Effective Tracking through Tree-Search

Daniel Freedman, *Member*, *IEEE*

Abstract—A new contour tracking algorithm is presented. Tracking is posed as a matching problem between curves constructed out of edges in the image, and some shape space describing the class of objects of interest. The main contributions of the paper are to present an algorithm which solves this problem accurately and efficiently, in a provable manner. In particular, the algorithm's efficiency derives from a novel tree-search algorithm through the shape space, which allows for much of the shape space to be explored with very little effort. This latter property makes the algorithm effective in highly cluttered scenes, as is demonstrated in an experimental comparison with a condensation tracker.

Index Terms—Contour tracking, tree-search, hybrid optimization, approximation algorithm, compact manifold.

1 Introduction

This paper is concerned with a new approach to contour tracking, which is referred to as the *manifold tracker*. This approach matches curves in the image constructed from edge-points, to a shape space, a set of curves which describe the object of interest. The latter is assumed to be known prior to the running of the algorithm; it might either be learned or postulated based on simple assumptions. The key contributions of this paper are threefold. First, an algorithm is presented for performing this matching in a treelike, or coarse-to-fine manner, in such a way that the entirety of shape space is explored in an efficient manner. Second, theoretical bounds are given, showing that this algorithm leads to near-optimal matching. Finally, through an experimental comparison with the condensation algorithm, it is shown that this ability to efficiently search through shape space can be critical for actual tracking.

Let us pose the tracking problem in order to clarify some of the ideas expressed above. Suppose $\mathbf{E}$ is the set of curves which can be constructed from edge-points in the image, and $\mathbf{C}$ is the shape space. (The construction and properties of these sets will be described in greater detail in Section 2.1.) Then, a natural problem to solve is

$$\min_{\mathbf{e}\in\mathbf{E},\mathbf{c}\in\mathbf{C}} \|\mathbf{e}-\mathbf{c}\|, \tag{1}$$

where $\|\cdot\|$ is the L_2 norm. The minimizing argument, $\mathbf{e}^*$, is taken to be the tracked curve. The main thrust of the paper will to be present an algorithm which will allow for the above minimization problem to be solved both efficiently and accurately. Efficiency is achieved through coarse-to-fine, treelike search through the set $\mathbf{C}$; accuracy is proven via an upper bound on the amount by which the value of the solution generated by the algorithm can differ from the value of the optimal solution.

How can such an algorithm be compared with condensation? Condensation is a stochastic algorithm involving dynamical considerations; by contrast, manifold tracking is deterministic and does not make use of dynamics. Nonetheless, the manifold tracking formulation may be viewed dynamically. In particular, the stochastic dynamical model corresponding to the formulation in (1) is of an object with a uniform probability density over the shape space $\mathbf{C}$. Most often, the shape space for a particular frame is "centered" around the previous frame's estimate; thus, the set $\mathbf{C}$ actually contains a hidden (though simple) dynamical assumption. In consequence of this fact, the minimization problem looks quite similar to a condensation problem (where the observation model is the standard one employed in [4]).

Bearing this in mind, the main distinction between condensation and manifold tracking can now be drawn. Condensation simply fixes the number of samples of $\mathbf{C}$ that it will examine; new samples are generated stochastically each frame, and are compared with the edges in the image via the observation model. By contrast, the manifold tracker is able, by examining the same number of samples as condensation, to *effectively* look at many more samples. This is because of the tree approach. The algorithm uses a strategy which allows for branches of the tree to be pruned when it is guaranteed that the true solution cannot lie in these branches. In practice, this pruning is generally extremely successful, and allows for many more effective samples to be examined. As a result, the manifold tracker allows for much greater accuracy than condensation, using the same number of operations. As has been noted, this accuracy can sometimes be extremely important; in highly cluttered scenes, it can be the difference between maintaining and losing lock. (This assertion will be justified in Section 7.)

The outline of this paper is as follows: The remainder of this section reviews the literature on contour tracking. Section 2 presents a mathematical overview of the problem, including formal properties of the sets $\mathbf{E}$ and $\mathbf{C}$, as well as challenges inherent in the problem. Section 3 presents the algorithm itself; this section is somewhat long, as it is necessary to introduce a certain amount of notation in order

• *The author is with Rensselaer Polytechnic Institute, Department of Computer Science, Troy, NY, 12180-3590. E-mail: freedman@cs.rpi.edu.*

Manuscript received 28 Mar. 2001; revised 12 Apr. 2002; accepted 23 Sept. 2002.
Recommended for acceptance by P. Anandan.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 113902.

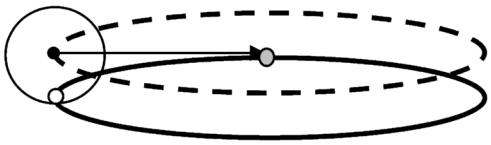

Fig. 1. Edge search takes place in circular regions. The dashed curve represents the previous frame's contour (from which search emanates), the solid curve the contour in the current frame. Although the aperture problem dictates search normal to the dashed contour, it is clear that this is ineffective when the motion is not infinitesimal. The reason is that in areas of high curvature, normal search results in the gray point, which is quite far away from the correct point on the solid contour, whereas with circular search, the correct (white) point is detected.

to describe the algorithm. Section 4 states two theorems about the algorithm: the first concerns its accuracy in solving the problem in (1), and the second, its efficiency. Section 5 proves the two theorems. Section 6 establishes the complexity of the algorithm. Finally, Section 7 shows the results of three experimental comparisons with the condensation tracker.

1.1 Review of Existing Literature

Contour tracking algorithms present several interesting applications. These include the surveillance of individuals [19], [33], [30], [23], [27], [2], biomedical image analysis [25], [3], [18], [1], [15], [28], audio-visual recognition and enhancement of degraded speech [9], [14], [26], [22], [13], [21], [24], [17], [16], and guidance of autonomous vehicles [34], [31], [32].

There are a number of existing approaches to the problem of contour tracking. The deformable template approach [36], [25] involves finding a model parameterization for the contours of the object to be tracked, and matching this representation with successive images in the video stream in order to detect the contours of interest. The active contour approach [20], [35], [11], [12] also features an energy minimization problem, in which the energy is a functional of the entire curve. Minimizing the energy leads to smooth curves which are attracted to edges in the image. A third set of algorithms includes both the Kalman tracker [5], [7], [10] and its successor, the condensation algorithm [6]. This approach specifies a stochastic model, consisting of a dynamical model for the curves, as well as an observation model describing the effect of noise (due to clutter, etc.) on the observed edges. Observations and dynamics are combined to yield the optimal estimation for the curve's location. The condensation tracker is quite popular in the literature, and is therefore the natural choice for comparison purposes.

2 Outline of the Problem

The tracking problem has already been posed, in (1), as the solution to

$$\min_{\mathbf{e}\in\mathbf{E},\mathbf{c}\in\mathbf{C}} \|\mathbf{e}-\mathbf{c}\|.$$

The tracked curve is given by

$$\mathbf{e}^* = \operatorname*{argmin}_{\mathbf{e}\in\mathbf{E}} \left[\min_{\mathbf{c}\in\mathbf{C}} \|\mathbf{e}-\mathbf{c}\| \right].$$

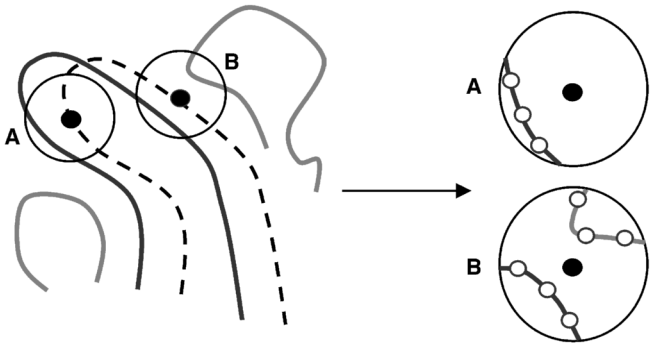

Fig. 2. Construction of an observed curve $\mathbf{e}$. The detected contour of the previous frame is marked in dashed lines. The outlines of objects in the current frame are marked in solid lines. The object of interest is black, the gray objects represent clutter. Edge-detection at two sites, A and B, is shown, the edge-points are shown at the right as white circles (a finite number, due to pixelization). At site B, edge-points representing both the object of interest as well as clutter are detected.

An informal description of the sets $\mathbf{E}$ and $\mathbf{C}$ was given in Section 1; we shall now give a more formal treatment.

2.1 The Sets $\mathbf{E}$ and $\mathbf{C}$

$\mathbf{E}$, the set of curves constructed from edge-points in the image, is referred to as the *observed set*, and is generated as follows: At N equally spaced points along the detected contour of the previous frame, edge-search takes place in circular regions, in the image of the *current* frame. (Note that many algorithms, for example, [5] use normal search due to the aperture problem; however, the aperture problem is truly relevant only in the case of infinitesimal motions between frames. The problem with using normal edge-search in real video-sequences is that many edge-points may be missed, particularly in regions of high curvature. This is illustrated in Fig. 1.) Each of these points are denoted *sites*. In each of these regions, a number of edge-points are detected; denote the set of edge-points detected in the $n\text{th}$ region by E_n. An element $\mathbf{e}\in\mathbf{E}$ may then be constructed as follows, see Fig. 2.

