Model-Based Temporal Object Verification Using Video

Baoxin Li, Rama Chellappa, *Fellow, IEEE*, Qinfen Zheng, and Sandor Z. Der

***Abstract*—An approach to model-based dynamic object verification and identification using video is proposed. From image sequences containing the moving object, we compute its motion trajectory. Then we estimate its three-dimensional (3-D) pose at each time step. Pose estimation is formulated as a search problem, with the search space constrained by the motion trajectory information of the moving object and assumptions about the scene structure. A generalized Hausdorff metric, which is more robust to noise and allows a confidence interpretation, is suggested for the matching procedure used for pose estimation as well as the identification and verification problem. The pose evolution curves are used to assist in the acceptance or rejection of an object hypothesis. The models are acquired from real image sequences of the objects. Edge maps are extracted and used for matching. Results are presented for both infrared and optical sequences containing moving objects involved in complex motions.**

***Index Terms*—Hausdorff matching, moving object recognition, object recognition, video processing.**

I. Introduction

FOR many years, object recognition algorithms have been based on a single image or a few images acquired from different aspects. While advances have been made in simple constrained situations such as indoor environments, object recognition in natural scenes remains a challenging problem. Among the many difficulties, a prominent one is that in real applications, theoretically there exist infinitely many poses (orientations) for a given object. Therefore, two-dimensional (2-D) approaches, which are largely based on 2-D matching under some simplified transformation group, will not solve the three-dimensional (3-D) object recognition problem. To overcome the need for search in the viewpoint space, approaches based on geometric invariants have been proposed (for example, see [16] and [18]). Although the invariance approach is theoretically attractive, it would be difficult to apply it to complex objects in natural scenes. Appearance-based recognition schemes (for example, see [11]) try to tackle the viewpoint problem by using visual learning. In [11], the authors reported promising results for a test data set. Although appearance-based approaches do not require explicit feature extraction, their success relies on visual learning from a training set. A good training set is not always easy to obtain. Besides, due to shape variations, training images always contain some background region. Although when training, one can set the background to a uniform value (as in [11]), it is not always possible to black out the background at the recognition stage—one needs to know the object type and its exact orientation in order to do so, which is what a recognition algorithm is attempting to do. Backgrounds can greatly affect the projection of an input image onto the eigenspace. In addition, when the camera sensor is infrared, as in most surveillance applications, the object signature becomes too variable to be characterized by only a few images even at a fixed pose. In [9], some recognition algorithms including several learning algorithms were compared, using a large database containing over 17 000 images of ten object classes. It was reported that even the best recognition results were unsatisfactory for this infrared database. One possible explanation for the results in [9] is that when objects have abundant pose variations, the appearance manifolds become heavily overlapped, making recognition harder. In such a situation, one may have to resort to some geometric (shape) features, which, unfortunately, are again dependent on viewpoint.

An interesting observation is that when the object is moving, human beings can quickly guess its pose, and then verify some features unique to that pose. This suggests that additional information can be exploited to make object recognition more feasible when a video sequence is available. This paper presents a technique for model-based temporal object verification/identification. In a sense, verification and identification are constrained cases of recognition. To be specific, in this paper, identification refers to the following problem: given an image sequence containing a moving object, to identify the object as one of a few hypotheses; or, to identify the desired object in a sequence containing multiple objects. Identification is dynamic in that we have a time-evolving scene due to object motion and possible sensor motion. Verification is used in a slightly different situation, which answers the following questions: Is this the object seen in the previous frames? and How confident of this am I? This is especially interesting in situations of temporary loss of tracking due to, for example, occlusion by other objects. Verification is in a sense similar to the tracking problem but here it emphasizes the acceptance or rejection of a certain object hypothesis, rather than just tracking by using some features. Obviously, model-based verification/identification has many applications. For example, in visual autonomous surveillance as in following a face in the crowd, the recognition problem can often be reduced to the verification/identification problem.

Manuscript received March 12, 1999; revised February 26, 2001. This work was supported by the Advanced Sensors Consortium (ASC) sponsored by the U.S. Army Research Laboratory under the Federated Laboratory Program, Cooperative Agreement DAAL01-96-2-0001. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Michael R. Frater.

B. Li is with Sharp Laboratories of America, Camas, WA 98607 USA (e-mail: bli@sharplabs.com).

R. Chellappa and Q. Zheng are with the Center for Automation Research University of Maryland, College Park, MD 20742 USA.

S. Z. Der is with the Army Research Laboratory, Adelphi, MD 20783 USA.

Publisher Item Identifier S 1057-7149(01)04479-7.

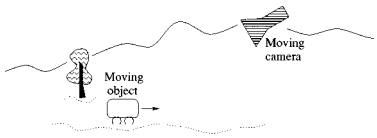

Fig. 1. Typical identification/verification setup using video from a moving camera platform.

In applications such as visual autonomous surveillance, the camera itself is often moving during the acquisition process. A general setup for this kind of problems is illustrated in Fig. 1. Due to camera motion, a sensor motion compensation process is often needed to remove the unwanted camera motion if we want to detect the object based on its motion.

In this paper, from image sequences containing the moving object, the 3-D pose of the object is estimated at each time step. Pose estimation is formulated as a search problem, with the search space strictly constrained by the motion trajectory information of the moving object and assumptions about the scene structure. A generalized L_p version of the Hausdorff metric [1], which is more robust to noise and allows a confidence interpretation, is suggested for the search problem. The pose evolution curves are used to assist in the acceptance or rejection of an object hypothesis. Experiments on several sequences are presented. The experiments demonstrate how the concepts and algorithms for model-based temporal identification/verification could work in real applications.

II. Matching Based on the Hausdorff Metric

The Hausdorff metric [7] is a mathematical measure for comparing two sets of points in terms of their least similar members. Formally, given two finite point sets $A = \{a_1, \ldots, a_p\}$ and $B = \{b_1, \ldots, b_q\}$, the Hausdorff metric is defined as

$$H(A, B) = \max\{h(A, B), h(B, A)\} \tag{1}$$

where

$$h(A, B) = \sup_{a \in A} \inf_{b \in B} \|a - b\| \tag{2}$$

and $\|\cdot\|$ is an underlying norm. If a model image and a scene image are first processed to give two characteristic point sets, then the model-scene matching is realized by comparing the point sets in terms of the Hausdorff metric. Intuitively, when there are multiple models, recognition is simply done by computing the corresponding Hausdorff distances between the models and the scene, and then picking out the best match.

A. Some Modified Versions of the Hausdorff Metric

Although theoretically attractive, the Hausdorff metric H is not directly usable in practice, because the $\sup$ or $\max$ operation in the definition makes h and hence H very sensitive to noise—a single noisy point can pull the value of H far from its noise-free counterpart. Some modifications have therefore been proposed in the literature. For example, in [9], a weighted sum version was proposed and found to slightly improve the recognition rate; and in [5], a Kth ranked partial "distance" $h(A, B)$ was used to detect a model in a static scene. The same partial "distance" was also used to track people in [8]. Although these modifications improve the robustness in practice, the obtained "distances" (a weighted one in [9] and a Kth ranked one in [5]) no longer possess the properties of a metric. That is to say they are not real *distances* in the strict sense. We argue that being a metric (i.e., obeying the axiomatic rules for a metric) is important because when doing identification or verification, generally we have several hypotheses, and we need to use a measure that can reflect our confidence in choosing one over the others. This is not like detection or tracking, where one only needs to find an optimal match for a given mask. For example, it's easy to construct examples where a partial distance does not give a measure of similarity between point sets. Although these examples are unlikely to occur, one does face difficulties when the models are relatively simple point sets (with not too many points) while the scene is highly cluttered. Therefore, the above-mentioned modified versions of the Hausdorff distance do not necessarily offer good measures for comparison among different models.

