MIT LABORATORY FOR INFORMATION & DECISION SYSTEMS TECHNICAL REPORT P-2603, MAY 2004
PRESENTED AT THE 2004 IEEE CVPR WORKSHOP ON GENERATIVE MODEL BASED VISION

Visual Hand Tracking Using Nonparametric Belief Propagation

Erik B. Sudderth, Michael I. Mandel, William T. Freeman, and Alan S. Willsky
Department of Electrical Engineering and Computer Science
Massachusetts Institute of Technology
esuddert@mit.edu, mim@mit.edu, billf@ai.mit.edu, willsky@mit.edu

***Abstract*— This paper develops probabilistic methods for visual tracking of a three-dimensional geometric hand model from monocular image sequences. We consider a redundant representation in which each model component is described by its position and orientation in the world coordinate frame. A prior model is then defined which enforces the kinematic constraints implied by the model's joints. We show that this prior has a local structure, and is in fact a pairwise Markov random field. Furthermore, our redundant representation allows color and edge-based likelihood measures, such as the Chamfer distance, to be similarly decomposed in cases where there is no self–occlusion.**

Given this graphical model of hand kinematics, we may track the hand's motion using the recently proposed nonparametric belief propagation (NBP) algorithm. Like particle filters, NBP approximates the posterior distribution over hand configurations as a collection of samples. However, NBP uses the graphical structure to greatly reduce the dimensionality of these distributions, providing improved robustness. Several methods are used to improve NBP's computational efficiency, including a novel KD-tree based method for fast Chamfer distance evaluation. We provide simulations showing that NBP may be used to refine inaccurate model initializations, as well as track hand motion through extended image sequences.

I. INTRODUCTION

Accurate visual detection and tracking of three–dimensional articulated objects is a challenging problem with applications in human–computer interfaces, motion capture, and scene understanding [25]. In this paper, we develop a probabilistic method for tracking a geometric hand model from monocular image sequences. Because articulated hand models have many (roughly 26) degrees of freedom, which are only indirectly related to the observed images, exact representation of the posterior distribution over model configurations is intractable. Extended and unscented Kalman filters [12, 14, 19] approximate the posterior by a single Gaussian, and update these approximations via a linearization of the measurement process. However, because many different hand configurations may approximately match a given image, the true posterior is often multimodal, making linear approximations ineffective.

Given the ambiguities inherent in visual tracking problems, many authors have considered nonparametric density representations. For example, particle filters [6] approximate the posterior distribution by a set of representative elements, and use Monte Carlo importance sampling rules to update these particles. However, due to the large number of degrees of freedom in hand tracking problems, particle filters cannot hope to accurately represent the true posterior. Instead, particles tend to concentrate in only a few of the most significant modes, and the tracker can suffer catastrophic failures. This problem has motivated previous authors to consider simplified models which only allow a limited range of object motions [11], as well as sophisticated prior models which better predict the object's dynamics [16, 26].

Deterministic nonparametric approximations are also possible, as demonstrated by a recently proposed tree–based estimator [20] which defines a multiscale discretization of the state space. This approach achieves computational savings by approximating image likelihoods as piecewise constant at coarse scales of the discretization, and only recursively evaluating those cells who's probability is above a predefined threshold. However, this approach is only effective when the tree structure is constructed using prior information which strongly constrains the hand's configuration. If the prior is uninformative, such pruning rules are very likely to miss important hand configurations.

Given the difficulties in approximating high–dimensional distributions, some authors have proposed replacing tracking by classification [1, 15], where classes correspond to some discretization of allowable hand configurations. These methods are most appropriate in applications such as sign language recognition, where only a small set of poses is of primary interest. Also, as these methods are based on precomputed images of the hand from all possible configurations, they require large amounts of storage space. A recently proposed method for interpolating between classes [22] makes no use of the image data during the interpolation, and thus assumes that the transition between any pair of hand pose classes is highly predictable.

An alternative way to address the high dimensionality of articulated tracking problems is to identify statistical structure within the posterior distribution. This structure can be described using a *graphical model*, or Bayesian network. Graphical models have been used to track view–based human body representations [13], contour models of restricted hand configurations [4], view–based 2.5D "cardboard" models of hands and people [24], and a full 3D kinematic human body model [17]. Because the variables in these graphical models are continuous, and discretization is intractable for three–dimensional models, most traditional graphical inference algorithms are inapplicable. Instead, these trackers are based on recently proposed extensions of particle filters to general graphs: mean field Monte Carlo in [24], and *nonparametric belief propagation* (NBP) [8, 21] in [17].

In this paper, we show that NBP may be used to track a three–dimensional geometric model of the hand. To derive

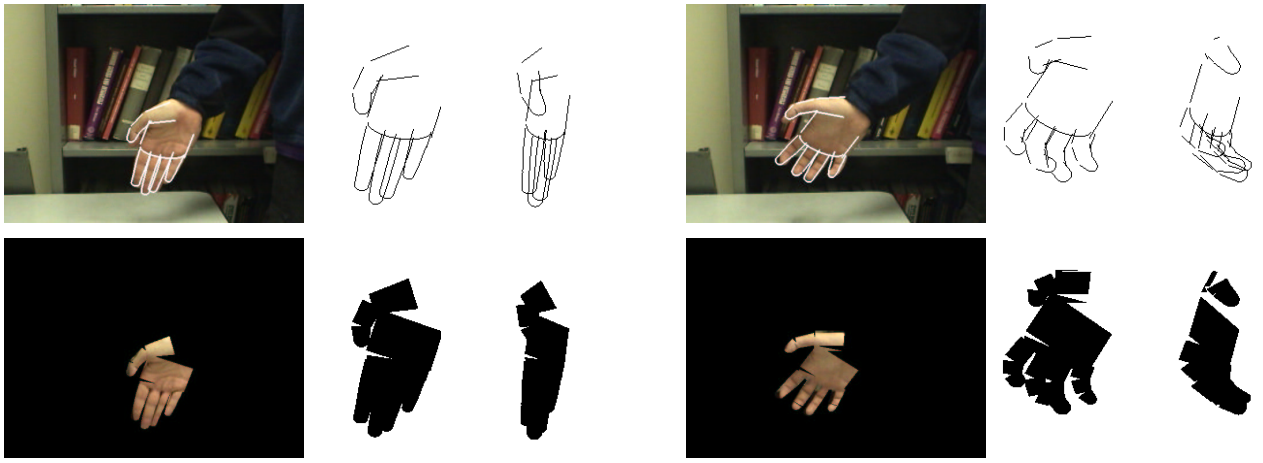


Fig. 1. Projected edges (top row) and silhouettes (bottom row) for two configurations (left and right blocks) of the 3D structural hand model. To aid visualization, the model joint angles are set to match the images (left), and then also projected following rotations by 35° (center) and 70° (right) about the vertical axis.