1. Take one edge-point $e_n \in E_n$ from each region $n = 1,\ldots,N$.
2. Smoothly interpolate the set of edge-points $e_1,\ldots,e_N$ into a curve $\mathbf{e}$. The method of interpolation is largely unimportant, as will be discussed in Section 2.2.

Thus, the set $\mathbf{E}$ is in one-to-one correspondence with the set $E \equiv E_1 \times \times E_N$. Suppose there are M edge-points detected per site, i.e., $|E_n| = M\ \forall n$; then the size of the set of observed curves is $|\mathbf{E}| = |E| = M^N$. (In reality, of course, $|E_{n_1}| \neq |E_{n_2}|$; however, this supposition is made merely to allow for easily intelligible complexity results.)

The key point is that at each of the sites, multiple edges may be detected; this is due to the fact that the object being tracked is not the only object present in the scene. The presence of this "clutter" is what makes the problem difficult. The notion of clutter is illustrated in Figs. 5, 6, and 7, which show images and their corresponding edge-maps; finding the object (head, ball, and finger) from the edge-map is not a straightforward task.

We may now turn to $\mathbf{C}$, which is referred to as the *shape space*. This set contains all possible curves which may

describe the object of interest; it encapsulates all of our knowledge of the object's geometry. $\mathbf{C}$ is generated before the algorithm is run (possibly by learning from training curves), and is a subset of curve space. The major assumptions on the structure of the shape space are the following:

- $\mathbf{C}$ is a finite dimensional compact C^1 manifold in curve space, with dimension K;
- $\mathbf{C}$ may be specified parametrically as

$$\mathbf{C} = \{\mathbf{c}(u) : u \in U\};$$

  where U is some known, K-dimensional, real, compact, convex set (for example, $U = [0,1]^K$);
- $\mathbf{c}(u)$ is a C^1 mapping from U to $\mathcal{C}$.

These are very reasonable and nonrestrictive assumptions, and allow for many classes of objects to be captured mathematically. The formalism also possesses a type of generality which conveniently allows camera-related transformations to be treated on an equal footing with noncamera deformations.

A simple example of a shape space is that of a rigid object which is allowed to translate or rotate by a certain amount each frame. Suppose that u_1 represents translation in the x-direction, and is constrained by $|u_1| \leq \Delta\mathbf{t}_x$; let u_2 be the analogous y variable with similar constraint. Finally, suppose u_3 is the rotation angle, and is subject to $|u_3| \leq \Delta\theta$. Then, we may write the shape space as

$$\mathbf{C} = \{\mathbf{c}(u) = R(u_3)\mathbf{c}_f + [u_1 \;\; u_2]^T : u \in U\},$$

where $\mathbf{c}_f$ is a template curve (most likely last frame's contour estimate), $R(u_3)$ is the rotation matrix corresponding to an angle of u_3, and $U = [-\Delta\mathbf{t}_x, \Delta\mathbf{t}_x] \times [-\Delta\mathbf{t}_y, \Delta\mathbf{t}_y] \times [-\Delta\theta, \Delta\theta]$.

A more complex example of a shape space is that of an object which is allowed to transform both rigidly and nonrigidly. In such a case, the degrees of freedom implicit in U will be divided between rigid and nonrigid deformations.

2.2 Other Considerations

Solving the optimization problem in (1) presents several difficulties. The first is that the problem is of a hybrid, or mixed continuous-discrete nature. While the set of observed curves $\mathbf{E}$ is discrete, the shape space $\mathbf{C}$ is a continuous, finite-dimensional manifold in curve space. There is no straightforward method for attacking such problems. Second, the observed set $\mathbf{E}$, while discrete, is extremely large. Recall, from Section 2.1, that if M edge-points are detected at each site, then $|\mathbf{E}| = M^N$. Typical values for M and N are 10 and 100, respectively; so $|\mathbf{E}| = 10^{100}$ is a fairly reasonable expectation. Clearly, exhaustive search over all of the elements of $\mathbf{E}$ is ruled out. The final aspect of the optimization problem which makes it difficult is that a *global* optimum is what is required. Of course, in any setting, a global optimum is preferable; however, the fact emerges in the experimental context that there are multiple local minima in typical cases, and that many of these are not very close to the global minimum. Thus, explicitly global methods ought to be developed. Each of these aspects of the problem would, on its own, present a reasonable challenge; together, they make the problem quite a difficult one. (Note, in this context, it is worth pointing out that the optimization problem is well-defined, i.e., that a minimum exists. This is due to the fact that U is a compact subset of $\mathbb{R}^K$, and $\|\mathbf{e} - \mathbf{c}(u)\|$ is continuous with respect to u; see, for example, [29].)

One complication which has not yet been addressed is the "correspondence problem." In forming a distance between curves such as the one used here, namely $\|\mathbf{e} - \mathbf{c}\|$, a natural issue arises: which point on the curve $\mathbf{e}$ corresponds with a given point on curve $\mathbf{c}$? Such a problem may be solved in a variety of ways; the choice here is to simply correspond points via their parameter values, where the parameter is equal to scaled arc-length (and so varies over the interval $[0,1]$).

This observation allows us to recast the problem slightly. Using the parametric form for $\mathbf{C}$, the problem may be rewritten

$$\min_{\mathbf{e}\in\mathbf{E}, u\in U} \|\mathbf{e} - \mathbf{c}(u)\|.$$

However, approximating the square of the L_2 norm by its Riemann sum gives

$$\begin{aligned}\|\mathbf{e} - \mathbf{c}(u)\|^2 &= \int_0^1 \|\mathbf{e}(s) - \mathbf{c}(s;u)\|^2 ds \\ &\approx \frac{1}{N}\sum_{n=1}^{N} \|e_n - c_n(u)\|^2,\end{aligned}$$

where $e_n = \mathbf{e}(s_n), c_n(u) = \mathbf{c}(s_n;u)$, and $s_n = \frac{(n-1)}{N-1}$. Note that $e_1, \ldots, e_N$ is simply the set of edge-points, culled from the sets $E_1, \ldots, E_N$, which were interpolated to give $\mathbf{e}$; sampling $\mathbf{e}$ gives back the original points. Denoting $e = (e_1, \ldots, e_N) \in \mathbb{R}^{2N}$ and similarly for $c(u)$, then the minimization problem may be approximated well by

$$\min_{e\in E, u\in U} \|e - c(u)\|, \tag{2}$$

if N is sufficiently large. Note that the norm in the above is the now the normal Euclidean norm in $\mathbb{R}^{2N}$, $E = E_1 \times \ldots \times E_N$ as before, and $c(\cdot) : U \to \mathbb{R}^{2N}$.

The recast problem, as expressed in (2), is the one which we shall solve. Like the original problem, this problem is not obviously amenable to solution, as E is still discrete and very large, while U is continuous. However, an approach for assault on the problem may now be outlined.

3 The Algorithm

The goal of the algorithm is to solve the problem

$$\min_{e\in E, u\in U} \|e - c(u)\|$$

accurately and efficiently. This may be broken down into two parts: minimization with respect to e and minimization with respect to u.

3.1 Minimization with Respect to e

If u is held fixed, the minimization with respect to e may proceed as follows: First, let us introduce the notation:

$$d(u) = \min_{e\in E} \|e - c(u)\|.$$

Finding $d(u)$ via exhaustive search through E (recall, E is discrete) requires $O(|E|) = O(M^N)$ operations; this is obviously infeasible. To reduce this complexity, the following observation will prove useful:

$$\begin{aligned} d^2(u) &= \min_{e \in E} \|e - c(u)\|^2 \\ &= \min_{e_1 \in E_1, \ldots, e_N \in E_N} \sum_{n=1}^{N} \|e_n - c_n(u)\|^2 \\ &= \sum_{n=1}^{N} \min_{e_n \in E_n} \|e_n - c_n(u)\|^2. \end{aligned}$$

Each minimization of the form $\min_{e_n \in E_n} \|e_n - c_n(u)\|^2$ may be solved independently; as a result, finding $d^2(u)$ and, hence, $d(u)$ may be achieved in $O(MN)$ steps (N minimizations, each over M values) rather than $O(M^N)$ steps. This represents a substantial savings. In fact, this result can be improved upon. By calculating nearest neighbors using Voronoi diagrams, it can be shown that each minimization of the form $\min_{e_n \in E_n} \|e_n - c_n(u)\|$ can be performed with $O(\log M)$ complexity, leading to an overall complexity of $O(N \log M)$. (Note, in order to gain this $\log$ factor, it is necessary to incur $O(M \log M)$ in overhead to calculate the relevant Voronoi diagram for each site; thus, the total overhead incurred is $O(NM \log M)$. This overhead, however, is small compared to the overall complexity of the algorithm; see Section 6 for further details.)

We have shown that if u is fixed, the minimization with respect to e can be performed efficiently, giving $d(u)$ for a particular u. Now, the tracking problem can be rewritten

$$\min_{u \in U} d(u).$$

The goal now is to show that this minimization with respect to U can also be performed accurately and efficiently. An obvious approach might be to set $\partial d / \partial u = 0$. Such a procedure generates a local, rather than a global minimum; experiments have shown that there are typically a very large number of such local minima. However, this consideration may be beside the point. Solving $\partial d / \partial u = 0$ is numerically infeasible because d is continuous, but not differentiable. This is due, essentially, to the fact two u's close to one another may have different minimizing e's. Thus, numerical methods such as gradient descent will not be applicable.