B. L_p Version of the Hausdorff Metric

Another equivalent representation of the Hausdorff metric is (see [6])

$$H(A, B) = \sup_{x \in X} |\rho(x, A) - \rho(x, B)| \tag{3}$$

with

$$\rho(x, A) \triangleq \inf_{a \in A} \{\rho(x, a)\}$$

where X is a set and ρ a metric such that (X, ρ) is a metric space, and $A \subseteq X$ and $B \subseteq X$. In the image analysis context, X can simply be the set of all the image grid points, and ρ is usually the L_2 norm, while A and B are two compact sets in the image plane. In this paper, we use edges as the features for matching; thus, A and B are just edge maps derived from intensity images.

To alleviate the instability in (3) due to the $\sup$ or $\max$ operation, Baddeley [1] has suggested an L_p average as follows:

$$H^p(A, B) = \left[\frac{1}{n(X)} \sum_{x \in X} |\rho(x, A) - \rho(x, B)|^p\right]^{1/p} \tag{4}$$

where $n(X)$ is the number of points in X, and $1 \leq p < \infty$. So defined $H^p(A, B)$ is still a metric, and topologically equivalent to $H(A, B)$, but is more robust to noisy data since the contribution of a single point has been weighted. Also, by using the average, (4) has an "expected risk" interpretation: given A, a set B which minimizes $H^p(A, B)$ is one which maximizes the pixelwise likelihood of $\{\rho(x, A) = \rho(x, B)\}$ (if A and B are treated as random sets). In applications, a cutoff function $w(t, c) = \min\{t, c\}$, for a fixed $c > 0$, is incorporated into (4) to give

$$H^p(A, B) = \left[\frac{1}{n(X)} \sum_{x \in X} |w(\rho(x, A), c) - w(\rho(x, B), c)|^p\right]^{1/p}. \tag{5}$$

The resulting $H^p(A, B)$ is again a metric, and topologically equivalent to $H(A, B)$. Note that in practice it is unnecessary to

compute $\rho(x, A)$ by its definition (i.e., by computing $\rho(x, y) = ||x - y||$), which is too expensive, especially with the L_2 norm. Instead, distance transformations [3] are used. Thus, using a supporting set X will not cause significant extra computation, although X is larger than A and B.

C. Identification/Verification with H^p

Given two point sets, H^p provides a similarity measure between them. When this measure is applied to the identification/verification problem, we are concerned not only with how good the match is but also with where the match happens in the scene. It would be meaningless to compute H^p between a small model and a large scene image. Instead, usually a region of interest (ROI) is detected first, and matching is carried out between the ROI and the model. In particular, in identification problems, given the edge map R of an ROI from the scene image and m models $M_i, i = 1, \ldots, m$, the task is to find a model M_j and a transformation $T' \in \mathcal{T}$ such that

$$H^p(R, T'(M_j)) = \min_{i=1}^{m} \min_{T \in \mathcal{T}} H^p(R, T(M_i)) \tag{6}$$

where $\mathcal{T}$ is an allowed transformation group for the application. Such M_j will be regarded as the potential object appearing in the current scene. Since H^p is a metric, we can also interpret the values $\min_{T \in \mathcal{T}} H^p(R, T(M_i)), i = 1, \ldots, m$ as a measure of confidence of choosing M_i in the current frame. If $m = 1$, then the problem is reduced to detecting an object in the scene; in addition, if the model is extracted from earlier frames in the sequence, the problem reduces to one of tracking and verification.

It is not hard to search over $\mathcal{T}$ when $\mathcal{T}$ is the translation group. However it is difficult to consider other transformation groups such as affine. Even if we consider only rotation and scale, the search becomes a daunting task. The authors of [8] have proposed an efficient search scheme for rotation using the fact that the image takes value only on a digitized grid. In Section III-B, motion-based segmentation is used to minimize the need for search over the scale space.

III. Model-Based Pose Estimation and Object Verification

In this section, we present an approach to pose estimation and verification based on matching using the L_p version of the Hausdorff metric, with the motion trajectory information from motion analysis being used as a constraint to reduce the search space. The model acquisition step is discussed in Section III-A. Section III-B gives a brief overview of a framework for detection, tracking and segmentation of moving objects in video acquired by a moving platform. Pose estimation and object identification are discussed in Section III-C. Section III-D discusses methods for excluding clutter from the ROI. The pose evolution curve is defined in Section III-E. Section III-F discusses the interpretation of H^p as a confidence measure, and a confidence figure is defined. Experimental results are presented in Section IV.

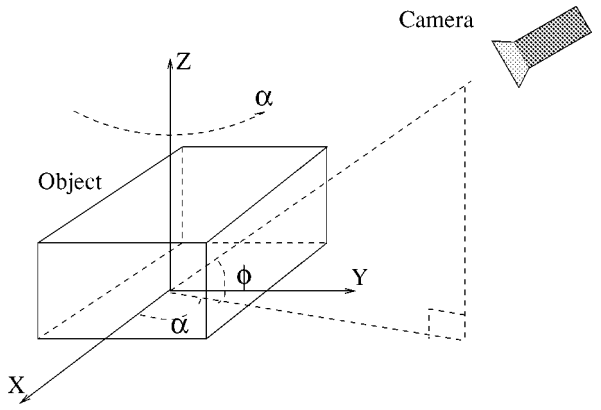

Fig. 2. Two angles defining the object orientation with respect to the camera under the assumption of level ground (i.e., the X–Y plane is horizontal).

A. Model Acquisition

When a 3-D object is subject to complex 3-D motion with respect to the camera, in general, multiple views of the object are needed for adequate modeling of the object. For a matching-based approach, images from these views constitute a model base. In general, there are two ways for constructing a model base: by using computer aided design (CAD) models or by extracting objects from real images. Three-dimensional CAD models allow one to easily manipulate the object orientation. However most objects of interest do not come with CAD models. In this paper, for the identification experiments, the models are constructed from real images: model images were taken at various camera depression angles, with the objects rotating horizontally. This allows the approach to extend to real applications easily: for any real object of interest, we can build its model by acquiring a set of images of the object at different viewpoints, hence relaxing the need for a 3-D CAD model.

Although in general, the orientation of a rigid object has three degrees of freedom, some assumptions can be made for specific applications. For example, if the object is on nearly level ground, as in most surveillance applications, its orientation can be characterized by only two variables. If we use an object-centered coordinate system, the object orientation is equivalent to the camera viewing angles, defined by two angles α and ϕ as illustrated in Fig. 2.

Notice that even under the above assumption, there are still infinitely many orientations in theory. But some observations can be made to determine the orientations that are characteristic. For example, with ϕ fixed, although α can vary from 0° to 360°, it is not necessary to store images at every degree of α since the object looks very similar when α changes only by a small number (say, less than 5°). A similar argument is valid for ϕ, which takes values in the interval [0°, 90°]. More constraints can be included for a specific application. For example, in many applications, the value of ϕ can only change within a small range or can even be fixed. Research has shown that it seems that the human visual system represents objects only by a few 2-D views (e.g., [14]). Not much is known, however, about the number of views required for a specific object. In this work, we represent an object with a model base in which α and ϕ take on only a finite set of values.

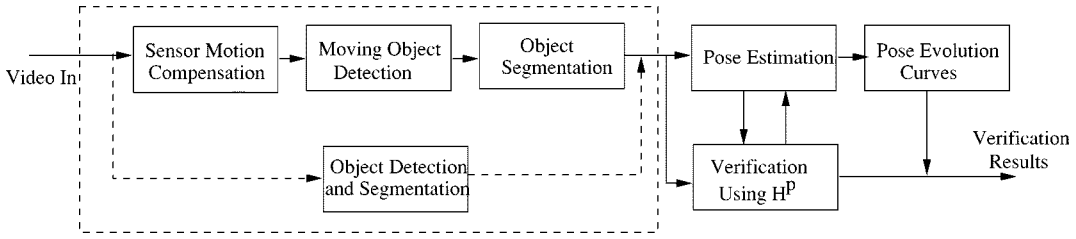

Fig. 3. Whole framework shown as a diagram of procedures.