a graphical model for the tracking problem, we consider a redundant *local* representation in which each hand component is described by its own three–dimensional position and orientation. We show that the model's kinematic constraints, including self–intersection constraints not captured by joint angle representations, take a simple form in this local representation. Furthermore, in cases where model components do not significantly occlude each other, standard edge and color based likelihood measures may be similarly decomposed. We describe the implementation of NBP on this model, as well as several methods for improving computational efficiency. These include a novel method for fast orientation–based Chamfer distance evaluation using KD–trees [2]. We conclude with simulations demonstrating that NBP can refine noisy initializations in single frames, as well as track hand motion over two extended sequences.

II. Geometric Hand Modeling

A. Structural Model

Structurally, the hand is composed of sixteen approximately rigid components: three phalanges or links for each finger and thumb, as well as the palm [25]. As proposed by [14, 19], we model each rigid body by one or more truncated quadrics (ellipsoids, cones, and cylinders). These geometric primitives are well matched to the true geometry of the hand, and in contrast to 2.5–dimensional "cardboard" models [24, 26], allow tracking from arbitrary viewing orientations. In addition, because the perspective projection of a quadric surface is a conic, one can efficiently determine the image points lying on the boundary or silhouette of the projection of any three–dimensional model configuration [3, 19].

Figure 1 shows the edges and silhouettes corresponding to two different configurations of the hand model, each of which is seen from three different viewpoints. Because our model is designed for estimation, not visualization, precise modeling of all parts of the hand is unnecessary. As our tracking results demonstrate, it is sufficient to capture the coarse structural features which are most relevant to the observation model described in Sec. III. Note also that we do not consider model self–occlusion when finding edges. See Sec. III-A for further discussion of this approximation.

B. Kinematic Model

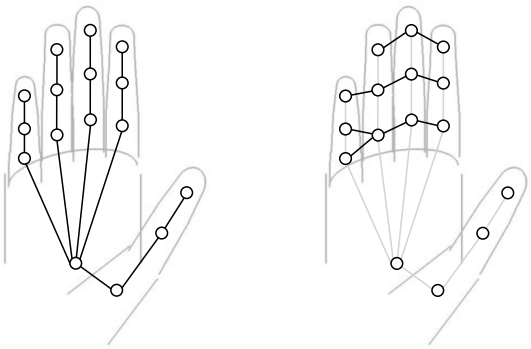
The kinematic constraints between different hand model components are well described by revolute joints [25]. Figure 2(a) shows a graph describing this kinematic structure, in which nodes correspond to rigid bodies and edges to joints. The two joints connecting the phalanges of each finger and thumb have a single rotational degree of freedom, while the joints connecting the base of each finger to the palm have two degrees of freedom (corresponding to grasping and spreading motions). Thus, twenty joint angles are required to describe the relative positions of all hand parts.

The full configuration of the hand is described by these angles along with the palm's global position and orientation, giving a total of 26 degrees of freedom. Given image measurements, calculation of a model configuration's likelihood generally requires the global position and orientation of each component. This forward kinematics problem is easily solved via a series of transformations derived from the position and orientation of each joint axis, along with the corresponding joint angles (see, for example, [12] for details).

C. Redundant Local State Representation

Most model–based hand trackers parameterize the model state in terms of the twenty joint angles described above, along with the palm's global position and orientation. In this paper, we instead explore a redundant representation in which the i^{th} rigid body is described by its position q_i and orientation r_i (a unit quaternion). Let $x_i = (q_i, r_i)$ denote this *local* description of each hand component's configuration, and $x = \{x_1, \ldots, x_{16}\}$ the configuration of the entire hand.

Clearly, there are dependencies among the elements of x implied by the kinematic constraints. Let $\mathcal{E}_K$ be the set of all



(a) (b)

Fig. 2. Graphs describing the hand model's physical constraints, where nodes correspond to different hand components. (a) Kinematic constraints corresponding to the revolute joints between neighboring components. (b) Structural constraints which prevent the intersection of hand components in three–dimensional space.

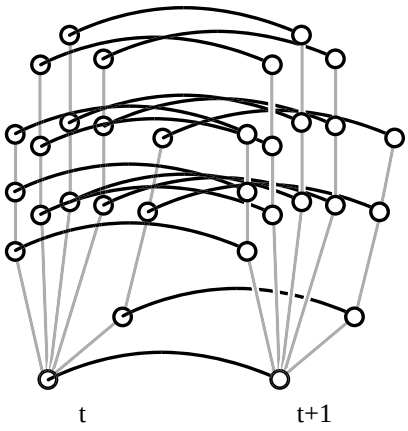


Fig. 3. Graphical model of the dynamics relating two consecutive time steps. For clarity, edges corresponding to structural potentials are not shown.

pairs of rigid bodies which are connected by joints, or equivalently the edges in the kinematic graph of Fig. 2(a). For each joint $(i,j) \in \mathcal{E}_K$, define an indicator function $\psi^K_{i,j}(x_i, x_j)$ which is equal to one if the pair (x_i, x_j) are valid rigid body configurations associated with *some* setting of the angles of joint (i,j), and zero otherwise. Viewing the component configurations x_i as random variables to be estimated, the following prior model explicitly enforces all of the constraints implied by the original joint angle representation:

$$p_K(x) \propto \prod_{(i,j)\in\mathcal{E}_K} \psi^K_{i,j}(x_i, x_j) \tag{1}$$

The structure of eq. (1) shows that $p_K(x)$ is a graphical model (in particular, a pairwise Markov random field). The graph describing the kinematic structure (Fig. 2(a)) is the same as the graph describing the Markov structure of $p_K(x)$. Intuitively, this graph expresses the fact that conditioned on the configuration of the palm, the position and orientation of each finger is described by an independent set of joint angles, and is thus statistically independent.