In order to present a more successful method of attacking the minimization with respect to u, it is necessary to first introduce some notation.

3.2 Notation

Definition. *V is said to be an ε-**cover** of the compact set U if $\forall u \in U$, $\exists v \in V$ such that $\|v - u\| \leq \varepsilon$, and ε is the smallest such value. Alternatively, $\varepsilon = \max_{u \in U}[\min_{v \in V} \|v - u\|]$. (Note that the maximum is well-defined since U is compact.)*

ε, in the above definition, is a measure of the sampling density. In particular, ε tells us about how well sampled the *poorest*-sampled region of U is. This is a useful way of proceeding, as it does not require that V sample U regularly or uniformly.

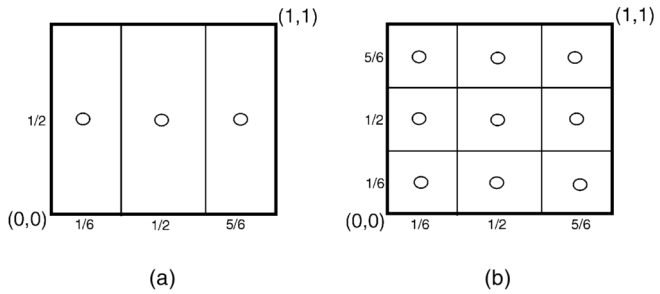

Fig. 3. An example of a TMS. $U = [0,1]^2$. The points in (a) belong to V_1; those in (b) belong to V_2. Note that $V_1 \subset V_2$. Other properties of this TMS, as well as an accompanying discussion, are given in the text.

Definition. *Given a compact set U, a set V satisfying $|V| < \infty$ and $V \subset U$, and a point $v \in V$, let*

$$S(v, V, U) = \{u \in U : \|v - u\| \leq \|v' - u\| \ \forall v' \in V\}.$$

Given sets V and U, the set of sets $\{S(v, V, U)\}_{v \in V}$ is almost a partition of U. The term "almost" is used because the collection $\{S(v, V, U)\}_{v \in V}$ does not quite satisfy all of the requirements which define a partition. These sets do exhaust U:

$$\bigcup_{v \in V} S(v, V, U) = U;$$

however, they are not quite pairwise disjoint:

$$S(v_1, V, U) \cap S(v_2, V, U) = \delta S(v_1, V, U) \cap \delta S(v_2, V, U),$$

where δS denotes the boundary of the set S. However, thinking of the collection as a partition helps in understanding what is to follow.

Definition. *An I-depth **tree minimization structure (TMS)** is the triple*

$$(U, \{V_i\}_{i=1}^{I}, \{\phi_i(\cdot)\}_{i=2}^{I}),$$

satisfying:

1. *U is a compact set*
2. *1) $|V_i| < \infty$, 2) $V_i \subset U$, and 3) $V_i \subset V_{i+1}$*
3. *For any $i \geq 2$, $\{\phi_i(v_{i-1})\}_{v_{i-1} \in V_{i-1}}$ is a partition of V_i such that 1) $v_{i-1} \in \phi_i(v_{i-1})$ and 2) $S(w, V_i, U) \subset S(v_{i-1}, V_{i-1}, U)$ $\forall w \in \phi_i(v_{i-1})$.*

The tree minimization structure is defined to capture the idea of coarse-to-fine, or treelike search through a dense set. To aid in understanding the definition and the following discussion, it may prove useful to refer to Fig. 3. In this figure are shown the rudiments of a small 2-depth TMS, which is given as follows:

- $U = [0,1]^2$,
- $V_1 = \{(\frac{1}{6},\frac{1}{2}),(\frac{1}{2},\frac{1}{2}),(\frac{5}{6},\frac{1}{2})\}$,
- $V_2 = \{(\frac{1}{6},\frac{1}{6}),(\frac{1}{2},\frac{1}{6}),(\frac{5}{6},\frac{1}{6}),(\frac{1}{6},\frac{1}{2}),(\frac{1}{2},\frac{1}{2}),(\frac{5}{6},\frac{1}{2}),(\frac{1}{6},\frac{5}{6}),(\frac{1}{2},\frac{5}{6}),(\frac{5}{6},\frac{5}{6})\}$,
- $\phi_2(\frac{1}{6},\frac{1}{2}) = \{(\frac{1}{6},\frac{1}{6}),(\frac{1}{6},\frac{1}{2}),(\frac{1}{6},\frac{5}{6})\}$
  $\phi_2(\frac{1}{2},\frac{1}{2}) = \{(\frac{1}{2},\frac{1}{6}),(\frac{1}{2},\frac{1}{2}),(\frac{1}{2},\frac{5}{6})\}$
  $\phi_2(\frac{5}{6},\frac{1}{2}) = \{(\frac{5}{6},\frac{1}{6}),(\frac{5}{6},\frac{1}{2}),(\frac{5}{6},\frac{5}{6})\}$.

Note that for any $v_1 \in V_1$, $S(v_1, V_1, U)$ is the rectangle which contains v_1 (see Fig. 3a); whereas for any $v_2 \in V_2$, $S(v_2, V_2, U)$ is the square which contains v_2 (see Fig. 3b).

Several aspects of the TMS definition bear commenting on. The set U through which search is to take place must be compact; the boundedness ensures that the search method proposed will indeed converge. The sets $\{V_i\}_{i=1}^{I}$ contain the tree structure in them: they are all subsets of U, and are packed like Babushka dolls one into the next. The idea then, is to be able to claim that after the ith stage in the algorithm that we have successfully searched through V_i; as $V_{i-1} \subset V_i$, each stage represents an improvement on the previous stage. The set of functions $\{\phi_i(\cdot)\}_{i=2}^{I}$ contains most of the action of the definition. The idea is that elements of V_i can be grouped into $|V_{i-1}|$ disjoint subsets of V_i; these subsets are denoted $\phi_i(v_{i-1})$ (one for each $v_{i-1} \in V_{i-1}$). The elements of $\phi_i(v_{i-1})$ may be thought of as the "children" of v_{i-1}. Now, the set V_{i-1} has the effect of (almost) partitioning U into the subsets $\{S(v_{i-1}, V_{i-1}, U)\}_{v_{i-1} \in V_{i-1}}$; similarly, the set V_i has the effect of (almost) partitioning U into the subsets $\{S(v_i, V_i, U)\}_{v_i \in V_i}$. The definition of the TMS requires that each $(i-1)$th stage subset of U, $S(v_{i-1}, V_{i-1}, U)$, must itself be such that it is (almost) partitioned by some collection of the ith stage subsets of U $\{S(v_i, V_i, U)\}_{v_i \in V_i}$; and this collection consists of precisely those subsets corresponding to the elements $v_i \in \phi_i(v_{i-1})$.

Definition. *A TMS $(U, \{V_i\}_{i=1}^{I}, \{\phi_i(\cdot)\}_{i=2}^{I})$ is said to be **convex** if $S(v, V_i, U)$ is convex for all $v \in V_i$; $i = 1, , I$.*

Convexity is useful in proving several of the lemmata, as they rely on application of the intermediate value theorem.

Definition. *Let $H(u) = \frac{\partial c}{\partial u}$, so that $H(u) \in \mathbb{R}^{2N \times K}$. Let $\lambda_1(u)$ be the largest eigenvalue of the $K \times K$ matrix $H^T(u)H(u)$. Then, for any $Y \subset U$, define $\xi(Y) = [\max_{u \in Y} \lambda_1(u)]^{1/2}$. In particular, let $\psi = \xi(U)$.*

The function $\xi(\cdot)$ gives some idea of the "wiggliness" of the manifold. Generally speaking, the value $\xi(Y)$ will be high if there is quite a bit of wiggliness in the manifold in the neighborhood of U given by Y; whereas if there is not much variation, $\xi(Y)$ will tend to be low. In the limit, where the manifold consists of a single curve, then $H(u) = 0 \; \forall u$, so that $\xi(Y) = 0 \; \forall Y \subset U$.

3.3 The Heart of the Algorithm: Tree Search through U

The algorithm, labeled MANIFOLD-TRACK, is now presented (see Fig. 4).

The arguments of the function are the observed set E and the TMS $(U, \{V_i\}_{i=1}^{I}, \{\phi_i(\cdot)\}_{i=2}^{I})$. The function returns an observed curve $e^\dagger$. Let us now pose the following questions. 1) What is the action of the algorithm? 2) Why does such an action allow accurate and efficient search through U?

