Tracking Articulated Hand Motion with Eigen Dynamics Analysis

Hanning Zhou and Thomas S. Huang
ECE Department, University of Illinois at Urbana-Champaign
Urbana, IL 61801
email: {hzhou,huang}@ifp.uiuc.edu

Abstract

This paper introduces the concept of eigen-dynamics and proposes an eigen dynamics analysis (EDA) method to learn the dynamics of natural hand motion from labelled sets of motion captured with a data glove. The result is parameterized with a high-order stochastic linear dynamic system (LDS) consisting of five lower-order LDS. Each corresponding to one eigen-dynamics. Based on the EDA model, we construct a dynamic Bayesian network (DBN) to analyze the generative process of a image sequence of natural hand motion. Using the DBN, a hand tracking system is implemented. Experiments on both synthesized and real-world data demonstrate the robustness and effectiveness of these techniques.

1 Introduction

The methods that have been proposed for tracking hand motion could be divided into two categories. One is appearance-based. Some tried to establish a mapping between the image feature space and the hand configuration space [4, 33, 25]. The other is model-based. Deformable hand shape models are fitted with statistical methods such as local principal component analysis (PCA) [11, 12] and sequential Monte Carlo [13]. 3D kinematic models are used in [16, 24, 23, 6, 34]. Recently, the idea of tracking-by-detection merge these two categories by doing exhaustive search in large databases of 2D templates [27, 1]. Stenger *et al*. proposed a tree-based filtering to discretize the finger configuration space for faster searching [28, 29].

The aim of this paper is to study the *dynamics* of natural hand motion, which can be used both in the context of tracking and in building structurally optimized template database for fast detection. Dynamic models have been widely used in tracking [22], classification [19, 21] and synthesis [35] of human body motion. Ghahramani [9] proposed a DBN framework for learning and inference in one class of switching linear dynamic system (SLDS) models. North et al. [19] proposed a framework of switching particle filters to learn multi-class dynamics. Although these methods could be applied to hand motion, they are not specifically tailored for the dynamics of human hand.

We define the concept of eigen-dynamics and propose an *eigen dynamics analysis* (EDA) method to learn the dynamics of natural hand motion as a high order stochastic LDS consisting of five decoupled lower order subsystems. Each corresponds to one eigen-dynamics. Based on the dynamic model, we introduce a DBN framework for tracking articulated hand motion, which incorporates a kinematic hand model, finger dynamics, color models and image observations. Using this DBN, we implemented a robust and effective system for tracking natural hand motion from a monocular view. In the observation phase, a new feature called *likelihood edge* is extracted. In the inference phase, we decompose hand motion into global motion and finger articulation and solve them iteratively in a divide-and-conquer fashion [32]. For global motion, we apply iterative closest point (ICP) algorithm[36]. For tracking finger articulation, we apply sequential Monte Carlo [18] to sample in the manifold spanned by the learned dynamic model.

Section 2 proposes the EDA method. Section 3 describes the DBN for tracking global and local hand motion. Section 4 introduces the new feature called likelihood edge. Section 5 describes the ICP based global hand tracking algorithm. Section 6 describes the inference of the finger articulation based on factored sampling. Section 7 provides experimental results in both quantitative and visual forms. Section 8 summaries our contributions and limitations of the system.

2 Modelling Hand Dynamics with EDA

The finger configuration $\mathbf{Z}[k]$ is represented by 20 joint angles based the kinematic model we use, where each finger is a kinematic chain with four degrees of freedom. Due to geometry and biomechanics constraints, the feasible finger configurations lie in a manifold $\mathcal{C} \subset \Re^{20}$, which we propose to model with EDA.

Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV 2003) 2-Volume Set
0-7695-1950-4/03 $17.00 © 2003 IEEE

2.1 The 10^{th} Order Stochastic LDS

Denoting the 6 dimensional PCA space by Υ, we project $\mathbf{Z}[k] \in$ Ç to $\mathbf{Y}[k] \in \Upsilon$ by

$$\mathbf{Y}[k] = U_{6\times 20}(\mathbf{Z}[k] - \mathbf{E}\{\mathbf{Z}\}) \quad (1)$$

where the rows of matrix $U_{6\times 20}$ are the eigen vectors corresponding to the six largest eigen values of the covariance matrix of $\mathbf{Z}$, and $\mathbf{E}\{\mathbf{Z}\}$ is the mean over Ç.