At first glance, the local representation described in this section may seem unattractive: the state dimension has increased from 26 to 96, and inference algorithms must now explicitly deal with the prior constraints described by $p_K(x)$. However, as we show in the following sections, local encoding of the model state greatly simplifies many other aspects of the tracking problem.

D. Structural Constraints

In reality, the joint angles describing hand configuration are not independent because different fingers can never occupy the same physical volume. The constraints that this places on joint angles are a complex function of the hand's geometry, and are difficult to express compactly. However, in the local representation of the previous section, these structural constraints take a simple form: the position and orientation of every pair of rigid bodies must be such that their component quadric surfaces do not intersect.

For computational efficiency, our tracking algorithm approximates this ideal constraint in two ways. First, we only explicitly constrain those pairs of rigid bodies which are most likely to intersect, corresponding to the edges $\mathcal{E}_S$ of the graph in Fig. 2(b). Furthermore, because the relative orientation of each finger's quadrics is implicitly constrained by the kinematic prior $p_K(x)$, we may detect most intersections based on the distance between object centroids:

$$\psi^S_{i,j}(x_i, x_j) = \begin{cases} 1 & ||q_i - q_j|| > \delta_{i,j} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Here, $\delta_{i,j}$ is a threshold determined from the radii of the cones or cylinders defining rigid bodies i and j. As for the kinematic constraints, we define a prior model which ensures that the structural constraints are not violated:

$$p_S(x) \propto \prod_{(i,j)\in\mathcal{E}_S} \psi^S_{i,j}(x_i, x_j) \tag{3}$$

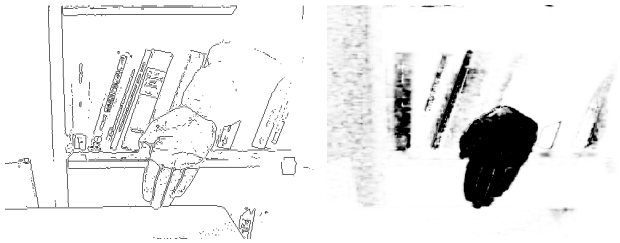
We have found this constraint to be important in our simulations to prevent different fingers from attempting to track the same image data.

E. Temporal Constraints

Thus far, our discussion has focused on the hand constraints present at a single point in time. In order to track hand motion, we must have some model of the hand's dynamics. Let $x_{t,i}$ denote the position and orientation of the i^{th} hand component at time t, and $x_t = \{x_{t,1}, \ldots, x_{t,16}\}$. For each component at time t, our dynamical model adds a Gaussian potential connecting it to the corresponding component at the previous time step:

$$p_T(x_t \mid x_{t-1}) = \prod_{i=1}^{16} \mathcal{N}(x_{t-1,i} - x_{t,i}; 0, \Lambda_i) \tag{4}$$

A graphical representation of these potentials is given in Fig. 3. Although this temporal model is factorized, the kinematic constraints at the following time step implicitly couple the



(a) (b)

Fig. 4. Image evidence used for tracking. (a) Intensity edges detected by a thresholded gradient operator. (b) Likelihood ratios at each pixel for a color–based skin detector.

corresponding random walks. These dynamics can be justified as the maximum entropy model given observations of the nodes' marginal variances Λ_i.

III. Observation Model

Our hand tracking system is based on a set of efficiently computed edge and color cues. For notational simplicity, we focus on a single video frame for the remainder of this section.

A. Edge Matching Using the Chamfer Distance

As a hand is moved in front of a camera, it obscures the background scene and thus tends to produce intensity edges along the boundaries of its projection in the image plane (see Fig. 4(a)). This edge cue is used by virtually all model–based hand tracking systems [11, 14, 19, 20, 24, 26]. Following [20], we use the Chamfer distance to measure discrepancies between projected model edges and image edges detected by a simple gradient operator. To improve accuracy, we measure distance in terms of both edge position and orientation.

Let $\Pi(x)$ denote the set of edges in the projection of three–dimensional model configuration x, and $\Delta(y)$ the output of an edge detector on the image y. The Chamfer distance $d_E(\Pi(x), \Delta(y))$ is then given by

$$d_E^2(\Pi(x), \Delta(y)) = \sum_{u \in \Pi(x)} \left[\min_{v \in \Delta(y)} g^2(u, v) \right] \tag{5}$$

Here, $g(u, v)$ determines the metric by which errors in edge matches are measured. Letting $u = (u_p, u_\theta)$ denote the position u_p and orientation u_θ of edge u, we define

$$g^2(u, v) = \min \left(\frac{||u_p - v_p||^2}{\sigma^2} + d_\pi^2(u_\theta, v_\theta), g_0 \right) \tag{6}$$

where $d_\pi(u_\theta, v_\theta)$ measures absolute differences in orientation modulo π, and g_0 adds robustness to edge detection failures. Finally, we associate this distance with a likelihood function as follows:

$$p_E(y|x) \propto \exp\left\{ -\lambda_E d_E^2(\Pi(x), \Delta(y)) \right\} \tag{7}$$

For a discussion of the generative model underlying this likelihood function, see [23].

B. Silhouette Matching Using Skin Color Statistics

Skin colored pixels are well known to have predictable statistics [9], and thus provide a powerful cue for hand tracking. We model the color distribution p_{skin} of skin pixels by a single Gaussian in RGB space, with mean and covariance estimated from hand–selected training patches. We assume that non–skin pixels have a uniform color distribution p_{bkgd}.