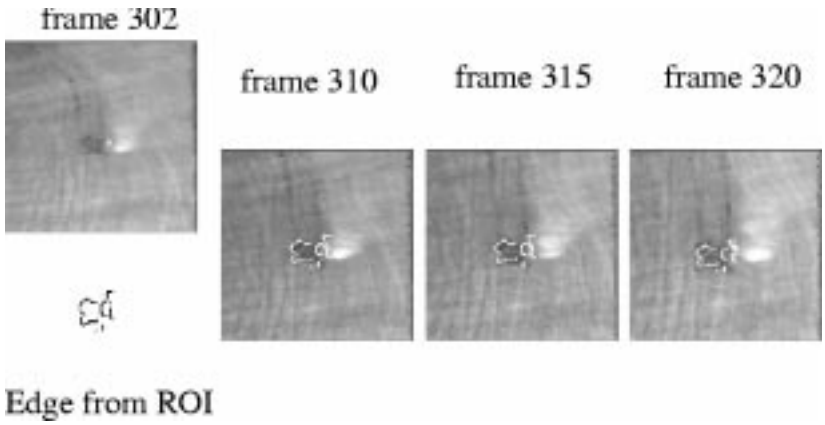

Fig. 4. Dynamic verification with the H^p metric: a moving object is first detected and its edge map is used in the following frames for verification. The sequence is infrared with frame size 128×128 pixels.

Besides orientation, *scale* is another variable that needs to be considered. It is possible to transform the scene object, the model, or both, to bring them to the same size before performing matching. Each method has its advantages and disadvantages, as discussed in [15]. For appearance-based approaches, the model base (or the parametric space) is constructed at a fixed scale. Therefore, during recognition, the object size needs to be normalized with respect to the model base. There is a potential problem with the above normalization: if the object is at a much lower resolution than the training images, normalization can only bring the object to the same *size*, but not to the same *scale* at which the training images were looked at. Considering this, we propose to acquire the model images at a resolution higher than that at which the object is most likely to appear in real applications. At the identification/verification stage, we bring the model to the scale of the scene object. Note that this is essentially a downsampling process, and we are getting rid of detailed information rather than trying to add more information. Equivalently, one can build a multi-scale model base which keeps several versions at different scales for an object at a certain orientation.

B. Framework for Moving Object Detection, Tracking, and Segmentation

As stated in the introduction, in many applications, the camera is moving during the acquisition process. Therefore sensor motion compensation is typically required before one can exploit the object-induced motion information. Sensor motion compensation is also known in the literature as image sequence stabilization. Roughly speaking, there are two types of stabilization methods: feature-based and optical flow-based. We believe that optical flow-based methods are more appropriate if the camera is of the infrared type (as in most surveillance applications) since reliable feature detection is more difficult in infrared imagery. Also, the brightness (thermal) constancy assumption is more appropriate for infrared images, which is essential to the computation of optical flow. We follow the framework reported in [10] which integrates image sequence stabilization, moving object detection, tracking and segmentation, to form the frontend of the recognition system. Stabilization is based on the optical flow-based approach reported in [13], where the optical flow is modeled as a weighted sum of basis functions, and permits accurate and fast motion computation. The computed flow field is then used to estimate the motion parameters. An affine transformation is used to model the sensor motion. That is, the transformation between the pixels of frame k and frame $k+1$ is defined by

$$\mathbf{P_i}' = \begin{pmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{pmatrix} \mathbf{P_i} + \begin{pmatrix} T_x \\ T_y \end{pmatrix} \tag{7}$$

where $\mathbf{P_i} = (x_i, y_i)^T$ and $\mathbf{P_i}' = (x_i', y_i')^T$ are pixels of frame k and frame $k+1$, respectively.

After sensor motion compensation, changing parts are detected from the camera motion compensated frame differences. These changing parts are segmented from the background to form ROI's, and then tracked and updated incrementally based on the successive motion measurements. If the object is big enough, it is also possible to get its boundary by motion-based segmentation. Otherwise, a bounding box is used to define the ROI. Segmentation greatly facilitates matching: recall from (5), that a supporting set X is needed for computing H^p. In practice, a smaller X is desired to facilitate the computation. Segmentation not only provides a small X but also greatly decreases the search region for the $\min$ operation in (6). For example, we can

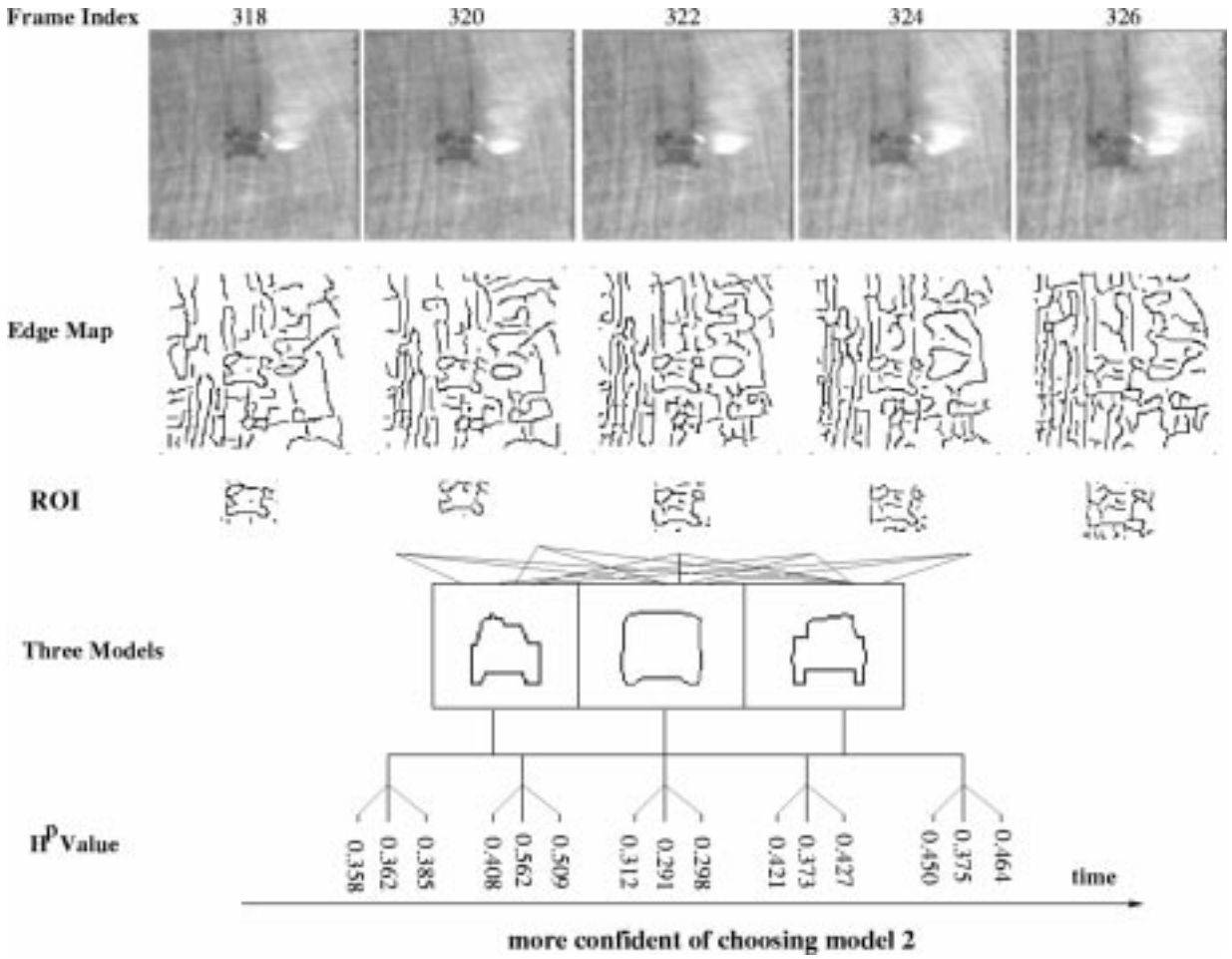

Fig. 5. Dynamic identification with the H^p metric: at each frame the models are compared against the detected ROI according to (6), and the H^p values are used to choose one out of the three hypotheses.

estimate the scaling from a model to the scene using the size of the ROI (this step, however, needs to account for the inaccuracies in the segmentation step). Another example is, when attempting verification, if the sensor-induced motion is dominant, the affine parameters computed from (7) can be used to estimate the scale factor s between two ROI's in corresponding frames by

$$s = \sqrt{(r_{11}^2 + r_{12}^2 + r_{21}^2 + r_{22}^2)/2} \tag{8}$$

as will be illustrated by an experiment in Section IV.