The algorithm's main action is to "prune the tree" inherent in the TMS. Note that, if we changed the loop in line 6 to "for all $x_i \in V_i - V_{i-1}$," then we would be calculating $d(u)$ for all $u \in V_I$. (In the first stage, we calculate $d(u) \; \forall u \in V_1$; in the second stage, $d(u) \; \forall u \in V_2 - V_1$; and so on. After I stages, we have calculated $d(u) \; \forall u \in V_1 \cup (V_2 - V_1) \cup \ldots \cup (V_I - V_{I-1}) = V_I$.) However, in the actual algorithm, branches of the tree are pruned at each stage. Examine, for example, the first stage. $X_1 = V_1$, and $d(u)$ is calculated $\forall u \in X_1$. How many values of $d(u)$ are calculated in stage 2? The answer is given in the final line of the pseudocode: any $x_1 \in X_1$ such that $D(x_1; V_1) \geq d^\dagger$ will be pruned, and none of its descendants will ever be examined. As a result, the algorithm evaluates $d(u)$ at $|X_I|$ different values of u, rather than $|V_I|$ values; and $|X_I| \leq |V_I|$.

```
MANIFOLD-TRACK(E, (U, {V_i}_{i=1}^I, {φ_i(·)}_{i=2}^I))

X_0 = ∅
X_1 = V_1
d† = ∞
for i = 1 to I
        ε_i = max_{u∈U} min_{v∈V_i} ||v − u||
        for all x_i ∈ X_i − X_{i−1}
                d(x_i) = min_{c∈E} ||e − c(x_i)||
                c(x_i) = argmin_{c∈E} ||e − c(x_i)||
                if d(x_i) < d†
                        d† = d(x_i)
                        e† = e(x_i)
                γ = ξ(S(x_i, V_i, U))ε_i
                D(x_i; V_i) = (d²(x_i) − 2γd(x_i) − 3γ²)^{1/2}
        if i < I
                X_{i+1} = ∪_{x_i∈X_i : D(x_i;V_i)<d†} φ_{i+1}(x_i)
return e†
```

Fig. 4. Pseudocode for the manifold tracker.

We may now turn to the second question: why does the action of MANIFOLD-TRACK allow for accurate and efficient search through U? Two theorems will be presented in the next section to address these claims formally. Theorem 2 addresses the issue of efficiency: it shows that evaluating $d(u) \; \forall u \in X_I$ is equivalent to evaluating $d(u) \; \forall u \in V_I$. In other words,

$$\min_{u \in X_I} d(u) = \min_{u \in V_I} d(u).$$

The difference in size between X_I and V_I is discussed below. Theorem 1 addresses the issue of accuracy. In particular, it bounds the difference between the true optimum,

$$\min_{u \in U} d(u),$$

and the value generated by MANIFOLD-TRACK

$$\min_{u \in V_I} d(u) = \min_{u \in X_I} d(u).$$

Let us now turn to the issue of the increase in speed which arises from this algorithm. Brute force exhaustive search through V_I requires $|V_I|$ evaluation of $d(u)$; MANIFOLD-TRACK requires only $|X_I|$ such evaluations. Unfortunately, all that can be formally established (see Theorem 2) is that $|X_I| \leq |V_I|$. However, the experimental results show an excellent improvement in speed due to the relative size differences of X_I and V_I. In the case of the finger experiment, a typical frame yields $|V_I| \approx 10^7$ versus $|X_I| \approx 10^3$; and in the case of the ball experiment $|V_I| \approx 10^8$ versus $|X_I| \approx 10^2$. (The full descriptions of the experiments are contained in Section 7.) The speed increase represents one of the major advantages of the manifold tracker over condensation. When a manifold tracker performs $|X_I|$ operations, it attains the accuracy corresponding to having performed $|V_I|$ operations. In other words, for a

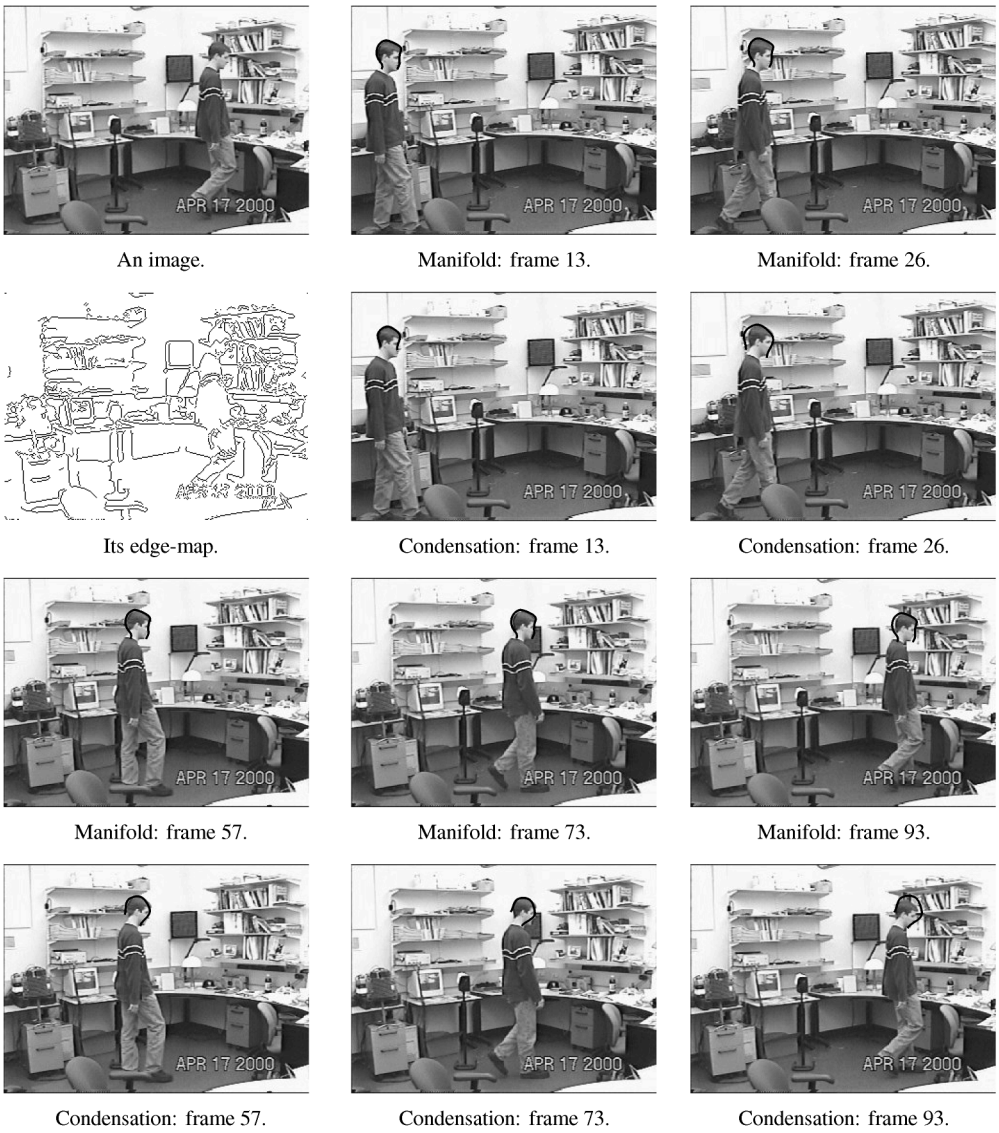

Fig. 5. The walking sequence.

small number of operations, it is effectively as though we have sampled U very densely. By contrast, when a condensation tracker performs $|X_I|$ operations, it attains the accuracy corresponding to having performed just exactly $|X_I|$ operations. Thus, it has sampled U much less densely than the manifold tracker. By sampling the shape space much more densely with the same number of operations, the manifold tracker yields a considerably more accurate match. In highly cluttered scenes, this factor can mean the difference between maintaining or losing lock. See Section 7 for more details.

4 Two Theorems

We are now ready to state the theorems which will allow us to attack the tracking problem, $\min_{e\in E, u\in U} \|e - c(u)\|$. The import of the theorems will be discussed after their formal statements.

Theorem 1. *Let V be any ε-cover of U. Further, let $d^* = \min_{e\in E,u\in U} \|e - c(u)\|$ and let $d^\dagger = \min_{u\in U} \|e^\dagger - c(u)\|$, where $e^\dagger = \operatorname{argmin}_{e\in E}(\min_{v\in V} \|e - c(v)\|)$. If $\Delta d = d^\dagger - d^*$, then*

$$0 \leq \Delta d \leq \frac{3\psi^2\varepsilon^2}{d^*} + 2\psi\varepsilon.$$

Theorem 1 is concerned with accuracy. It bounds the difference between the goodness of match using a finite sampling $V \subset U$, and the true optimum. There are two aspects to this upper bound on Δd which are worth commenting on. First, the ε dependence is as we would expect: as V samples U more and more finely, ε decreases, and the upper bound becomes increasingly small. In particular, as ε becomes very small, the quadratic term can be ignored, and the dependence of the upper bound on ε is linear. Second, the ψ dependence is also intuitive; as the manifold gets more wiggly, the upper bound grows. In order to mitigate the effect of a large ψ, U must be sampled finely enough to lead to a sufficiently small ε.