Let $\Omega(x)$ denote the set of pixels in the silhouette of projected hand model configuration x, and Υ the set of all image pixels. Assuming each pixel is independent, the likelihood of an image y is

$$\begin{aligned} p_C(y|x) &= \prod_{u \in \Omega(x)} p_{\text{skin}}(u) \prod_{v \in \Upsilon \setminus \Omega(x)} p_{\text{bkgd}}(v) \\ &\propto \prod_{u \in \Omega(x)} \frac{p_{\text{skin}}(u)}{p_{\text{bkgd}}(u)} \end{aligned} \tag{8}$$

The second equation follows by neglecting the proportionality constant $\prod_{v \in \Upsilon} p_{\text{bkgd}}(v)$, which is independent of x [4]. Note that we must only evaluate the likelihood ratio over the silhouette region $\Omega(x)$. Figure 4(b) plots these likelihood ratios for a sample hand image.

C. Local Decomposition of Likelihoods

Suppose that the hand model is in a three–dimensional configuration for which there is no self–occlusion. In this case, each hand component will project to a disjoint subset of the image pixels, and the Chamfer distance (eq. (5)) decomposes as

$$d_E^2(\Pi(x), \Delta(y)) = \sum_{i=1}^{16} d_E^2(\Pi(x_i), \Delta(y)) \tag{9}$$

This in turn implies that the edge–based likelihood (eq. (7)) factorizes into a product of terms which provide independent, local evidence for each component:

$$p_E(y|x) \propto \prod_{i=1}^{16} p_E(y|x_i) \tag{10}$$

Similarly, the skin color likelihood (eq. (8)) decomposes as

$$p_C(y|x) \propto \prod_{i=1}^{16} p_C(y|x_i) \tag{11}$$

Note that this statistical decomposition does *not* hold for the original joint angle representation, and is heavily dependent on our choice of a state representation in which the relationship between model parameters and image coordinates is local.

In cases where there is self–occlusion, the local decomposition of eq. (10, 11) will not hold. Nevertheless, we believe that this decomposition will often provide a good approximation. In particular, because occlusion reasoning can only reduce the number of projected model edges, the local decomposition of eq. (10) will always provide an upper bound on the true edge likelihood $p_E(y|x)$.

D. Fast Likelihood Computation

Because the nonparametric belief propagation algorithm proposed in this paper must evaluate many different hypotheses for each model component, it is important that the evaluations of our likelihood functions be computationally efficient. For the skin color term (eq. (8)), we precompute the cumulative sum of the log likelihood ratios along each row of pixels. We may then quickly integrate the likelihood of each hypothesized silhouette region, given only the boundaries of that silhouette.

For the Chamfer distance, our inclusion of orientation information makes it difficult to use standard distance transform methods. We instead use KD–trees [2] to exploit the geometric structure underlying our detected edges. For low–dimensional collections of points, KD–trees may be efficiently constructed, and then used to find nearest neighbors in logarithmic time.

Given a set of detected edges, we precompute a KD–tree representation of the three–dimensional vectors corresponding to each edge's position and orientation. To account for the fact that orientation distance must be measured modulo π, we also include a second, appropriately rotated copy of each point. Then, for each hypothesized model configuration, the minimization step of the Chamfer distance computation (eq. (5)) can be performed via efficient nearest–neighbor search in the KD–tree. Using KD–trees, we achieve very fast Chamfer distance computation without requiring excess storage or suffering from discretization artifacts.

IV. Nonparametric Belief Propagation

A. Graphical Models and Belief Propagation

In the previous sections, we have shown that a redundant, local representation of the geometric hand model's configuration x_t allows $p\left(x_t \mid y_t\right)$, the posterior distribution of the hand model at time t given image observations y_t, to be written as

$$p(x_t|y_t) \propto p_K(x_t)p_S(x_t)\left[\prod_{i=1}^{16} p_E(y_t|x_{t,i})p_C(y_t|x_{t,i})\right] \quad (12)$$

where $p_K(x_t)$ and $p_S(x_t)$ are kinematic and structural prior models corresponding to the graphs of Fig. 2. This expression is exact when there is no self–occlusion, and a potentially useful approximation more generally. When T video frames are observed, the overall posterior distribution is given by

$$p\left(x \mid y\right) \propto \prod_{t=1}^{T} p\left(x_t \mid y_t\right) p_T(x_t \mid x_{t-1}) \quad (13)$$

Equation (13) is an example of a pairwise Markov random field, which can more generally be written as

$$p(x|y) \propto \prod_{(i,j)\in\mathcal{E}} \psi_{i,j}\left(x_i, x_j\right) \prod_{i\in\mathcal{V}} \psi_i\left(x_i, y\right) \quad (14)$$

Here, $\mathcal{V}$ is a set of nodes, corresponding to the sixteen components of the hand model at each time step, and $\mathcal{E}$ is a set of edges specifying their statistical dependencies.

Given our analysis, hand tracking can be seen as a special example of inference in a graphical model. In this paper, we consider *belief propagation* (BP) [27], a method for solving inference problems via local message–passing. At each iteration of the BP algorithm, some node $i \in \mathcal{V}$ calculates a message $m_{ij}\left(x_j\right)$ to be sent to some neighboring node $j \in \Gamma(i) \triangleq \{j \mid (i,j) \in \mathcal{E}\}$:

$$m_{ij}^n\left(x_j\right) = \alpha \int_{x_i} \psi_{j,i}\left(x_j, x_i\right) \psi_i\left(x_i, y\right) \times \prod_{k\in\Gamma(i)\setminus j} m_{ki}^{n-1}\left(x_i\right) dx_i \quad (15)$$

Here, α denotes an arbitrary proportionality constant. At any iteration, each node can produce an approximation $\hat{p}(x_i \mid y)$ to the marginal distribution $p\left(x_i \mid y\right)$ by combining the incoming messages with the local observation:

$$\hat{p}^n(x_i \mid y) = \alpha\psi_i\left(x_i, y_i\right) \prod_{j\in\Gamma(i)} m_{ji}^n\left(x_i\right) \quad (16)$$

For tree–structured graphs, the approximate marginals, or beliefs, $\hat{p}^n(x_i \mid y)$ will converge to the true marginals $p\left(x_i \mid y\right)$ once the messages from each node have propagated to every other node in the graph. On graphs with cycles, the marginal distributions estimated by BP are only approximate, but these approximations are often highly accurate [27].