C. Pose Estimation and Object Identification

The segmentation step in the previous section locates potential moving objects. If the object is subject to nearly translational rigid motion, we can use the average of the flow field in the segmented area to approximate the object velocity $\mathbf{V}$. If the object is too small to support the average computation, an alternative way is to estimate the velocity $\mathbf{V}$ from the change in the mass center of the detected changing area. The details of the algorithm can be found in [17]. For the pose estimation algorithm, only the direction of $\mathbf{V}$ is used to assist search over the model base, although the value of $\mathbf{V}$ is potentially usable.

With $\mathbf{V} = (\mathbf{V}_x, \mathbf{V}_y)$ plotted in a regular X–Y coordinate system [with horizontal X-axis, vertical Y-axis and $(1, 1)$ lying in the upper-right quadrant], we can easily identify the constraints on the angles α, ϕ imposed by the signs of the components of $\mathbf{V}$. For example, consider a forward moving object. Assuming $\phi \in [0°, 90°]$, we have

if $\phi = 0°$, then $\mathbf{V}_x < 0 \Rightarrow \alpha \in (0°, 180°)$;
if $\phi = 0°$, then $\mathbf{V}_x > 0 \Rightarrow \alpha \in (180°, 360°)$;
if $\phi = 90°$, then $\mathbf{V}$ totally determines the object orientation in a top view;
if $\phi \in (0°, 90°)$, then $\mathbf{V}_x > 0$ and $\mathbf{V}_y > 0 \Rightarrow \alpha \in (180°, 270°)$, etc.

One can find that constraints of the last type are most effective, and also most common in real applications. For example, a typical surveillance camera may have ϕ between 0° and 90°. Under the constraint provided by motion analysis, pose estimation and object identification problems are reduced to the following search problem: given an ROI, find the best matching model from only the model images whose orientations (α, ϕ) lie in the subspace $\mathcal{A} \times \Phi$ with $\mathcal{A} = (\alpha_1, \alpha_2)$ and $\Phi = (\phi_1, \phi_2)$ being two intervals, and α_1 and α_2 being estimated as above. The values of ϕ_1 and ϕ_2 are application-dependent. In the experiments reported in this paper, we let $\Phi = (0, 90)$. Formally,

by using (9), we estimate the object pose for the current frame as

$$(\hat{\alpha}, \hat{\phi}) = \underset{\alpha, \phi \in \mathcal{A} \times \Phi}{\operatorname{argmin}} H^p(R, M(\alpha, \phi)) \tag{9}$$

where R is defined as before, and $M(\alpha, \phi)$ is the model with orientation parameter (α, ϕ) [note that in practice, a local search using (6) is still needed to account for the inaccuracy of the segmentation step]. Obviously, (6) and (9) involve essentially similar computations except that we use the former to choose the object type and the latter to decide the orientation. In fact, if we treat the same object at different poses as different classes, then (9) is implied by (6).

The benefit from the constrained search is twofold: the search is speeded up; and more importantly, by reducing the number of candidates, the probability of false match is reduced. Further constraints can be obtained if we consider the relationship among frames which are temporally close. In this work, we exploit this type of information through what we refer to as the *pose evolution curve* as discussed in Section III-E.

Note that, in the worst case where motion analysis gives totally false information for, the constraints obtained above are no longer valid. To deal with this situation, the algorithm should resort to basic full search whenever the confidence measure (see Section III-F) of the current estimate drops below a threshold.

D. Excluding Clutter in the ROI

The detected ROI contains not only the potential object but also background clutter. According to (5), every edge pixel within the ROI could contribute to $H^p(A, B)$, which is undesirable. When doing identification, the following technique is used to exclude clutter before calculating H^p: given a model M and an ROI R, we keep points in the ROI only if they are within a certain distance of $T(M)$. Here $T(M)$ is a transformed version of M under transformation T. That is, a new ROI R' is formed by

$$R' = \{x: \forall x \in R \text{ and } \rho(x, T(M)) < t\} \tag{10}$$

where t is a small positive number. On the other hand, if we are attempting verification, the model is typically an ROI from previous frames. In this situation, the segmentation boundary estimated in Section III-B will also be used to constrain R', and a larger t should be used to account for the inaccuracies in the segmentation.

E. Pose Evolution Curve

By plotting the estimated pose over time, we get the *pose evolution curve* for the object. The curve is an additional indicator of identification/verification confidence: under the continuous motion assumption, the pose evolution curves should display some smoothness in either the $\mathcal{A}$ or Φ domain. For example, if the pose evolution curve suffers from random jitter between adjacent frames, chances are that the object hypothesis is wrong to begin with. Of course, when objects are similar in shape, there may not be enough information from the pose evolution curves only, but the corresponding H^p value should offer information

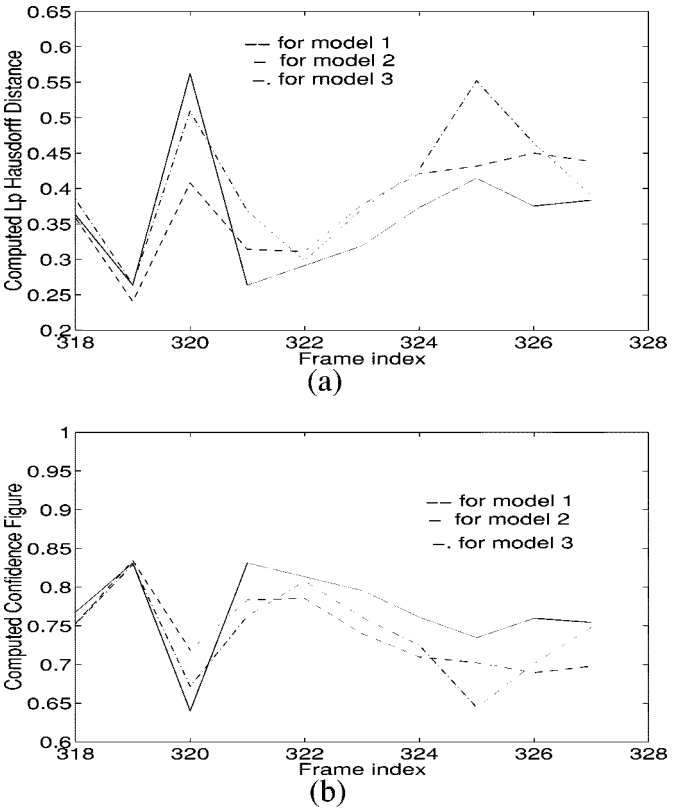

Fig. 6. H^p values and confidence figures for the sequence in Fig. 5, from frame 318 to 327: (a) H^p values versus frame index and (b) computed confidence figure P_c.

for identification purposes. The following quantity S is defined to give a quantitative description of the smoothness of the pose evolution curve:

$$S = \frac{1}{N} \sum_t (p(t) - q(t))^2 \tag{11}$$

where

N	number of frames;
t	frame index;
$p(t)$	pose evolution curve;
$q(t) = p(t) * window(t)$	smoothed version of $p(t)$ ($*$ denotes convolution).

Here, $window(t)$ is a discrete window function whose support is $[-I, I]$, with I a small positive integer. In the experiments in this paper, $I = 1$ and $window(-1) = 0.25$, $window(0) = 0.5$, $window(1) = 0.25$. In general, given a sequence, a correct hypothesis should yield a pose evolution curve with a smaller S than an incorrect hypothesis does.