After PCA dimension reduction, around 99.79% of the variance is preserved. Because of the intrinsic nonlinearity, we need a high dimensional linear system to parameterize the dynamics of finger articulation. Therefore we assume $\mathbf{Y}[k]_{6\times 1}$ as the output of a 10^{th} order stochastic LDS:

$$\begin{aligned} \mathbf{X}[k+1] &= A_{10\times 10}\mathbf{X}[k] + w[k] \\ \mathbf{Y}[k] &= C_{6\times 10}\mathbf{X}[k] + v[k] \end{aligned} \quad (2)$$

where $w[k]$ and $v[k]$ are both zero mean Gaussian. Using the data collected with *CyberGlove*, we could directly train a 10^{th} order stochastic LDS with subspace identification algorithm [31], and obtain a LDS $\bar{\mathbf{S}} = \{\bar{A}_{10\times 10}, \bar{C}_{6\times 10}, \Sigma_{\bar{w}10\times 10}, \Sigma_{\bar{v}6\times 6}\}$. However, the variance in $w[k]$ and $v[k]$ are very large. In the next subsection, we will show how EDA will improve the training results by imposing structural information specific to the finger dynamics.

2.2 The Eigen Dynamics Analysis Method

We define an *eigen-dynamics* as the dynamics of intensional flexing/extending[1] one finger while the other four fingers moving autonomically. Please note that we are not considering each finger as independent to the others, because each eigen-dynamics models all the 20 joint angles in the five fingers. In each training set, all five fingers are moving in a naturally constrained way, although the major motion is performed by one of them. The conjecture is that the five eigen-dynamics span the whole manifold of Ç. To verify it, we separately trained the five LDS with the corresponding $CyberGlove$ data. Denoting the parameters for the i^{th} eigen-dynamics with $\hat{\mathbf{S}}^{(i)} = \{A^{(i)}_{2\times 2}, C^{(i)}_{6\times 2}, \Sigma_{w^{(i)}2\times 2}, \Sigma_{v^{(i)}6\times 6}\}$, we obtain the 10^{th} order stochastic LDS $\hat{\mathbf{S}}$ as

$$\begin{aligned} \hat{A}_{10\times 10} &= diag^5_{i=1}\{A^{(i)}_{2\times 2}\} & \Sigma_{\hat{w}10\times 10} &= diag^5_{i=1}\{\Sigma^{(i)}_{w\,2\times 2}\} \\ \hat{C}_{6\times 10} &= [C^{(1)}_{6\times 2}\cdots C^{(5)}_{6\times 2}] & \Sigma_{\hat{v}6\times 6} &= \textstyle\sum_{i=1}^{5} \frac{1}{5}\Sigma^{(i)}_{v\,6\times 6} \end{aligned} \quad (3)$$

The fact that $rank(\hat{C}_{6\times 10}) = 6$ proves that the five eigen-dynamics *do* span the whole manifold of Ç. The quantitative results in Section 7.1 show that $\hat{\mathbf{S}}^{(i)}$ is sufficient to capture the nonlinearity in each eigen-dynamics. EDA significantly reduce the variance in the LDS $\bar{\mathbf{S}}$, by imposing a specific structure tailored for finger dynamics, that is, $\hat{\mathbf{S}}$ consists of five decoupled subsystems and each corresponding to one eigen-dynamics. $\hat{\mathbf{S}}$ is not similar[2] to $\bar{\mathbf{S}}$. Therefore decoupling $\bar{\mathbf{S}}$ with standard SVD-based linear algebraic techniques will not give the same result as EDA. In fact, such decoupling would have the same noise variance as that of $\bar{\mathbf{S}}$, which is much larger than that of $\hat{\mathbf{S}}$.

A more careful study on $\hat{\mathbf{S}}^{(i)}$ shows that the five $A^{(i)}_{2\times 2}$ matrices are very similar. Each subsystem has two modes: one is slightly unstable[3], the other is stable. For example, $A^{(4)} = \begin{bmatrix} 1.0176 & -0.0014 \\ 0.0069 & 0.9910 \end{bmatrix}$. Physically, we could interpret the unstable component of $\mathbf{X}^{(i)}[k]$ as the joint angle, and the stable one as the angular velocity. The velocity will be driven away from zero by an impulse, and then converge back to zero as the motion stops, while the joint angles will end at a value than the starting value. This sigmoid shape captures the similar dynamics of the inner states $\mathbf{X}^{(i)}$. Nevertheless, the five $C^{(i)}_{6\times 2}$ matrices are distinct, which characterize the five distinct output spaces. These results are consistent with the biomechanics structure of the hand, that each finger moves in a similar constrained fashion, while their major motions involve different sets of joint angles.

This method of training the subsystems of a high order LDS independently with labelled motion trajectories, is defined as *eigen dynamics analysis*. The word "eigen" comes from the analogues between PCA and EDA summarized in Table 1.