B. Nonparametric Representations

For the hand tracking problem, the variables x_i take on continuous values. Because accurate discretization of the six degrees of freedom at each node is intractable, and the BP message update (eq. (15)) has no closed form for the potentials underlying hand tracking, exact implementation of BP is infeasible. Instead, we explore nonparametric, particle–based approximations to these messages using the nonparametric belief propagation (NBP) algorithm [21].

In NBP, each message is represented using either a sample–based density estimate (a mixture of Gaussians) or an analytic function. Both types of messages are needed for hand tracking, as we discuss below. Each NBP message update involves two stages: sampling from the estimated marginal, followed by Monte Carlo approximation of the outgoing message. For the general form of these updates, see [21]. In the following sections, we give a high–level overview focusing on the unique features of the hand tracking application.

The hand tracking application is complicated by the fact that the orientation component r_i of $x_i = (q_i, r_i)$ is an element of the rotation group $SO(3)$. Following [5, 17], we represent orientations as unit quaternions, and use a linearized approximation when constructing density estimates. Any sampled orientations may be projected back to $SO(3)$ by normalizing the corresponding four–dimensional vector. This approximation is most appropriate for densities with tightly concentrated rotational components.

C. Marginal Computation

From eq. (16), we see that the BP estimate of the local marginal distribution $\hat{p}(x_i \mid y)$ is equal to the product of the

Given input messages $m_{ji}(x_i)$ from kinematic neighbors $\Gamma_K(i)$, structural neighbors $\Gamma_S(i)$, and temporal neighbors $\Gamma_T(i)$:

1) Draw M independent samples $\{x_i^{(\ell)}\}_{\ell=1}^M$ from the product

$$x_i^{(\ell)} \sim \prod_{j \in \Gamma_T(i)} m_{ji}(x_i) \prod_{k \in \Gamma_K(i)} m_{ki}(x_i)$$

using the multiscale sampling methods of [7].
2) For each $x_i^{(\ell)} = (q_i^{(\ell)}, r_i^{(\ell)})$, normalize the orientation $r_i^{(\ell)}$.
3) Compute an importance weight for each sample $x_i^{(\ell)}$:

$$w_i^{(\ell)} \propto p_E(y|x_i^{(\ell)}) p_C(y|x_i^{(\ell)}) \prod_{j \in \Gamma_S(i)} m_{ji}(x_i^{(\ell)})$$

4) Use a bandwidth selection method (see [18]) to construct a kernel density estimate $\hat{p}(x_i \mid y)$ from $\{x_i^{(\ell)}, w_i^{(\ell)}\}_{\ell=1}^M$.

Alg. 1. NBP update of the estimated marginal distribution $\hat{p}(x_i \mid y)$.

Given M weighted samples $\{x_i^{(\ell)}, w_i^{(\ell)}\}_{\ell=1}^M$ from $\hat{p}(x_i \mid y)$, and the incoming message $m_{ji}(x_i)$ used to construct $\hat{p}(x_i \mid y)$:

1) Reweight each sample $x_i^{(\ell)}$ as $\bar{w}_i^{(\ell)} \propto w_i^{(\ell)} / m_{ji}(x_i^{(\ell)})$.

KINEMATIC EDGES:

2) Draw M samples $\{\bar{x}_i^{(\ell)}\}_{\ell=1}^M$ with replacement from the discrete distribution defined by the weights $\{\bar{w}_i^{(\ell)}\}_{\ell=1}^M$.
3) For each $\bar{x}_i^{(\ell)}$, sample uniformly from the allowable angles for joint (i,j). Determine $x_j^{(\ell)}$ via forward kinematics.
4) Use a bandwidth selection method to construct a kernel density estimate $m_{ij}(x_j)$ from the unweighted samples $\{x_j^{(\ell)}\}_{\ell=1}^M$.

TEMPORAL EDGES:

2) Construct a kernel density estimate $m_{ij}(x_j)$ with centers $\{x_i^{(\ell)}\}_{\ell=1}^M$, weights $\{\bar{w}_i^{(\ell)}\}_{\ell=1}^M$, and uniform bandwidths Λ_i.

STRUCTURAL EDGES:

2) For any $x_j = (q_j, r_j)$, let $\mathcal{L} = \{\ell \mid ||q_i^{(\ell)} - q_j|| > \delta_{i,j}\}$.
3) Calculate $m_{ij}(x_j) = \sum_{\ell \in \mathcal{L}} \bar{w}_i^{(\ell)}$.

Alg. 2. NBP update of the nonparametric message $m_{ij}(x_j)$ sent from node i to node j as in eq. (17), for each of the three potential types.

incoming messages from neighboring nodes with the local observation potential. Like particle filters, NBP uses importance sampling to approximate this product. As we describe in the following section, our NBP hand tracker employs Gaussian mixtures for some messages (along kinematic and temporal edges), and analytic functions for others (structural edges). The image likelihood $p_E(y|x_i)p_C(y|x_i)$ is an analytic function which can be efficiently evaluated at any candidate x_i using the methods of Sec. III-D.

The importance sampling update of the marginal estimate $\hat{p}(x_i \mid y)$ is summarized in Alg. 1. First, M samples $\{x_i^{(\ell)}\}_{\ell=1}^M$ are drawn directly from the product of the kinematic and temporal Gaussian mixture messages. Note that this sampling problem is nontrivial: given d mixtures of M Gaussians, their product is a mixture of M^d Gaussians. However, in this paper we use a recently proposed multiscale Gibbs sampler [7] to efficiently draw accurate, albeit approximate, samples. Following normalization of the rotational component, each sample $x_i^{(\ell)}$ is assigned a weight $w_i^{(\ell)}$ equal to the product of the color and edge likelihoods with any messages along structural edges. Finally, the computationally efficient "rule of thumb" heuristic [18] is used to set the bandwidth of Gaussian smoothing kernels placed around each sample, producing an estimate of the desired marginal distribution.