When the number of model images is large, it is helpful to also consider several top matches given by (9), instead of only looking at the best matching one. If we keep several top matches and plot the estimated poses in a common coordinate system at each frame, we get the band of pose estimates. This pose information is also helpful for testing the hypotheses; if a hypothesis is correct, the corresponding pose band should be more concentrated than that of a wrong hypothesis, unless the object's appearance changes dramatically even with small changes in orientation. Again, to quantitatively evaluate how a pose band is

Fig. 7. Sample images in the model base. The upper row is for model 1; the lower row for model 2.

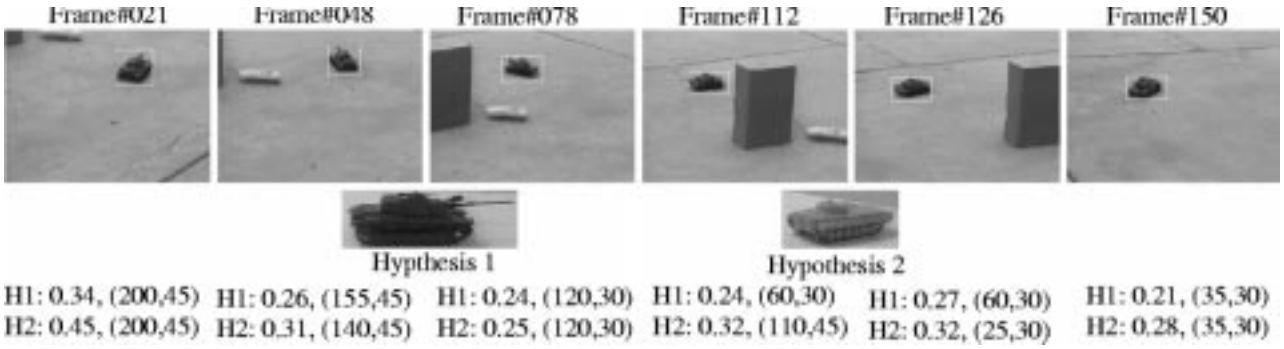

Fig. 8. Dynamic identification with the H^p metric. The task is to identify the moving object in the scene as one of the two hypotheses. Top row: sample images (of size 320×240) from one sequence with black tank moving around. The moving object has been highlighted by a bounding box defined at the segmentation step. Middle row: two hypotheses. Bottom row: corresponding H^p values and estimated poses (α, ϕ) at each frame for the two hypotheses. Even though the two hypotheses are similar in shape, at each frame, the algorithm gives smaller H^p values for hypothesis 1 than for hypothesis 2.

concentrated, the following quantity C is defined, if the top B matches are kept:

$$C = \frac{1}{N} \sum_t \left\{ \frac{1}{B} \sum_{i=1}^{B} (p_i(t) - \overline{p}(t))^2 \right\} \tag{12}$$

where N, t are as before, and $\overline{p}(t)$ is the average of the pose angles given by the top B matches at frame t. In general, given a sequence, a correct hypothesis should yield a band of poses with smaller C than an incorrect hypothesis does. The motivation behind the above definitions is that S and C are in a sense like sample mean square deviations.

Notice that the pose angle α is periodic, i.e., 360° is equivalent to 0°. Therefore when calculating the average of angles [the moving average for $q(t)$ in (11) and the average over the B angles in (12)], the value of $p(t)$ should be wrapped around with respect to 360° whenever the periodicity demands it.

F. Interpreting H^p as a Confidence Measure

As mentioned earlier, H^p has some nice properties such as being a metric, improved robustness, "expected risk" interpretation, etc. Being a metric is important especially from a theoretical point of view. For example, we would like a measure $(\cdot, \cdot)$ that gives the same result for (X_1, X_2) and (X_2, X_1). These properties of H^p allow a confidence interpretation. For the identification problem, this means that at each step the H^p value for each model is treated as a measure of *confidence* in choosing a certain model: the smaller this number is, the more confident we are of choosing the model. If multiple models are kept as frames are processed, although at some time we may make the wrong choice, subsequent updates will hopefully provide the right choice.

For the verification problem, a confidence interpretation is also helpful: whenever there is a sharp decrease in confidence, what may have happened is that the object is no longer the previous one, *or* the orientation of the object has changed dramatically. This information can be used to update the model hypotheses. Verification, in this respect, is similar to a tracking problem like that in [8]. But keeping a 3-D model of the object allows one to tackle problems involving more general complex 3-D motions.

To conform with the common understanding of *confidence*, i.e., with real numbers 1.0 and 0.0 representing the most and least confidences respectively, we define a confidence figure based on the H^p value as

$$P_c = 1 - \frac{H^p}{1 + H^p}. \tag{13}$$

The meaning is obvious: when $H^p(A, B) = 0.0$, which means $A = B$, we set $P_c = 1.0$; and when $H^p(A, B)$ goes to infinity, $P_c = 0.0$. The underlying reason for choosing function $f(x) = x/(1 + x)$ is that it is a convex function on $[0, \infty)$ and thus

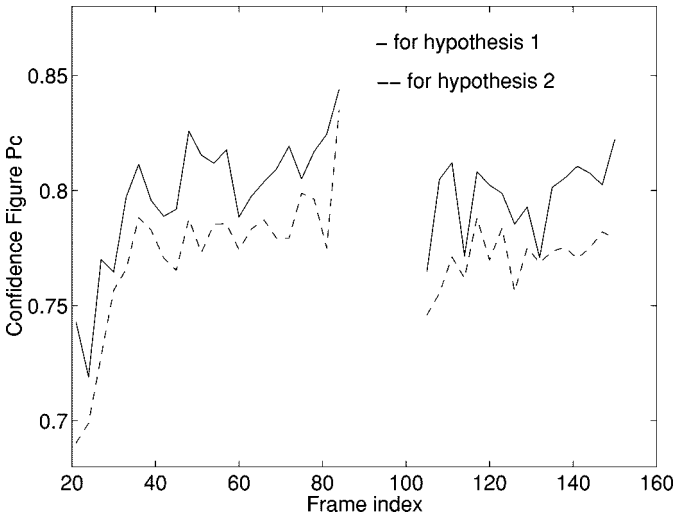

Fig. 9. Confidence figure P_c versus frame index (computed every three frames; note that nothing is shown for those frames in which the object is invisible; same for Figs. 10 and 11).

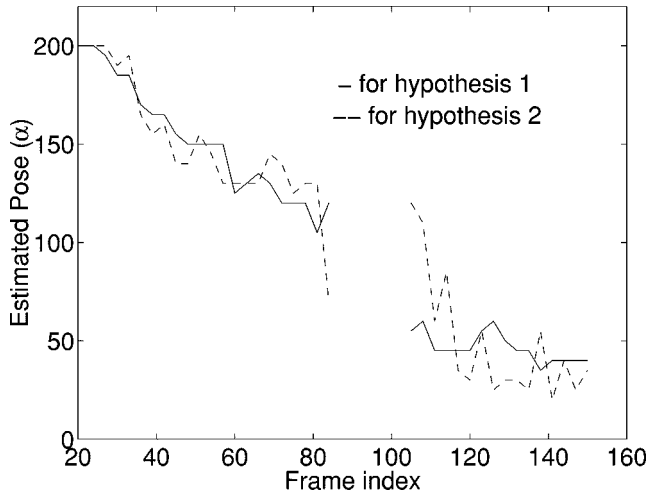

Fig. 10. Pose evolution curves (angle α) for the sequence shown in Fig. 8. The computed S values [see (11)] are 8.1 and 64.9 for hypothesis 1 and hypothesis 2, respectively, strengthening the confidence in choosing hypothesis 1.

$f \circ H^p$ is still a metric, which is desirable for comparing multiple hypotheses. It is worth pointing out that this confidence figure depends on the measurement of H^p, thus it will change if H^p is reparameterized (for instance, if the resolution of the image is doubled). Therefore, it is better to use this figure for comparing different hypotheses than to interpret it for a single hypothesis along the temporal axis unless the size of the supporting X is fixed for all frames.

