have to be known. There is no probability propagation process and consequently there is no risk for the system to drift.

References

Baird, H., 1985. Model-based Image Matching Using Location. MIT Press, Cambridge, MA.

Bar-Shalom, Y., Fortmann, T., 1988. Tracking and Data Association. Academic Press, New York.

Black, M., Jepson, A., 1996. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. In: Proc. European Conf. on Computer Vision, vol. I, Cambridge, UK, pp. 329–342.

Breuel, T., 1992. Fast recognition using adaptive subdivisions of transformation space. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, Champaign, IL, pp. 445–451.

Cass, T., 1990. Feature matching for object localization in the presence of uncertainty. In: Proc. IEEE Internat. Conf. on Computer Vision, Osaka, Japan, pp. 360–364.

Cass, T., 1992. Polynomial-time object recognition in the presence of clutter, occlusions, and uncertainty. Proc. European Conf. on Computer Vision, Santa Margherita Ligure, Italy, pp. 834–842.

Crowley, J., Stelmaszyk, P., Skordas, T., Puget, P., 1992. Measurement and integration of 3-d structures by tracking edge lines. Internat. J. Comput. Vision 8, 29–52.

DeMenthon, D., 1993. De la vision artificielle à la réalité synthétique: Système d'interaction avec un ordinateur utilisant l'analyse d'images vidéo. These, Univ. Grenoble I, Laboratoire TIMC/IMAG.

DeMenthon, D., Davis, L., 1992. Model-based object pose in 25 lines of code. In: Proc. European Conf. on Computer Vision, Santa Margherita Ligure, Italy, pp. 19–22.

Deriche, R., Faugeras, O., 1990. Tracking line segments. Image and Vision Comput. 8 (4), 261–270.

Gavila, D., Groen, F., 1992. 3D object recognition from 2D images using geometric hashing. Pattern Recognition Lett. 13, 263–278.

Gennery, D., 1992. Visual tracking of known three-dimensional objects. Internat. J. Comput. Vision 7 (3), 243–270.

Grimson, W., Huttenlocher, D., 1990. On the sensitivity of the Hough transform for object recognition. IEEE Trans. Pattern Anal. Machine Intell. 12 (3), 255–274.

Harris, C., 1992. Tracking with rigid models. In: A. Blake and A. Yuille. Active Vision, MIT Press, Cambridge, MA, pp. 59–74.

Huttenlocher, D., Cass, T., 1992. Measuring the quality of hypotheses in model-based recognition. In: Proc. European Conf. on Computer Vision, Santa Margherita Ligure, Italy, pp. 773–777.

Isard, M., Blake, A., 1996. Contour tracking by stochastic propagation of conditional density. In: Proc. European Conf. on Computer Vision, vol. I, Cambridge, UK, pp. 343–356.

Koller, D., Daniilidis, K., Nagel, H., 1993. Model-based object tracking in monocular image sequences of road traffic scene. Internat. J. Comput. Vision 10 (3), 257–281.

Manjunath, B., Shekhar, C., Chellappa, R., 1996. A new approach to image feature detection with applications. Pattern Recognition 29 (4), 627–640.

Stockman, G., Kopstein, S., Bennet, S., 1982. Matching images to model for registration and object detection. IEEE Trans. Pattern Anal. Machine Intell. 4 (3), 229–241.

Tan, T., Sullivan, G., Baker, K., 1994. Pose determination and recognition of vehicles in traffic scenes. In: Proc. European Conf. on Computer Vision.

Thompsom, D., Mundy, J., 1987. Three dimensional model matching from an unconstrained viewpoint. In: Proc. Robotics and Automation, Raleigh, NC, pp. 208–220.

Zhang, Z., Faugeras, O., 1992. Three-dimensional motion computation and object segmentation in a long sequence of stereo frames. Internat. J. Comput. Vision 7, 211–241.