The previous procedure assumes that at least one of the incoming messages is a Gaussian mixture. For the hand tracker, this is true except for the initial message updates on the first frame, when the only incoming message is the local analytic likelihood function. For the simulations presented in this paper, we initialized the tracker by hand–specifying a high variance Gaussian proposal distribution centered roughly around the true starting hand configuration. In the future, we hope to replace this manual initialization by automatic image–based feature detectors.

D. Message Propagation and Scheduling

To derive the message propagation rule, as suggested by [10] we rewrite the message update equation (15) in terms of the marginal distribution $\hat{p}(x_i \mid y)$:

$$m_{ij}^n(x_j) = \alpha \int_{x_i} \psi_{j,i}(x_j, x_i) \frac{\hat{p}^{n-1}(x_i \mid y)}{m_{ji}^{n-1}(x_i)} dx_i \qquad (17)$$

Our explicit use of the current marginal estimate $\hat{p}^{n-1}(x_i \mid y)$ helps focus the Monte Carlo approximation on the most important regions of the state space.

Consider first the case where $(i,j) \in \mathcal{E}_K$, so that $\psi_{j,i}^K$ corresponds to a kinematic constraint. The message propagation step makes direct use of the particles $\{x_i^{(\ell)}\}_{\ell=1}^M$ sampled during the last marginal estimate. We reweight each particle $x_i^{(\ell)}$ by $1/m_{ji}(x_i^{(\ell)})$, and then resample to get M unweighted particles $\{\bar{x}_i^{(\ell)}\}_{\ell=1}^M$ (see Alg. 2). We must then sample candidate x_j configurations from the conditional distribution $\psi_{j,i}^K(x_j, \bar{x}_i^{(\ell)})$. Because $\psi_{j,i}^K$ is an indicator potential, this sampling has a particularly appealing form: first sample uniformly among allowable joint angles, and then use forward kinmatics to find the $x_j^{(\ell)}$ corresponding to each $\bar{x}_i^{(\ell)}$. Finally, the "rule of thumb" bandwidth selection method [18] is used to construct the outgoing Gaussian mixture message.

Because the temporal constraint potentials are Gaussian, the sampling associated with kinematic message updates is unnecessary. Instead, as suggested by [8], we simply adjust the bandwidths of the current marginal estimate $\hat{p}(x_i \mid y)$ to match the temporal covariance Λ_i (see Alg. 2). This update implicitly assumes that the bandwidth of $\hat{p}(x_i \mid y)$ is much smaller than Λ_i, which will hold for sufficiently large M.

For structural constraint edges $\mathcal{E}_S$, a different approach is needed. In particular, from eq. (2) we see that the pairwise potential is one for all state configurations outside some ball, and therefore the outgoing message will not be finitely integrable. For structural edges, messages must then take the form of analytic functions. In principle, at some point x_j the message $m_{ij}(x_j)$ should equal the integral of $\hat{p}(x_i \mid y)/m_{ji}(x_i)$ over all configurations outside some ball centered at q_j. We

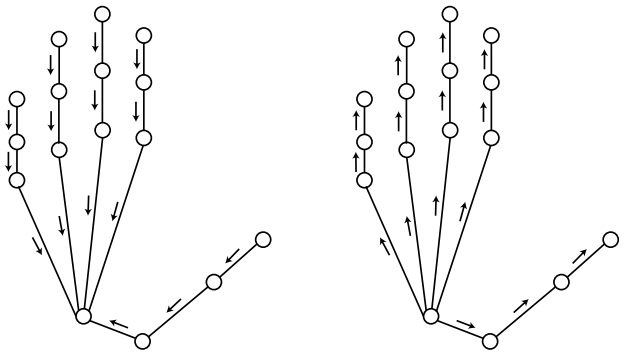


Fig. 5. Scheduling of the kinematic constraint message updates for NBP: messages are first passed from fingertips to the palm, and then back to the fingertips. Structural constraint messages (not shown) are updated as needed.

approximate this quantity by the sum of the weights of all kernels in $\hat{p}(x_i \mid y)$ outside that ball (see Alg. 2).

For NBP, the message update order effects the outcome of each local Monte Carlo approximation, and may thus effect the quality of the final marginal estimates. Given a single frame, we iterate the tree–based message schedule of Fig. 5, in which messages are passed from fingertips to the palm, and then back to the fingertips. The structural messages, which for clarity are not shown, are also updated whenever the source node's belief changes. For video, we process the frames in sequence, updating the temporal messages to the next frame following a fixed number of kinematic message sweeps. However, the tracker could be easily extended to incorporate information from future video frames using reverse–time messages.

E. Related Work

The NBP algorithm has also recently been used to develop a three–dimensional person tracker [17]. However, this person tracker uses a "loose–limbed" formulation of the kinematic constraints which differs significantly from our hand tracker. In particular, the loose–limbed tracker represents the conditional distribution of each limb's location given its neighbor via a Gaussian mixture estimated from training data. For each joint, the two needed conditional densities (for example, upper arm given lower arm and lower arm given upper arm) are learned independently. In general, however, there may be no pairwise clique potential which is consistent with these conditionals. Thus, there may be no globally consistent generative model underlying their results, making the standard theoretical justifications of belief propagation inapplicable. The two–dimensional tracking results of [8, 24] are also based on explicit (and sometimes inconsistent) relaxations of the true kinematic constraints.

In contrast, we have shown that an NBP tracker may be built around the local structure of the true kinematic constraints. Conceptually, this has the advantage of providing a clearly specified, globally consistent generative model whose properties can be analyzed. Practically, our formulation avoids the need to explicitly approximate the kinematic constraints, and allows us to build a functional tracker without the need for training data.

V. Simulations

In this section, we examine the empirical performance of the NBP hand tracker. All results are based on 720×480 images (or video sequences) recorded by a calibrated camera. The physical dimensions of the quadrics composing the hand model were measured offline. All messages were represented by $M = 200$ particles, and the result figures show the projections of the final density estimates' five largest modes.