In summary, the whole framework is illustrated by the diagram shown in Fig. 3. It is worth pointing out that we have assumed that the object is moving and that detection and segmentation are based on analyzing the object motion. If the object is stationary and/or the detection and segmentation are accomplished by other means, the verification step still works (in this case, pose variation may still exist due to sensor motion). We have used the dotted-lined path to show that possibility although it is not implemented in the current work. Notice that, if no pose variations are considered, then the pose estimation

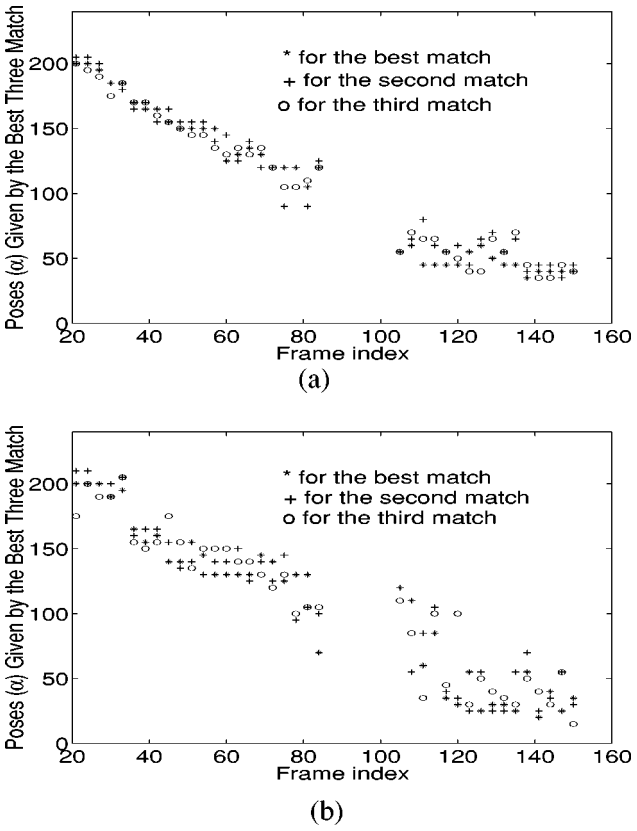

Fig. 11. Bands of the pose estimates (angle α) versus frame index. (a) Hypothesis 1 and (b) hypothesis 2. The computed C values [see (12)] are 20.9 and 80.4 for hypothesis 1 and hypothesis 2, respectively, strengthening the confidence in choosing hypothesis 1.

module would not be used. This is equivalent to choosing (6) or (9) based on the specific problem.

IV. Experimental Results and Analysis

Experiments have been performed with both infrared and optical sequences. Several experimental results are presented in this section. For convenience of presentation, the detailed discussion of the results is mainly focused on two sequences, one infrared and one optical. Experiments on other data will be briefly listed. In the experiments presented in this paper, p and c in (5) are fixed at 1 and 4 respectively, and t in (10) is 5. The edge maps were detected using Canny's algorithm [4].

In the first experiment, simple verification is carried out on an infrared sequence acquired by a helicopter flying toward a tank. Due to the fast camera motion, the scaling effect becomes significant within a few frames. Fig. 4 illustrates the verification procedure. A moving object is first detected, which is possible only after the dominant sensor motion is compensated. Then an ROI is formed and processed to get the edge map of the object. This edge map, used as the model, is verified in subsequent frames. In Fig. 4, the ROI from frame 302 is superimposed on frames 310, 315, and 320, after the locations have been estimated using (6). Note the substantial scaling of the object—a typical scale factor is 1.02 by (8) for two consecutive frames, thus the object gets almost 1.4 times larger in frame 320 than when it was in frame 302 (this is only during a period slightly more than half a second with a 30 Hz frame rate). It would be

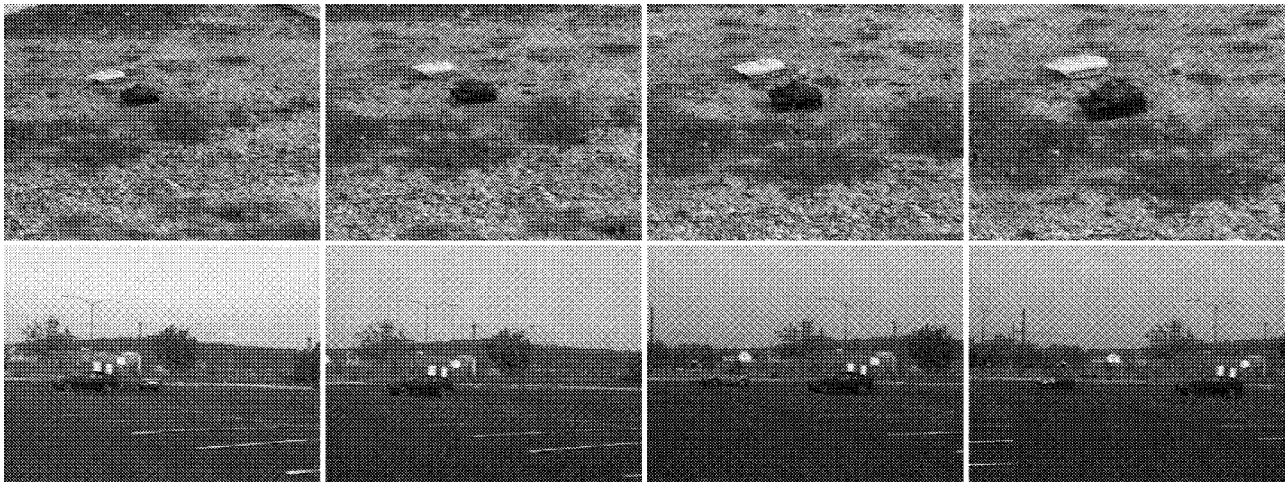

Fig. 12. Each row contains sample images from an optical sequence. The first sequence is tested with the modelbase given by Fig. 7. The second sequence is tested with models containing two cars, one of which being the moving car in the sequence, and rotation is allowed horizontally only.

very expensive if one wanted to handle this by searching over the scale space using the Hausdorff metric. However, by compensating the scaling using (8) the algorithm is able to locate the tank and report small H^p values (meaning high confidence).

In the above example, the object motion is approximately 2-D; thus verification is similar to a tracking problem after the object has been detected. However, if the object is lost somehow (e.g., due to occlusion), then re-appears later, the algorithm should be able to verify if it is the previous object. The idea becomes more obvious when the object motion is 3-D and induces dramatic orientation changes. In this situation, even if simple tracking can be done on a frame-by-frame basis, it is hard to say if it has found the same object because the object looks too different in later frames than in earlier frames. What might have happened is that the tracker has drifted away. However, with a model in mind, verification can still be done by updating the model according to the motion trajectory information: if the confidence figure for certain model at the current frame drops suddenly, but a new pose of the model (either predicted by the trajectory or by a full search in pose domain) gives high confidence, then the new pose will be kept and the present object is verified to be the previous one. This is represented by the feedback path from verification to pose estimation in Fig. 3.

Fig. 5 shows how identification works for the same sequence as in Fig. 4. Given the sequence, the task is to identify which of the three hypotheses is present in the current sequence. In this example the ground truth is model 2. In this experiment, object contours derived from CAD models are used as models. For each frame, an ROI is detected, then each model is warped to the size of the ROI. To account for inaccuracies in the detection and segmentation step, for each model, a local search in translation space and in scale space is carried out according to (6), and the best H^p value is used for this model. It is clear from the figure that, although in some individual frames the algorithm reports false identification results (i.e., the H^p value for model 2 is bigger than those for the others), by using multiple frames, the overall confidence of choosing model 2 is higher than for choosing others. To see this more clearly, we plotted in Fig. 6 the H^P value and the confidence figure for each model from frames 318 to 327. The overall confidence in model 2 is obvious.