Table 1. Analogues between PCA and EDA

PCA	EDA
reduce data dimensionality	reduce the order of LDS
decorrelate 2nd-order statistics	decouple the subsystems
the i^{th} component	the i^{th} eigen-dynamics
eigen value λ_i	inner state $\mathbf{X}^{(i)}$
eigen vector P_i	observation matrix $C^{(i)}_{6\times 2}$
P_i and P_j $(i \neq j)$ are orthogonal	$\mathbf{X}^{(i)}$ and $\mathbf{X}^{(j)}$ $(i \neq j)$ are statistically independent

Figure 1 illustrates how EDA method models the dynamics of $\mathbf{Y}[k]$ in Υ. The five[4] colored cubes span the whole space Υ. The dynamic constraints are modelled by the fact that $\mathbf{X}^{(i)}[k]_{2\times 1}$ can only move along (or near) the colored dashed curve in the 2D state space spanned by $A^{(i)}$. However, since the five $\mathbf{X}^{(i)}[k]$ are moving independently, the

[1] We also study the difference between the dynamics of extending and flexing the same finger. The identification results show that after reversing the time index, they are almost identical, that is, $A^{(i)}_{flexing} \times A^{(i)}_{extending} \approx I_{2\times 2}$, while $C^{(i)}_{flexing}$ and $C^{(i)}_{extending}$ are almost the same.

[2] Two LDS's are similar if there exists a similarity transformation [5] between them.

[3] A mode is unstable if the eigen value of A corresponding to that mode is larger than 1 [5].

[4] For clarity, we only draw three of them.

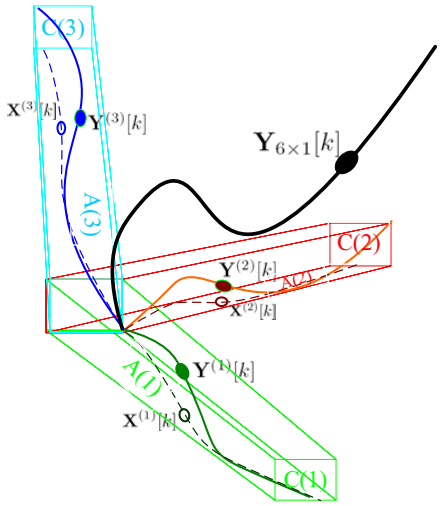

Figure 1. A conceptual illustration of the manifold spanned by EDA. For clarity, only three of the five eigen-dynamics are drawn in red, green and blue respectively.

Table 2. Notations in Figure 1

6-dim output space spanned by $C^{(i)}_{6\times2}$	colored rectangular cube
2D state space spanned by $A^{(i)}_{2\times2}$	bottom side of the colored cube
hidden states $\mathbf{X}^{(i)}[k]_{2\times1}$	small circle on the colored dashed curve
trajectory of $\mathbf{X}^{(i)}[k]$	colored dashed curve
output $\mathbf{Y}^{(i)}[k]_{6\times1}$	spheroid dot on the colored solid curve
trajectory of $\mathbf{Y}^{(i)}[k]$	colored solid curve
combined output $\mathbf{Y}[k]_{6\times1}$	spheroid dot on the black solid curve
trajectory of $\mathbf{Y}[k]$	black solid curve

combined output $\mathbf{Y}[k]$ can be very nonlinear, just as the actual motion of the hand is.

EDA method stratifies the intrinsic biomechanical nonlinearity from the apparent nonlinearity due to unsynchronized motion among different fingers. The former is modelled by the similar sigmoid-shape nonlinear dynamics of the inner states (the colored dashed curve), while the latter is modelled as the three hidden states $\mathbf{X}^{(i)}[k]$ sliding along the colored dashed curve *independently*. This stratification enables factored sampling to reduce the sampling space from the whole $\Re^{20}$ to a manifold in $\Re^{10}$ (corresponding to the five random walks along the nonlinear dashed curve induced by $A^{(i)}$). In this sense, we claim that EDA can compress the actual dimensionality of the manifold of feasible finger configurations.

To compare EDA method with other LDS-based models, we could introduce an ancillary state as the set of the labels of the *active* subsystems: $S[k] = \{i|i = \{1\dots5\}s.t.\|\mathbf{X}^{(i)}[k] - \mathbf{X}^{(i)}[k-1]\| \geq \varepsilon > 0\}$, where ε would be a threshold to specify how large motion an active subsystem should have between consecutive frames. If we further restrict $S[k]$ to have only one element, i.e. only one active subsystem at each time, EDA model becomes similar to switching linear dynamic systems (SLDS) [20] [22], which switches between different LDS. At each time, SLDS only cover one of the subspaces induced by the $\hat{\mathbf{S}}^{(i)}$ subsystems (the colored rectangular cubes), while EDA can cover the whole manifold spanned by all five $\hat{\mathbf{S}}^{(i)}$. To learn $\hat{\mathbf{S}}^{(i)}$, we use the stochastic subspace identification algorithm [31] and do not assume a specific form, unlike, e.g., auto-regressive moving average (ARMA) model used in [26] or the AR model used in [8] [35].