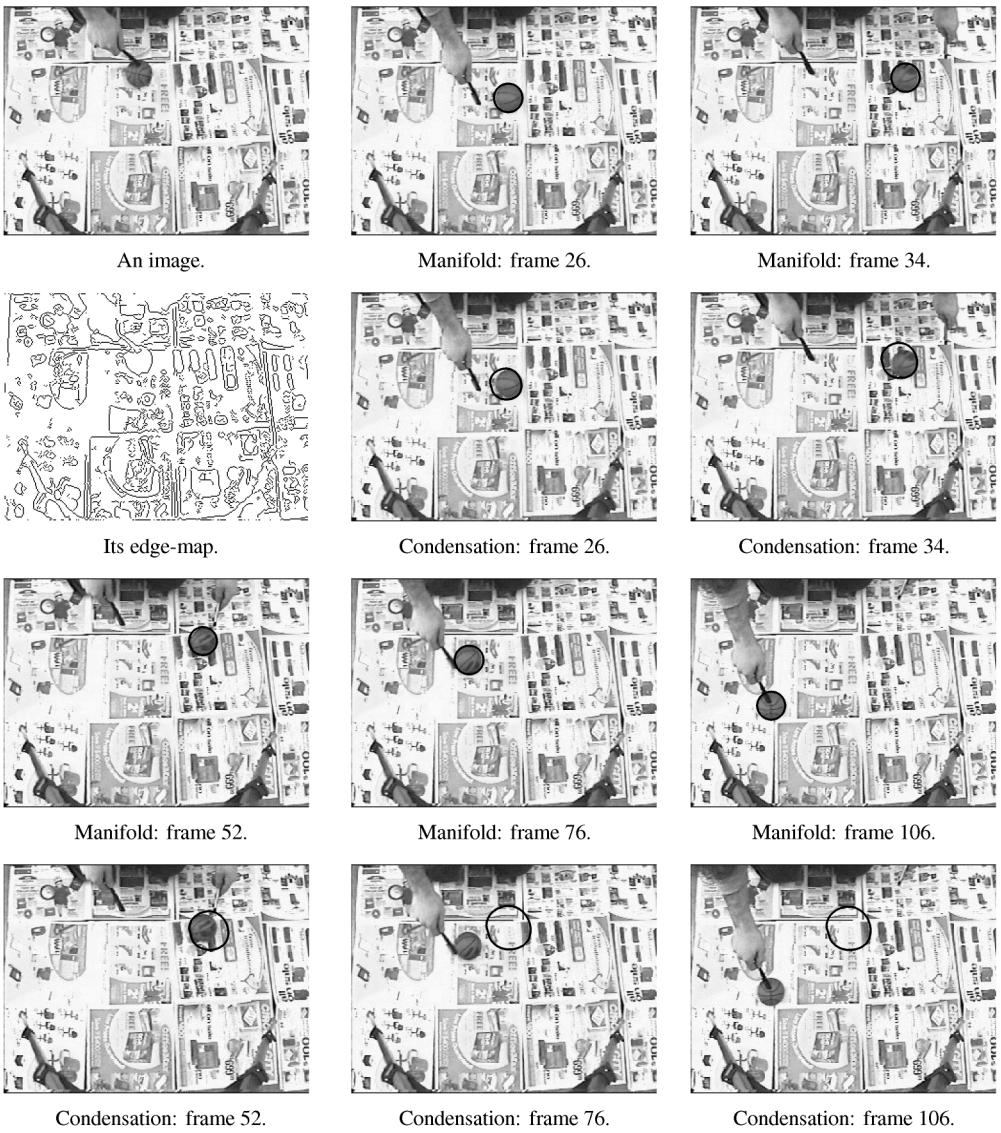

Fig. 6. The ball sequence.

Theorem 2. *Given a convex TMS* $(U, \{V_i\}_{i=1}^{I}, \{\phi_i(\cdot)\}_{i=2}^{I})$, *and an observed set* E, *let* X_I *be the set generated by* MANIFOLD-TRACK $(E, (U, \{V_i\}_{i=1}^{I}, \{\phi_i(\cdot)\}_{i=2}^{I}))$. *Then*

$$\min_{e\in E, x\in X_I} \|e - c(x)\| = \min_{e\in E, v\in V_I} \|e - c(v)\|,$$

and $|X_I| \leq |V_I|$.

Theorem 2 is concerned with efficiency. In particular, it presents a more efficient way of solving the problem $\min_{e\in E, v\in V} \|e - c(v)\|$, as long as V can be expressed as V_I for an I-depth TMS. To understand the intuition for this result, it is critical to understand the pruning mechanism by which branches of X_i are eliminated in generating X_{i+1}. It will prove useful to explain this idea with respect to the simple TMS illustrated in Fig. 3 and discussed in Section 3.2. At the first stage, $X_1 = V_1 = \{(\frac{1}{6}, \frac{1}{2}), (\frac{1}{2}, \frac{1}{2}), (\frac{5}{6}, \frac{1}{2})\}$. Suppose the values of d at these three u's are given by $d(\frac{1}{6}, \frac{1}{2}) = 10$, $d(\frac{1}{2}, \frac{1}{2}) = 15$, and $d(\frac{5}{6}, \frac{1}{2}) = 50$. The last value is considerably worse than the first two; on what grounds could we prune the last branch? We would need to be sure that for any u in the rectangle surrounding $(\frac{5}{6}, \frac{1}{2})$, $d(u) \geq 10$. If that were the case, then we would know that no u in this rectangle could ever be better than $u = (\frac{1}{6}, \frac{1}{2})$. In fact, this is precisely what is done. $D(x_i; V_i)$ is a lower bound on the minimal value of $d(u)$ for any u in $S(x_i, V_i, U)$ (where the latter is the aforementioned rectangle in Fig. 3). As a result, if $D(x_i; V_i) \geq d^\dagger$ (where $d^\dagger$ is the smallest d-value found thus far), then we can prune the branch emanating from x_i.

5 Proof of the Theorems

The following section formally establishes the validity of Theorems 1 and 2. In order to prove the two results, five lemmata will be stated and proven. In what follows, $d_i^\dagger$ denotes the value of $d^\dagger$ after i stages of MANIFOLD-TRACK have been completed.

Lemma 1. $X_i \subset V_i$.

Proof. Proceed by induction. Since $X_1 = V_1$, the lemma is satisfied trivially for $i = 1$. Suppose it is true for $i = k$: $X_k \subset V_k$. Then,

Fig. 7. The finger sequence.

$$
\begin{aligned}
X_{k+1} &= \bigcup_{x_k \in X_k : D(x_k;V_k) < d_k^\dagger} \phi_{k+1}(x_k) \\
&\subset \bigcup_{x_k \in X_k} \phi_{k+1}(x_k) \\
&\subset \bigcup_{x_k \in V_k} \phi_{k+1}(x_k) \\
&= V_{k+1},
\end{aligned}
$$

where the third line follows from the induction hypothesis, and the fourth line follows from the fact that $\{\phi_i(v_{i-1})\}_{v_{i-1} \in V_{i-1}}$ is a partition of V_i (Property 3 of the TMS definition). Note that this also establishes the second part of Theorem 2: $|X_I| \leq |V_I|$. □

Definition. *$v_{i_1} \in V_{i_1}$ is the i_1th **stage ancestor** of $v_{i_2} \in V_{i_2}$ if 1) $i_1 < i_2$ and 2) $\exists$ a sequence $\{v_i\}_{i=i_1+1}^{i_2-1}$ with $v_i \in V_i$, $i = i_1 + 1, \ldots, i_2 - 1$ and $v_{i+1} \in \phi_{i+1}(v_i)$, $i = i_1, \ldots, i_2 - 1$.*

The ancestor relationship, as defined above, is quite straightforward. Recall that the TMS embodies a coarse-to-fine, or treelike structure. The notion of ancestry here is exactly analogous to ancestry within a tree: we can trace a direct path downwards through the tree from an ancestor to its descendant. Here, the "path" is traced by the ϕ_i functions.

The following lemma establishes conditions that must hold if a point v^* is such that it belongs to V_I, but not to X_I.

Lemma 2. *$v^* \in V_I - X_I \Rightarrow \exists i < I$ and $v_i^* \in V_i$ such that 1) v_i^* is the i^{th} stage ancestor of v^* and 2) $D(v_i^*; V_i) \geq d_i^\dagger$.*

Proof. First, note that the condition $v^* \in V_I - X_I$ only makes sense because of Lemma 1, which establishes that $X_I \subset V_I$. Now, proceed by contradiction. Then if $v_1^*, \ldots, v_{I-1}^*$ are the first through $(I-1)$th stage ancestors of v^* (note the fact that there is only one ancestor at each stage due to the fact that the $\{\phi_i(\cdot)\}$ are bijections) $\Rightarrow D(v_i^*; V_i) < d_i^\dagger \ \forall i = 1, \ldots, I-1$. Now, proceed by induction. $v_1^* \in X_1$ since $X_1 = V_1$. Suppose $v_i^* \in X_i$ for $i > 1$. Then, $X_{i+1} = \bigcup_{x_i \in X_i : D(x_i;V_i) < d_i^\dagger} \phi_{i+1}(x_i)$, so that $\phi_{i+1}(v_i^*) \subset X_{i+1}$. But $v_{i+1}^* \in \phi_{i+1}(v_i^*)$ by the ancestry definition, so $v_{i+1}^* \in X_{i+1}$. Thus, by induction,

$v^* \in X_I$. This is a contradiction, since it was assumed that $v^* \in V_I - X_I$. □

Lemma 3 establishes a condition that is implied by ancestry.