A. Refinement of Coarse Initializations

Given a single image, NBP may be used to progressively refine a coarse, user–supplied initialization into an accurate estimation of the hand's configuration. See Fig. 6 for two examples of such a refinement. In the second example, note that the initial finger positions are not only misaligned, but the user has supplied no information about the grasping configuration of the hand. By the fourth NBP iteration, however, the system has aligned all of the joints properly. In both images, a poorly aligned palm is eventually attracted to the proper location by well–fit fingers. For these examples, each NBP iteration (a complete update of all messages in the graph) requires about 1 minute on a Pentium IV workstation.

B. Temporal Tracking

Two video sequences demonstrating the NBP hand tracker are available at `http://ssg.mit.edu/nbp/`. Total computation time for each video sequence, including all likelihood calculations, is approximately 4 minutes per frame. The first shows the hand rigidly moving in three–dimensional space. The extrema of this motion are shown in Fig. 7. The NBP estimates closely track the hand throughout the sequence, but are noisiest when the fingers point towards the camera because the sharp projection angle reduces the amount of image evidence. Note, however, that the estimates quickly lock back onto the true hand configuration when the hand rotates away from the camera.

The second video sequence exercises the hand model's joints, containing both individual finger motions and combined grasping motions (see Fig. 8). Our model supports all of these degrees of freedom, and maintains accurate estimates even when the ring finger is partially occluded by the middle finger (bottom row of Fig. 8). This robustness to moderate occlusions comes from our use of structural potentials to prevent self–intersection, and is only reliable when the hand's motion is well predicted by the dynamical model.

VI. Discussion

We have demonstrated that the geometric models commonly used for hand tracking naturally have a graphical structure, and exploited this fact to build an effective hand tracking algorithm using nonparametric belief propagation. We are currently investigating more challenging test sequences, as well as a rigorous comparison of our algorithm to existing methods. Preliminary results indicate that accurate tracking through significant self–occlusion will require a more sophisticated local likelihood approximation, as well as richer

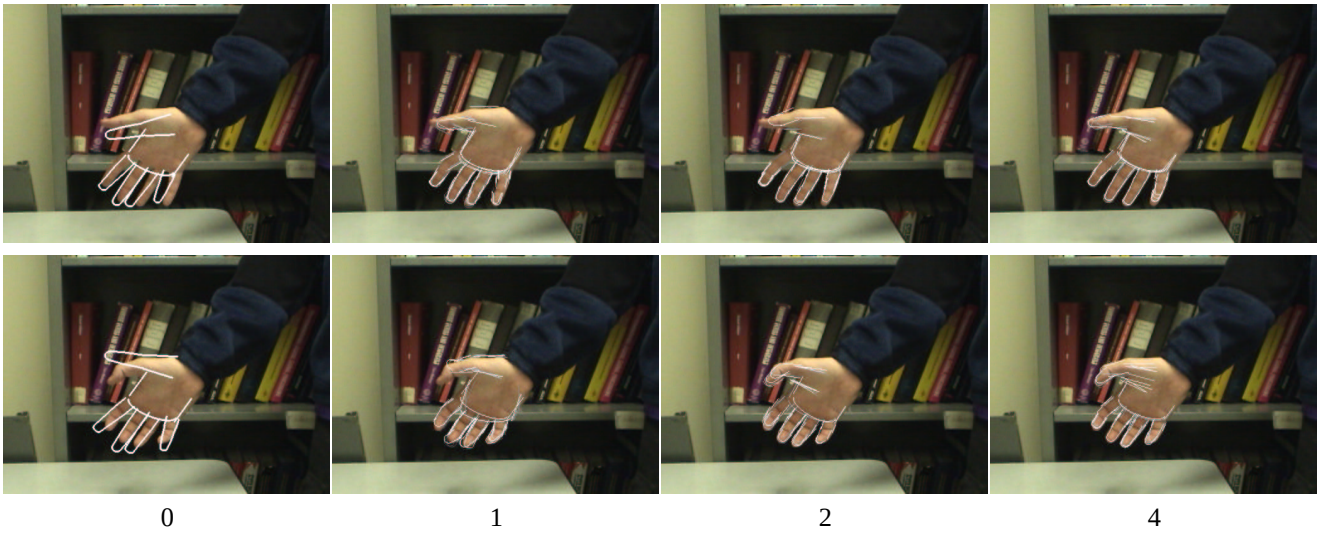


Fig. 6. Two examples of refinement of a coarse hand model initialization using NBP. We show results following 1, 2, and 4 iterations of the message schedule in Fig. 5. Plots show the projections of the most significant marginal modes.

Fig. 7. Four frames showing extrema of a hand's rigid motion, and the position estimates produced by the NBP tracker.

dynamical models. In addition, we hope to use local hand feature detectors to improve our method's robustness.

ACKNOWLEDGMENTS

The authors thank C. Mario Christoudias and Michael Siracusa for their help with video data collection. We would also like to thank Michael Black, Alexander Ihler, Michael Isard, and Leonid Sigal for helpful conversations. This research was supported in part by AFOSR Grant F49620-00-1-0362.

REFERENCES

[1] V. Athitsos and S. Sclaroff, "Estimating 3D hand pose from a cluttered image," in *CVPR*, vol. 2, 2003, pp. 432–439.

[2] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Comm. ACM*, vol. 18, no. 9, pp. 509–517, Sept. 1975.

[3] J. Blinn, "The algebraic properties of second–order surfaces," in *Introduction to Implicit Surfaces*, J. Bloomenthal, Ed. Morgan Kaufmann, 1997, pp. 52–97.

[4] J. M. Coughlan and S. J. Ferreira, "Finding deformable shapes using loopy belief propagation," in *ECCV*, vol. 3, 2002, pp. 453–468.

[5] J. Deutscher, M. Isard, and J. MacCormick, "Automatic camera calibration from a single Manhattan image," in *ECCV*, vol. 4, 2002, pp. 161–174.