In the next experiment, the sequences are optical images acquired by a hand-held video camera. Model images were similarly acquired, with the help of a turntable. The model base was constructed as explained in Section III-A. Currently, the model base contains two model tanks. Fig. 7 shows four model images for each of the models. Accurate pose information can be obtained for the model images if the camera is under fine control, as in [11]. In this work, the model pose was obtained manually through visual inspection of the model image. The Φ domain is only coarsely divided, with ϕ taking only two values, 30° and 45°; and for each ϕ, α varies by approximately 5° in [0°, 355°] (see Fig. 2 for the definitions of the angles).

Fig. 8 illustrates how identification works when the object is subject to 3-D motion. The sequence contains two objects, and only the black one is moving. The task is to identify the moving object as one of the two hypotheses (i.e., the object needs to be identified as one of the two candidate models). Of course, in this example, the ground truth is model 1 (the black object). The moving object is first detected and segmented by the aforementioned procedure; then identification and pose estimation are carried out in each frame. The first row in Fig. 8 shows sample frames from one of the sequences with the detected moving object highlighted by a bounding box (note that only the black object is moving in this sequence). Below each frame are the corresponding H^p value and the estimated pose (two angles α and ϕ) computed according to (9). In the computation, again a local search (of size 3×3 pixels) in the translation space was performed. Although no search in scale space was performed in this example, the result is good enough, implying that the segmentation step gives a good estimate of the scale factor between the model and the scene object (a search over $s = 0.9, 1.0, 1.1$ is used in other experiments in the paper). It is obvious that, although the appearances of the objects are dramatically different, they have similar geometric shapes. Therefore it is in fact a difficult task to distinguish between these two hypotheses only from their edge maps. Yet in each frame, the algorithm is able to choose model 1 correctly, and the pose es-

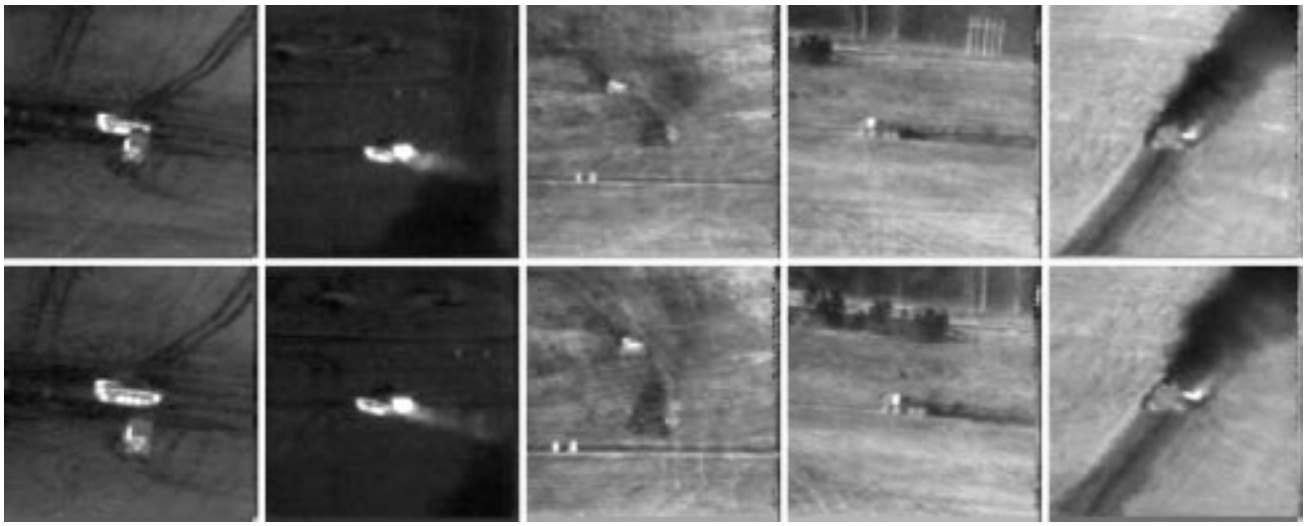

Fig. 13. Each column contains sample images from an IR sequence. The sequences are named as $rng16\_18$, $rng17\_20$, $rng19\_06$, $rng19\_07$, and $rng19\_18$, from left to right, respectively. Three hypotheses are assumed during verification. The object motion is assumed to be 2-D (no 3-D model of the objects is available in this case).

TABLE I
PERFORMANCE OF THE ALGORITHM ON SOME TEST SEQUENCES

Sequence Name	Number of Hypotheses Assumed	Number of Frames Used for Verification	Number of Correct Frames By H^0	Correct By S?	Correct By C?
two_tank1	2	100	91	Yes	Yes
two_car3	2	120	107	Yes	Yes
rng16_08	3	25	19	-	-
rng17_01	3	25	21	-	-
rng19_06	3	25	22	-	-
rng19_07	3	25	20	-	-
rng19_18	3	25	21	-	-

timates are close enough (see Fig. 8 for the estimated poses for these sample frames).

Fig. 9 shows the confidence figure computed according to (13) By plotting the estimated pose in each frame, we can get the pose evolution curve for each hypothesis, as shown in Fig. 10. The pose evolution curve is found to be able to strengthen the confidence since the curve for hypothesis 1 is smoother than that for hypothesis 2, which is reasonable under the assumption of continuous motion. According to (11), the computed S values are 8.1 and 64.9 for hypothesis 1 and hypothesis 2, respectively. Note that for the aforementioned reason, the ϕ value is not informative in this experiment; thus only angle α is plotted in the pose evolution curve.

As mentioned before, when the number of model images is large, it is helpful to keep several top matches given by (9), and consider the band of pose estimates. For the sample sequence in Fig. 8, the pose bands are plotted in Fig. 11 (for angle α only). This pose information is illustrative: the band for hypothesis 1 is more concentrated than that for hypothesis 2, thus strengthening our confidence in choosing hypothesis 1. According to (11), the computed C values are 20.9 and 80.4 for hypothesis 1 and hypothesis 2, respectively.

We now briefly list the experimental results for other sequences, including optical and infrared imagery. Fig. 12 shows two optical sequences in which the objects are subject to 3-D motion. The first sequence (two_tank1) was tested with the model base shown in Fig. 7. The second sequence (two_car3) was tested with two models containing two cars at different horizontal orientations, one of which being the moving car in the sequence. Fig. 13 shows sample sequences from a large FLIR database. Each of these sequences was tested with three hypotheses which are edge maps from other sequences in the database, with one being true hypotheses. The experimental results are summarized in Table I. Since each sequence is relatively long and the object is only big enough near the terminal portion of the sequence, we used only the final few frames in verification, as shown in the column "number of frames used." Note that there is no 3-D model available for the sequences, and the object motion is largely 2-D in the last few frames. Thus verification was done by using (6), and no S and C values were computed. For sequences in Fig. 12, (9) was used, and corresponding S and C values were computed. From the table, it is obvious that although at some frames the algorithm may be confused, there is not a single case where the algorithm fails if all the frames are considered. Specifically, S and C values alone are able to give correct results when they are available.

V. SUMMARY

Object recognition is a well-researched area, and has been approached from different aspects (e.g., [2], [12]). In this paper,

we attempted to exploit temporal information in video to assist recognition. Specifically, in addition to using motion for detection and segmentation, our approach uses the trajectory of a moving object to predict its approximate pose. Also, the motion coherence of an underlying object is measured through the pose evolution curses. The experiments demonstrate how the concepts and algorithms for dynamic identification/verification work on real video. The paper is focused on studying the potential of the idea of temporal object verification using video. The work reported here is intended to show how temporal information can be exploited to help verification/identification when video is available, rather than to build an end-to-end system. When building an end-to-end system, many system-level issues arise, such as time and space complexity, which are not addressed in this paper.