Lemma 3. *If $v_i \in V_i$ is the ith stage ancestor of $v^* \in V_I$, then $v^* \in S(v_i, V_i, U)$.*

Proof. If $v_i \in V_i$ is the ith stage ancestor of $v^* \in V_I$, then $\exists$ a sequence $v_{i+1}, \ldots, v_{I-1}$ such that $v_{i+1} \in \phi_{i+1}(v_i)$. By Property (3b) of the TMS definition $v_{k+1} \in \phi_{k+1}(v_k) \Rightarrow S(v_{k+1}, V_{k+1}, U) \subset S(v_k, V_k, U)$; repeated application for $k = i, \ldots, I-1$ gives $S(v^*, V_I, U) \subset S(v_i, V_i, U)$. But, $v^* \in S(v^*, V_I, U)$ by definition, so $v^* \in S(v_i, V_i, U)$. □

Lemma 4 is critical both for establishing the upper bound in Theorem 1, as well as for establishing the lower bound D used in Theorem 2 in constructing the sets X_i. It is not easy to give an intuitive statement about the content of Lemma 4; instead, it is best thought of as a tool for proving the theorems. Its proof is also somewhat longer than those of the other lemmata.

Lemma 4. *Let $Q \subset U$ be a convex, compact set and let R be an ε-covering of Q. Let $e_1, e_2 \in E$, $d_i = \min_{u \in Q} \|e_i - c(u)\|$, and $\tilde{d}_i = \min_{v \in R} \|e_i - c(v)\|$. Then, $d_2^2 - d_1^2 \leq d_2^2 - \tilde{d}_1^2 + 3\xi^2(Q)\varepsilon^2 + 2d_2\xi(Q)\varepsilon$.*

Proof. Make the following definitions: for $i = 1, 2$, let

- $u_i = \operatorname{argmin}_{u \in Q} \|e_i - c(u)\|, \quad d_i = \|e_i - c(u_i)\|$
- $\breve{u}_i = \operatorname{argmin}_{v \in R} \|e_i - c(v)\|, \quad \tilde{d}_i = \|e_i - c(\breve{u}_i)\|$
- $\hat{u}_i = \operatorname{argmin}_{v \in R} \|u_i - v\|, \quad \hat{d}_i = \|e_i - c(\hat{u}_i)\|$

Then:

$$\begin{aligned} d_2^2 - d_1^2 &= \|e_2 - c(u_2)\|^2 - \|e_1 - c(u_1)\|^2 \\ &\leq \|e_2 - c(\hat{u}_2)\|^2 + \|c(\hat{u}_2) - c(u_2)\|^2 \\ &\quad - \|e_1 - c(\hat{u}_1)\|^2 + \|c(\hat{u}_1) - c(u_1)\|^2 \\ &= \hat{d}_2^2 - \hat{d}_1^2 + \|c(\hat{u}_1) - c(u_1)\|^2 + \|c(\hat{u}_2) - c(u_2)\|^2, \end{aligned}$$

where the inequality in the second line is a double application of the triangle inequality. Now,

1. $\hat{d}_1 \geq \tilde{d}_1$ by definition, so $-\hat{d}_1^2 \leq -\tilde{d}_1^2$
2. Expanding $\hat{d}_2^2 = \|e_2 - c(\hat{u}_2)\|^2$ gives

$$\hat{d}_2^2 = \|e_2 - [c(u_2) + H(\breve{u}_2)(\hat{u}_2 - u_2)]\|^2,$$

where $H(u) = \frac{\partial c}{\partial u}$ and $\breve{u}_2 \in Q$. This is the multivariable mean value theorem, which is valid due to the convexity of Q, see [8]. Thus,

$$\begin{aligned} \hat{d}_2^2 &= \|e_2 - c(u_2)\|^2 + \|H(\breve{u}_2)(\hat{u}_2 - u_2)\|^2 \\ &\quad + 2(e_2 - c(u_2))^T H(\breve{u}_2)(\hat{u}_2 - u_2). \end{aligned}$$

Now, break down the above expression term-by-term:

a. $\|e_2 - c(u_2)\|^2 = d_2^2$

b. $\|H(\breve{u}_2)(\hat{u}_2 - u_2)\|^2 = (\hat{u}_2 - u_2)^T H^T(\breve{u}_2) H(\breve{u}_2)(\hat{u}_2 - u_2)$. Since R is an ε-cover of Q, $\exists v$ such that $\|u_2 - v\| \leq \varepsilon$. But, by definition, $\hat{u}_2 = \operatorname{argmin}_{v \in R} \|u_2 - v\|$; thus, $\|\hat{u}_2 - u_2\| \leq \varepsilon$. But then,

$$\begin{aligned} &(\hat{u}_2 - u_2)^T H^T(\breve{u}_2) H(\breve{u}_2)(\hat{u}_2 - u_2) \\ &\leq \lambda_1(\breve{u}_2)\varepsilon^2 \leq \left(\max_{u \in Q} \lambda_1(u)\right)\varepsilon^2 \equiv \xi^2(Q)\varepsilon^2, \end{aligned}$$

where $\lambda_1(\cdot)$ and $\xi(\cdot)$ are defined as before.

c. Finally,

$$\begin{aligned} &(e_2 - c(u_2))^T H(\breve{u}_2)(\hat{u}_2 - u_2) \leq |(e_2 - c(u_2))^T \\ &H(\breve{u}_2)(\hat{u}_2 - u_2)| \leq \|e_2 - c(u_2)\| \|H(\breve{u}_2)(\hat{u}_2 - u_2)\|, \end{aligned}$$

where the latter inequality is due to the Cauchy-Schwartz inequality. But, $\|e_2 - c(u_2)\| = d_2$ and from the previous argument $\|H(\breve{u}_2)(\hat{u}_2 - u_2)\| \leq \xi(Q)\varepsilon$. Thus, $(e_2 - c(u_2))^T H(\breve{u}_2)(\hat{u}_2 - u_2) \leq d_2\xi(Q)\varepsilon$.

3. Using the mean value theorem once again

$$\begin{aligned} \|c(\hat{u}_i) - c(u_i)\|^2 &= \|c(u_i) + H(\breve{u}_i)(\hat{u}_i - u_i) - c(u_i)\|^2 \\ &= \|H(\breve{u}_i)(\hat{u}_i - u_i)\|^2 \\ &\leq \xi^2(Q)\varepsilon^2. \end{aligned}$$

Thus,

$$d_2^2 - d_1^2 \leq d_2^2 - \tilde{d}_1^2 + 3\xi^2(Q)\varepsilon^2 + 2d_2\xi(Q)\varepsilon.$$

□

The final lemma establishes the fact that D really is a lower bound on the minimum in a region of U: if the minimal value at stage i, $d_i^\dagger$, is less than D for a particular region, then that region can be eliminated.

Lemma 5. $D(v_i; V_i) \geq d_i^\dagger \Rightarrow \min_{e \in E} \|e - c(u)\| \geq d_i^\dagger \quad \forall u \in S(v_i, V_i, U)$.

Proof. For any $e_1, e_2 \in E$, Lemma 4 states that $d_2^2 - d_1^2 \leq d_2^2 - \tilde{d}_1^2 + 3\xi^2(Q)\varepsilon^2 + 2d_2\xi(Q)\varepsilon \Rightarrow -d_1^2 \leq -\tilde{d}_1^2 + 3\xi^2(Q)\varepsilon^2 + 2d_2\xi(Q)\varepsilon$. But, $\tilde{d}_2 \geq d_2 \Rightarrow d_1^2 \geq \tilde{d}_1^2 - 3\xi^2(Q)\varepsilon^2 - 2\tilde{d}_2\xi(Q)\varepsilon$. Now, choose

$$\begin{aligned} e_1 &= \operatorname*{argmin}_{e \in E}\left(\min_{u \in Q} \|e - c(u)\|\right) \\ e_2 &= \operatorname*{argmin}_{e \in E}\left(\min_{v \in R} \|e - c(v)\|\right). \end{aligned}$$

Then, $\tilde{d}_2 \leq \tilde{d}_1$ (see the definitions of $\tilde{d}_i$). Thus,

$$d_1 \geq (\tilde{d}_2^2 - 3\xi^2(Q)\varepsilon^2 - 2\tilde{d}_2\xi(Q)\varepsilon)^{1/2}.$$

Let $Q = S(v_i, V_i, U)$ and $R = \{v_i\}$ (i.e., a single element set). Then, $\tilde{d}_2 = d(v_i)$ and $d_1 = \min_{e \in E, u \in S(v_i, V_i, U)} \|e - c(u)\|$, so that the above inequality becomes

$$\min_{e \in E, u \in S(v_i, V_i, U)} \|e - c(u)\| \geq D(v_i; V_i).$$

Finally,

$$\begin{aligned} D(v_i; V_i) \geq d_i^\dagger &\Rightarrow \min_{e \in E, u \in S(v_i, V_i, U)} \|e - c(u)\| \geq d_i^\dagger \\ &\Rightarrow \min_{e \in E} \|e - c(u)\| \geq d_i^\dagger \ \forall u \in S(v_i, V_i, U). \end{aligned}$$

□

We are finally in a position to prove the two theorems.