[6] A. Doucet, N. de Freitas, and N. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*. New York: Springer-Verlag, 2001.

[7] A. T. Ihler, E. B. Sudderth, W. T. Freeman, and A. S. Willsky, "Efficient multiscale sampling from products of Gaussian mixtures," in *NIPS*, 2003.

[8] M. Isard, "PAMPAS: Real–valued graphical models for computer vision," in *CVPR*, vol. 1, 2003, pp. 613–620.

[9] M. J. Jones and J. M. Rehg, "Statistical color models with application to skin detection," *IJCV*, vol. 46, no. 1, pp. 81–96, 2002.

[10] D. Koller, U. Lerner, and D. Angelov, "A general algorithm for approximate inference and its application to hybrid Bayes nets," in *UAI 15*, 1999, pp. 324–333.

[11] J. MacCormick and M. Isard, "Partitioned sampling, articulated objects, and interface–quality hand tracking," in *ECCV*, vol. 2, 2000, pp. 3–19.

[12] I. Mikic, M. Trivedi, E. Hunter, and P. Cosman, "Human body model acquisition and tracking using voxel data," *IJCV*, vol. 53, no. 3, pp. 199–223, 2003.

[13] D. Ramanan and D. A. Forsyth, "Finding and tracking people from the bottom up," in *CVPR*, vol. 2, 2003, pp. 467–474.

[14] J. M. Rehg and T. Kanade, "DigitEyes: Vision–based hand tracking for human–computer interaction," in *Proc. IEEE Workshop on Non–Rigid and Articulated Objects*, 1994.

[15] G. Shakhnarovich, P. Viola, and T. Darrell, "Fast pose estimation with parameter sensitive hashing," in *ICCV*, 2003, pp. 750–757.

[16] H. Sidenbladh, M. J. Black, and L. Sigal, "Implicit probabilistic models of human motion for synthesis and tracking," in *ECCV*, vol. 1, 2002, pp. 784–800.

[17] L. Sigal, M. Isard, B. H. Sigelman, and M. J. Black, "Attractive people: Assembling loose–limbed models using nonparametric belief propagation," in *NIPS*, 2003.

[18] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*. London: Chapman & Hall, 1986.

[19] B. Stenger, P. R. S. Mendonca, and R. Cipolla, "Model–based 3D tracking of an articulated hand," in *CVPR*, vol. 2, 2001, pp. 310–315.

[20] B. Stenger, A. Thayananthan, P. H. S. Torr, and R. Cipolla, "Filtering using a tree–based estimator," in *ICCV*, 2003, pp. 1063–1070.

[21] E. B. Sudderth, A. T. Ihler, W. T. Freeman, and A. S. Willsky, "Nonparametric belief propagation," in *CVPR*, vol. 1, 2003, pp. 605–612.

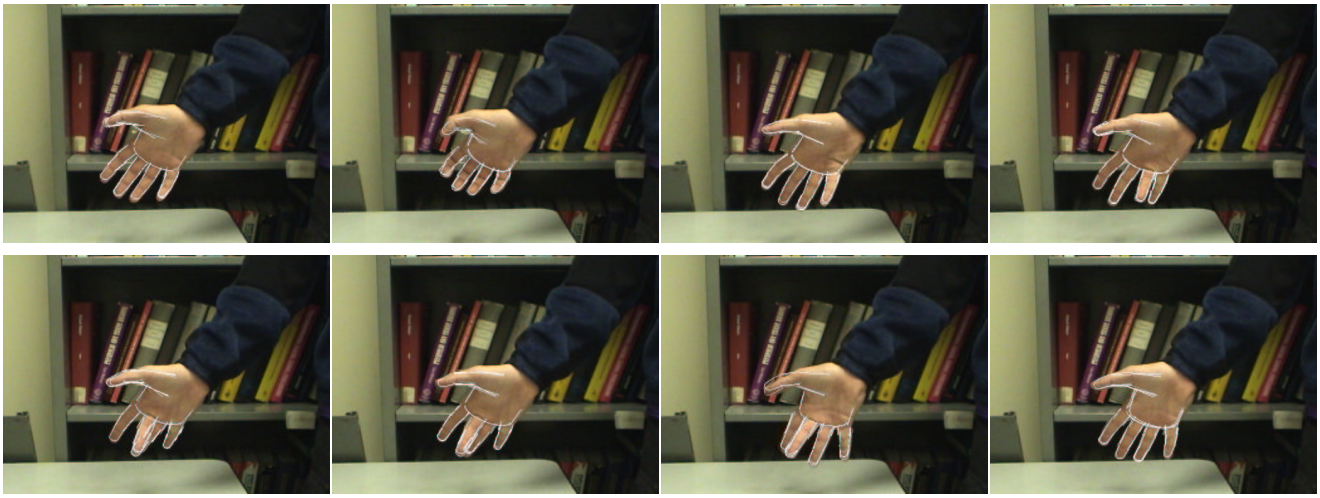


Fig. 8. Eight frames from a tracking sequence in which the hand makes grasping motions and individual finger movements. Note that the ring finger is accurately tracked even through a partial occlusion by the middle finger (bottom row).

[22] C. Tomasi, S. Petrov, and A. Sastry, "3D Tracking = Classification + Interpolation," in *ICCV*, 2003, pp. 1441–1448.

[23] K. Toyama and A. Blake, "Probabilistic tracking with exemplars in a metric space," *IJCV*, vol. 48, no. 1, pp. 9–19, 2002.

[24] Y. Wu, G. Hua, and T. Yu, "Tracking articulated body by dynamic Markov network," in *ICCV*, 2003, pp. 1094–1101.

[25] Y. Wu and T. S. Huang, "Hand modeling, analysis, and recognition," *IEEE Signal Proc. Mag.*, pp. 51–60, May 2001.

[26] Y. Wu, J. Y. Lin, and T. S. Huang, "Capturing natural hand articulation," in *ICCV*, 2001.

[27] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Constructing free energy approximations and generalized belief propagation algorithms," MERL TR2002-35.