In this work, although feature (edge) detection is necessary, we use the L_p version of Hausdorff metric-based matching, which does not require feature correspondence and is robust to noise, hence relaxing the requirements on the feature detection step. This may be attractive especially in applications that use infrared sensor, since in this situation, feature-detection is not reliable, and appearance-based approaches may also have difficulties. However, the work is not intended to replace other methods such as appearance-based recognition approaches. In fact, our major focus has been on how to utilize temporal information available in a video for better recognition. Some basic ideas developed in the work are meaningful irrespective of the specific representation of the object, such as pose prediction based on the object's trajectory, and computing motion coherence based on pose evolution curves, etc. Also, the approach is proposed for a more constrained scenario: verification and identification (with much fewer hypotheses than a recognition algorithm usually assumes), and simple object motion has been assumed. Even though this is a constrained scenario, similar situations can be found in many applications such as surveillance systems. We have found the proposed method promising for verification/identification tasks.

ACKNOWLEDGMENT

The authors would like to thank Dr. R. Sims for providing the FLIR data.

REFERENCES

[1] A. J. Baddeley, "Errors in binary images and an L_p version of the Hausdorff metric," *Nieuw Archief voor Wiskunde*, vol. 10, pp. 157–183, 1992.
[2] R. Basri and D. Jacobs, "Recognition using region correspondence," in *Proc. Int. Conf. Computer Vision*, 1995.
[3] G. Borgefors, "Distance transformations in digital images," in *Comput. Vis., Graph., Image Process.*, vol. 34, 1986, pp. 344–371.
[4] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-8, pp. 679–698, 1986.
[5] D. Doria and D. Huttenlocher, "Progress on the fast adaptive target detection program," *RSTA Tech. Rep. ARPA IU Program*, pp. 589–594, 1996.
[6] H. Federer, *Geometric Measure Theory*. Berlin, Germany: Springer-Verlag, 1967.
[7] F. Hausdorff, *Set Theory*, 2nd ed. New York: Chelsea, 1962.
[8] D. Huttenlocher, J. Noh, and W. Rucklidge, "Tracking nonrigid objects in complex scenes," in *Proc. Int. Conf. Comput. Vis.*, Berlin, Germany, 1993, pp. 93–101.
[9] B. Li, R. Chellappa, Q. Zheng, and S. Der, "Experimental evaluation of neural, statistical and model-based approaches to FLIR ATR," *Proc. SPIE*, vol. 3371, pp. 388–397, 1998.
[10] B. Li, Q. Zheng, and S. Der, "Moving object detection and tracking in FLIR images acquired by a looming platform," in *Proc. Joint Conf. Information Sciences*, Research Triangle Park, NC, 1998, pp. 319–322.
[11] H. Murase and S. Nayar, "Visual learning and recognition of 3-D objects from appearance," *Int. J. Comput. Vis.*, vol. 14, pp. 5–24, 1995.
[12] R. P. N. Rao, "Dynamic appearance-based recognition," in *Proc. IEEE Conf. Computer Vision Pattern Recognition*, 1997, pp. 540–546.
[13] S. Srinivasan and R. Chellappa, "Image stabilization and mosaicking using the overlapped basis optical flow field," in *Proc. IEEE Int. Conf. Image Processing*, Santa Barbara, CA, 1997.
[14] M. Tarr and S. Pinker, "Mental rotation and orientation-dependence in shape recognition," *Cogn. Psychol.*, vol. 21, pp. 233–282, 1989.
[15] S. Ullman, *High-Level Vision*. Cambridge, MA: MIT Press, 1996.
[16] I. Weiss, "Geometric invariants and object recognition," *Int. J. Comput. Vis.*, vol. 10, pp. 207–231, 1993.
[17] Q. Zheng and R. Chellappa, "Motion detection in image sequences acquired from a moving platform," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, Minneapolis, MN, 1993, pp. 201–204.
[18] A. Zisserman, D. Forsyth, J. Mundy, C. Rothwell, J. Liu, and N. Pillow, "3D object recognition using invariance," *Artif. Intell.*, vol. 78, pp. 239–288, 1995.

Baoxin Li received the B.S. and M.S. degrees in electrical engineering from the University of Science and Technology, China, in 1992 and 1995, respectively. He received the Ph.D. degree in electrical engineering from the University of Maryland, College Park, in 2000.

He is currently with Sharp Laboratories of America, Inc., Camas, WA, working on multimedia analysis for consumer applications. He was previously with the Center for Automation Research, University of Maryland, working on face and object tracking and verification in video, automatic target recognition, and neural networks. His interests include pattern recognition, computer vision, neural networks, and multimedia processing.

Rama Chellappa (S'78–M'79–SM'83–F'92) received the B.E. (Hons.) degree from the University of Madras, Madras, India, in 1975 and the M.E. (Distinction) degree from the Indian Institute of Science, Bangalore, in 1977. He received the M.S.E.E. and Ph.D. degrees in electrical engineering from Purdue University, West Lafayette, IN, in 1978 and 1981, respectively.

Since 1991, he has been a Professor of electrical engineering and an Affiliate Professor of computer science with the University of Maryland, College Park. He is also an Associate Director with the Center for Automation Research and is also affiliated with the Institute for Advanced Computer Studies. Prior to joining the University of Maryland, he was an Associate Professor and Director of the Signal and Image Processing Institute with the University of Southern California, Los Angeles. During the last 20 years, he has published numerous book chapters and peer-reviewed journal and conference papers. Several of his journal papers have been reproduced in collected works published by IEEE Press, IEEE Computer Society Press, and MIT Press. He has edited a collection of papers on *Digital Image Processing* (Santa Clara, CA: IEEE Computer Society Press), co-authored a research monograph on *Artificial Neural Networks for Computer Vision* (with Y. T. Zhou) (Berlin, Germany: Springer-Verlag), and co-edited a book on *Markov Random Fields* (with A. K. Jain) (New York: Academic). He also served as Co-Editor-in-Chief of *Graphical Models and Image Processing*. His current research interests are

image compression, automatic target recognition from stationary and moving platforms, surveillance and monitoring, automatic design of vision algorithms, synthetic aperture radar image understanding, and commercial applications of image processing and understanding.

Dr. Chellappa has served as an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON IMAGE PROCESSING, and IEEE TRANSACTIONS ON NEURAL NETWORKS. He served as a member of the IEEE Signal Processing Society Board of Governors from 1996 to 1999. He is currently serving as the Editor-in-Chief of IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE. He has received several awards, including the 1985 NSF Presidential Young Investigator Award, a 1985 IBM Faculty Development Award, the 1991 Excellence in Teaching Award from the School of Engineering, University of Southern California, and the 1992 Best Industry Related Paper Award from the International Association of Pattern Recognition (with Q. Zheng). He has been recently elected as a distinguished Faculty Research Fellow (1996–1998) at the University of Maryland. He is a Fellow of the International Association for Pattern Recognition. He has served as a General and Technical Program Chair for several IEEE international and national conferences and workshops.

Qinfen Zheng received the B.S. and M.S. degrees in electrical engineering from the University of Science and Technology, China, in 1981 and 1984, respectively. He received the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, in 1992.

From 1992 to 1994, he was an Assistant Research Scientist with the Center for Automation Research, University of Maryland, College Park. During 1994–1995, he was a Scientist with the Lockheed Martin Laboratory, Baltimore, MD. Since 1996, he has been an Associate Research Scientist with the Center for Automation Research, University of Maryland. His research interests include image and video analysis, automatic target detection/recognition, human identification, motion analysis, and remote sensing.

Sandor Z. Der received the B.S. and M.S. degrees in electrical engineering from Virginia Polytechnic Institute, Blacksburg, in 1986 and 1988, respectively. He received the Ph.D. degree in electrical engineering from the University of Maryland, College Park, in 1995.

He is currently with the U.S. Army Research Laboratory, Adelphi, MD, working on image exploitation, including automatic target recognition and sensor modeling. He was previously with the U.S. Army Night Vision and Electronic Sensors Directorate, working on synthetic image generation, automatic target recognition, and sensor simulation. His interests include pattern recognition, computer vision, neural networks, and optical sensors.