TABLE 1
Summary of Experimental Results

Experiment	Length (frames)	Length Before Tracker Lost Lock (frames)	Processing Speed / Frame (seconds)	Samples Examined	Samples Effectively Seen
Walker: Manifold	101	101	$\approx$ 30-60	$\lvert X_I\rvert \approx 1000$	$\lvert V_I\rvert \approx 10^7$
Walker: Condensation	101	101	$\approx$ 20-30	$\nu = 2000$	$\nu = 2000$
Ball: Manifold	183	183	$\approx$ 30-60	$\lvert X_I\rvert \approx 100$	$\lvert V_I\rvert \approx 10^8$
Ball: Condensation	183	69	$\approx$ 20-30	$\nu = 2000$	$\nu = 2000$
Finger: Manifold	202	202	$\approx$ 30-60	$\lvert X_I\rvert \approx 1000$	$\lvert V_I\rvert \approx 10^7$
Finger: Condensation	202	47	$\approx$ 20-30	$\nu = 2000$	$\nu = 2000$

Proof of Theorem 1. For any $e_1, e_2 \in E$, Lemma 4 states that $d_2^2 - d_1^2 \leq d_2^2 - \tilde{d}_1^2 + 3\xi^2(Q)\varepsilon^2 + 2d_2\xi(Q)\varepsilon$. However, $d_2 \leq \tilde{d}_2$ so that $d_2^2 - d_1^2 \leq \tilde{d}_2^2 - \tilde{d}_1^2 + 3\xi^2(Q)\,\varepsilon^2 + 2d_2\xi(Q)\varepsilon$. Now, let $Q = U$ and $R = V$; further, let $e_1 = e^* = \text{argmin}_{e\in E}\,[\min_{u\in U}\|e - c(u)\|]$ and $e_2 = e^\dagger = \text{argmin}_{e\in E}\,[\min_{v\in V}\|e - c(v)\|]$. Then, by definition, $\tilde{d}_2 \leq \tilde{d}_1$, so that $\tilde{d}_2^2 - \tilde{d}_1^2 \leq 0$ and, thus,

$$d^{\dagger 2} - d^{*2} \leq 3\xi^2(U)\varepsilon^2 + 2d^\dagger\xi(U)\varepsilon$$

$$d^\dagger - d^* \leq \frac{3\xi^2(U)\varepsilon^2 + 2d^\dagger\xi(U)\varepsilon}{d^* + d^\dagger} = \frac{\frac{3\xi^2(U)\varepsilon^2}{d^\dagger} + 2\xi(U)\varepsilon}{1 + \frac{d^*}{d^\dagger}}.$$

However, $1/(1 + \frac{d^*}{d^\dagger}) \leq 1$ since $d^* \geq 0$; also, substituting $\psi = \xi(U)$ gives

$$\Delta d \leq \frac{3\psi^2\varepsilon^2}{d^\dagger} + 2\psi\varepsilon \leq \frac{3\psi(U)\varepsilon^2}{d^*} + 2\psi\varepsilon.$$

□

Proof of Theorem 2. Proceed by contradiction. Suppose $\min_{e\in E, x\in X_I}\ \|e - c(x)\| \neq \min_{e\in E, v\in V_I}\|e - c(v)\|$; in particular, since by Lemma 1 $X_I \subset V_I$, suppose that $d_I^\dagger \equiv \min_{e\in E, x\in X_I}\ \|e - c(x)\| > \min_{e\in E, v\in V_I}\|e - c(v)\|$. Let $v^* = \text{argmin}_{v\in V_I}\ (\min_{e\in E}\|e - c(v)\|)$; then, the previous supposition implies that $v^* \in V_I - X_I$. By Lemma 2, there exists an ith stage ancestor $v_i^* \in V_i$ of v^*, for some $i < I$, such that $D(v_i^*; V_i) \geq d_i^\dagger$. But, by Lemma 5, it follows that $\min_{e\in E}\|e - c(u)\| \geq d_i^\dagger \quad \forall u \in S(v_i^*, V_i, U)$. Lemma 3 asserts that $v^* \in S(v_i^*, V_i, U)$; thus, in particular $\min_{e\in E}\|e - c(v^*)\| \geq d_i^\dagger$. Finally, note that $d_i^\dagger \geq d_I^\dagger$; thus, $\min_{e\in E}\|e - c(v^*)\| \geq d_I^\dagger$. This is a contradiction. □

6 Complexity

Let us consider the complexity of MANIFOLD-TRACK. As has been noted in Section 3.1, calculating $d(u)$ for a single u is an $O(N\log M)$ procedure, for which an overhead of $O(NM\log M)$ is incurred. $d(u)$ is evaluated for all $u \in X_I$; thus, the total complexity is

$$O(N|X_I|\log M + NM\log M) = O(N|X_I|\log M),$$

as typically, $M \ll |X_I|$.

It is more informative to express the complexity in terms of Δd, the proximity of the actual solution to the true optimum, rather than $|X_I|$. To do so, use a dimensional argument. Let V be an ε-covering of U; then, using something akin to sphere-packing, it is clear that $vol(U) \approx |V|\varepsilon^K$, where $K = dim(U) = dim(C)$. That is, $|V| \propto \varepsilon^{-K}$. Now, assuming that Δd is fairly small, it can be shown that the upper bound on Δd from Theorem 1 is proportional to ε. (Proof: From Theorem 1, the upper bound is given by $\Delta d = a\varepsilon^2 + b\varepsilon$, where the constants a and b are given in Theorem 1. In this case, we can solve to show that $\varepsilon = \frac{b}{2a}(\sqrt{1 + \frac{4a\Delta d}{b^2}} - 1)$. If Δd is small, then Taylor Expansion gives that $\varepsilon \approx \Delta d/b$, so that $\varepsilon = O(\Delta d)$.) Finally, note that $|X_I| \leq |V_I| \propto \varepsilon^{-K} \propto \Delta d^{-K}$. Thus, the algorithm has complexity

$$O(N\Delta d^{-K}\log M).$$

Based on experiments, it is assumed that in many cases, $|X_I|$ will be *much* smaller than $|V_I|$, although as yet no formal result to illustrate this has been achieved; all that is known is that $|X_I| \leq |V_I|$. (For positive experimental results concerning the relative sizes of $|X_I|$ and $|V_I|$, see Section 7.) In terms of the more relevant parameter Δd, it is hoped that a result may be proven to show that the complexity using the algorithm described in Theorem 2 is of the form $O(N\Delta d^{-\beta}\log M)$, where $\beta < K$; the difference between β and K will depend heavily on the behavior of the manifold $\mathbf{C}$.

7 Results and Conclusions

Results from tracking three sequences are shown in Figs. 5, 6, and 7. In each case, the manifold tracker was tested against a condensation tracker. Statistics about the performance of the two trackers is summarized in Table 1. In order to make the comparison between the two trackers fair, the number of samples of U examined by the manifold tracker ($|X_I|$) is not allowed to exceed the number of samples examined by the condensation tracker (ν). Also, note that both of the algorithms were implemented on a 300 MHz Pentium II machine in uncompiled MATLAB.

In the first sequence, an individual's head is tracked as he walks through mildly cluttered scene. The shape space $\mathbf{C}$ is taken to be a small set of translations, rotations, and scalings of a fixed head template (the template is culled from a training image); this set is centered about the previous frame's estimate. The dynamical model for the condensation tracker is simply a uniform distribution over this set. Both trackers successfully follow the head for the entire length of the sequence, although the condensation tracker is generally further from the truth; see, for example, frames 26 and 93 in Fig. 5. The reason for this is the fact that condensation samples the shape space considerably less densely than does the manifold tracker; whereas $\nu = 2,000$ samples are used in condensation, $|V_I| \approx 10^7$ samples are effectively examined by the manifold tracker, despite the fact that, in reality, $|X_I| \approx 1,000$ samples are actually looked at. It is therefore natural that the condensation estimate would be less refined.

In the second sequence, a ball is tracked through a scene with a considerable amount of clutter (generated by the newspaper lying underneath the ball); the degree of clutter is illustrated by the edge-map in Fig. 6. The shape space $\mathbf{C}$ is similar to that described for the head sequence (with the exceptions that the template is known a priori to be a circle, and as such no rotations are necessary), as is the dynamical model. The manifold tracker successfully tracks all 183 frames, while the condensation tracker loses lock permanently after frame 69. Here, the denser sampling provided by the MANIFOLD-TRACK algorithm is absolutely critical: it is the difference between maintaining and losing lock. The more cluttered a scene is, the more important dense sampling is. In this experiment, there is sufficient clutter that several spurious, ball-like arrangements of edge-points exist; while these arrangements are not quite as ball-like as the true ball, they are close enough to fool a tracker which does not sample $\mathbf{C}$ sufficiently densely. In this case, the efficacy of tree-search is even more striking: while the manifold tracker actually looks at about 10 times *fewer* samples than condensation ($\nu = 2,000$ while $|X_I| \approx 100$), it effectively looks at 100,000 times more samples ($|V_I| \approx 10^8$).

Finally, the last sequence shows a finger being tracked as it flexes and translates. In this case, the shape space $\mathbf{C}$ is learned from a training sequence (using PCA-type methods). Furthermore, a second-order linear dynamical model which lives in $\mathbf{C}$ is trained on the sequence, using the techniques described in [4]. In this case, condensation yields odd results, as can be seen in frames 46 and 49 of Fig. 7. During the short sequence of frames 44-50, the finger is stationary. However, the condensation tracker chooses a flexed finger for its estimate, as this is what its dynamical model tells it is probable. Thus, this tracker has the bizarre property that, in this sequence, it is unable to track a stationary object! The manifold tracker has no such problem, as it searches over a larger space of finger configurations (given that it has no dynamical model). Once again, it is able to search through this larger set due to the efficient tree search justified in Theorem 2.

On the negative side of the ledger, we should note that condensation outperforms the manifold tracker by a factor of about 2 in terms of actual speed; this is primarily due to some of the more complex operations (such as finding of eigenvalues) need for manifold tracking. Whether or not the manifold tracker can make up this difference through more careful implementation is an open question. Another advantage of condensation over manifold tracking is the ability to handle missing edges. An underlying assumption of the manifold tracker assumes that the set of edges at each site is nonempty: $E_n \neq \emptyset$. One way of dealing with empty sites is to simply ignore them in the calculation; in other words, whereas previously $d(u)$ was calculated as

$$d(u) = \sum_{n=1}^{N} \min_{e_n \in E_n} \|e_n - c_n(u)\|^2,$$

it is now emended to be

$$d(u) = \sum_{j=1}^{J} \min_{e_{n_j} \in E_{n_j}} \|e_{n_j} - c_{n_j}(u)\|^2,$$

where $\{n_j\}_{j=1}^{J}$ are the nonempty sites. Such a method will work well when there are only a small number of empty sites. However, in the case of major occlusions, it is not at all clear that it will succeed. In particular, due to its reliance on dynamics, condensation is able to (in some cases) deal with almost *complete* occlusions; it is unlikely that the manifold tracker, in its present form, would be able to do so.

There are several directions for future research. First, an attempt will be made to theoretically characterize the gains in efficiency due to tree-search. Second, the algorithm will be modified to more properly deal with occlusions. Third, the algorithm will be extended to take account of intensity information (in addition to edge information). Fourth, the problem will be adapted to the tracking of surfaces in range images. Finally, more attention will be paid to the implementation of the algorithm, in order to test its true speed.

REFERENCES

[1] Y. Akgul and C. Kambhamettu, "A Scale-Space Based Approach for Deformable Contour Optimization," *Proc. Second Int'l Conf. Scale-Space Theories in Computer Vision,* pp. 410-421, 1999.

[2] A. Arsenio and J. Santos-Victor, "Robust Visual Tracking by an Active Observer," *Proc. 1997 IEEE/RSJ Int'l Conf. Intelligent Robot and Systems: Innovative Robotics for Real-World Applications,* pp. 1342-1347, 1997.

[3] N. Ayache, I. Cohen, and I. Herlin, "Medical Image Tracking," *Active Vision,* Mass.: MIT Press, A. Blake and A. Yuille, eds., pp. 285-302, 1992.

[4] A. Blake, *Active Contours.* Springer Verlag, 1998.

[5] A. Blake, R. Curwen, and A. Zisserman, "A Framework for Spatio-Temporal Control in the Tracking of Visual Contours," *Int'l J. Computer Vision,* vol. 11, no. 2, pp. 127-145, 1993.

[6] A. Blake and M. Isard, "Condensation—Conditional Density Propagation for Visual Tracking," *Int'l J. Computer Vision,* vol. 29, no. 1, pp. 5-28, 1998.

[7] A. Blake, M. Isard, and D. Reynard, "Learning to Track the Visual Motion of Contours," *Artificial Intelligence,* vol. 78, pp. 101-134, 1995.

[8] W. Boothby, *An Introduction to Differentiable Manifolds and Riemannian Geometry.* Second ed., Academic Press, 1986.

[9] C. Bregler and Y. Konig, "Eigenlips for Robust Speech Recognition," *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing,* vol. 2, pp. 669-672, 1994.

[10] R. Brockett and A. Blake, "Estimating the Shape of a Moving Contour," *Proc. 33rd IEEE Conf. Decision and Control,* pp. 3247-3252, 1994.

[11] V. Caselles, F. Catte, T. Coll, and F. Dibos, "A Geometric Model for Active Contours in Image Processing," *Numerical Math.,* vol. 66, pp. 1-31, 1993.

[12] V. Caselles, R. Kimmel, and G. Sapiro, "On Geodesic Active Contours," *Int'l J. Computer Vision,* vol. 22, no. 1, pp. 61-79, 1997.

[13] M. Chan, Y. Zhang, and T. Huang, "Real-Time Lip Tracking and Bimodal Continuous Speech Recognition," *Proc. 1998 IEEE Second Workshop Multimedia Signal Processing,* pp. 65-70, 1998.

[14] B. Dalton, R. Kaucic, and A. Blake, "Automatic Speechreading Using Dynamic Contours," *Proc. NATO ASI Conf. Speechreading by Man and Machine: Models, Systems, and Applications,* NATO Scientific Affairs Division, Sept. 1995.

[15] J. Dong, "Stable Snake Algorithm for Convex Tracking of MRI Sequences," *Electronics Letters,* vol. 135, no. 13, pp. 1070-1071, 1999.

[16] L. Girin, E. Foucher, and G. Feng, "An Audio-Visual Distance for Audio-Visual Speech Vector Quantization," *Proc. 1998 IEEE Second Workshop Multimedia Signal Processing,* pp. 523-528, 1998.

[17] L. Girin, G. Feng, and J. Schwartz, "Noisy Speech Enhancement by Fusion of Auditory and Visual Information: A Study of Vowel Transitions," *Proc. Fifth European Conf. Speech Comm. and Technology,* pp. 2555-2558, 1997.

[18] A. Hill, A. Thornham, and C. Taylor, "Model Based Interpretation of 3D Medical Images," *Proc. Fourth British Machine Vision Conf.,* pp. 339-348, 1993.

[19] D. Hogg, "Model-Based Vision: A Program to See a Walking Person," *Image and Vision Computing,* vol. 1, no. 1, pp. 5-20, 1983.

[20] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active Contour Models," *Proc. First IEEE Int'l Conf. Computer Vision,* June 1987.

[21] R. Kaucic and A. Blake, "Accurate, Real-Time, Unadorned Lip Tracking," *Proc. Sixth IEEE Int'l Conf. Computer Vision,* pp. 370-375, 1998.

[22] R. Kaucic, B. Dalton, and A. Blake, "Real-Time Lip Tracking for Audio-Visual Speech Recognition Application," *Proc. Fourth European Conf. Computer Vision,* pp. 376-387, 1996.

[23] D. Koller, J. Weber, and J. Malik, "Towards Realtime Visual Based Tracking in Cluttered Traffic Scenes," *Proc. Intelligent Vehicles '94 Symp.,* pp. 201-206, 1994.

[24] M. Lievin, P. Delmas, P. Coulon, F. Luthon, and V. Fristol, "Automatic Lip Tracking: Bayesian Segmentation and Active Contours in a Cooperative Scheme," *Proc. IEEE Int'l Conf. Multimedia Computing and Systems,* pp. 691-696, 1999.

[25] P. Lipson, A. Yuille, D. O'Keefe, J. Cavanaugh, J. Taafe, and D. Rosenthal, "Deformable Templates for Feature Extraction from Medical Images," *Proc. First European Conf. Computer Vision,* 1990.

[26] J. Luettin, N. Thacker, and S. Beet, "Visual Speech Recognition Using Active Shape Models and Hidden Markov Models," *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing,* pp. 817-820, 1996.

[27] S. McKenna and S. Gong, "Tracking Faces," *Proc. Second Int'l Conf. Automatic Face and Gesture Recognition,* pp. 271-276, 1996.

[28] M. Mignotte and J. Meunier, "Deformable Template and Distribution Mixture-Based Data Modeling for the Endocardial Contour Tracking in an Echographic Sequence," *Proc. 1999 IEEE Computer Soc. Conf. Computer Vision and Pattern Recognition,* pp. 225-230, 1999.

[29] J. Munkres, *Topology: A First Course.* Prentice-Hall, 1975.

[30] B. Rao, H. Durrant-Whyte, and J. Sheen, "A Fully Decentralized Multi-Sensor System for Tracking and Surveillance," *Int'l J. Robotics Research,* vol. 12, no. 1, pp. 20-44, 1993.

[31] J. Sanchiz, F. Pla, and J. Marchant, "Vision-Based Approach to Automate Spraying in Crop Fields," *Proc. SPIE—The Int'l Soc. for Optical Eng.,* vol. 3364, pp. 287-297, 1998.

[32] I. Schwartz, "Primus. Realization Aspects of an Autonomous Unmanned Robot," *Proc. SPIE—The Int'l Soc. for Optical Eng.,* vol. 3364, pp. 328-334, 1998.

[33] G. Sullivan, "Visual Interpretation of Known Objects in Constrained Scenes," *Philosophical Trans. Royal Soc. of London B,* vol. 337, pp. 109-118, 1992.

[34] F. Thomanek, E. Dickmanns, and D. Dickmanns, "Multiple Object Recognition and Scene Interpretation for Autonomous Road Vehicle Guidance," *Proc. Intelligent Vehicles '94 Symp.,* pp. 231-236, 1994.

[35] G. Xu, E. Segawa, and S. Tsuji, "Robust Active Contours with Insensitive Parameters," *Proc. Fourth IEEE Int'l Conf. Computer Vision,* May 1993.

[36] A. Yuille, P. Hallinan, and D. Cohen, "Feature Extraction from Faces Using Deformable Templates," *Int'l J. Computer Vision,* vol. 8, no. 2, pp. 99-112, 1992.

Daniel Freedman (M '00) received the AB degree in physics from Princeton University in 1993 and the PhD degree in engineering sciences from Harvard University in 2000. He has been at Rensselaer Polytechnic Institute in Troy, New York since 2000, where he is currently an assistant professor in the Department of Computer Science. He is a member of Sigma Xi, the IEEE, and the IEEE Computer Society. His research interests include computational geometry, computer vision, and image processing.