3 The Dynamic Bayesian Network Based on EDA

Based on EDA, we construct a DBN for tracking global and local hand motion. The tracking problem is formulated as inference both between consecutive frames and within each time frame given the observations.

Table 3. A list of the notations used in the DBN model

STATE VARIABLES	PHYSICAL MEANING
$\mathbf{M}[k] = [R[k]\|T[k]]$	3D motion w.r.t initial pose
$\mathbf{Z}[k]$	finger configuration
$V[k]$	feature observation
$\mathbf{X}[k]$	inner state of the 10^{th} order LDS
$\mathbf{X}^{(i)}[k]$	inner states of five *eigen dynamics*
$\mathbf{Y}[k]$	output of the LDS
$SH[k]$	skin histogram
$BH[k]$	background histogram

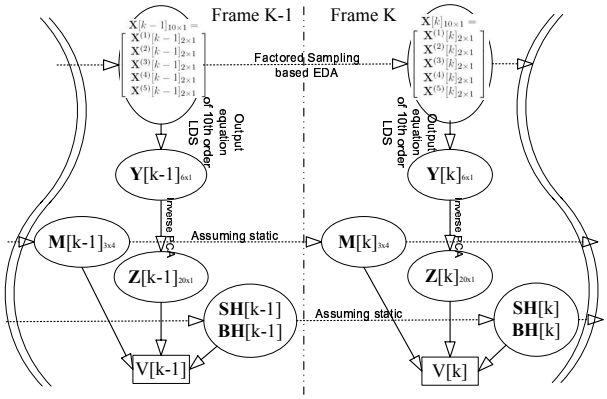

Figure 2. The dependency graph of the DBN for tracking articulated hand motion.

Figure 2 shows the dependency graph of the DBN depicting dependencies among the state variables and the ob-

servations at each frame. Table 3 lists the notations in the DBN.

Assuming the DBN is first-order Markovian, we can write the posterior of $\mathbf{M}[k]$ and $\mathbf{Z}[k]$ as

$$p(\mathbf{M}[k], \mathbf{Z}[k]|V[k], \lambda) = \frac{p(V[k]|\mathbf{M}[k], \mathbf{Z}[k], \lambda) \times p(\mathbf{M}[k], \mathbf{Z}[k]|\lambda)}{p(V[k]|\lambda)} \tag{4}$$

where the *a priori* knowledge λ includes: $SH[k-1]$, $\mathbf{M}[k-1]$ and $\mathbf{Z}[k-1]$. $p(V[k]|\lambda)$ is invariant with respect to $\mathbf{M}[k]$ and $\mathbf{Z}[k]$.

With this DBN, tracking hand motion can be cast as the maximum *a posteriori* (MAP) estimate of $\mathbf{M}[k]$ and $\mathbf{Z}[k]$ given the *a priori* and the observation $V[k]$. Since $\mathbf{M}[k]$ and $\mathbf{Z}[k]$ are conditionally independent given λ, we can use divide-and-conquer iteration to find the MAP estimate. Figure 3 shows the flow chart that implements the observation and inference within one frame of the DBN.

4 Observation: *Likelihood Edge*

Given an RGB image $F[k]$, we convert it to a gray-scale image $G[k]$ and an HSI(hue, saturation and intensity) image $H[k]$, which is further converted to a *likelihood ratio image* $L[k]$ based on $SH[k-1]$ and $BH[k-1]$ as:

$$L[k](u,v) = \frac{p(H[k](u,v)|skin)}{p(H[k](u,v)|nonskin)} \tag{5}$$

for each pixel (u, v). Most color segmentation algorithms [14] threshold the likelihood ratio to get a binary map of labels for different regions. In contrast, we propose not to threshold but to keep the quantitative information of the likelihood ratio and use it as sufficient statistics to generate a new feature called *likelihood edge*, that is, edge gradients on the likelihood ratio image $L[k]$, denoted by $LE[k]$[5]. In $G[k]$, we extract grayscale edge $GE[k]$. The edge points are candidates for matching with the sample points on the 2D shape model shown in Figure 5.

5 Inference: Divide and Conquer

As Figure 3 shows, the inference flow within one frame consists of two embedded loops: the outer (blue) loop is the divide-and-conquer iteration that solves $\mathbf{M}[k]$ and $\mathbf{Z}[k]$ by maximizing Equation (4) and the inner (green) loop is the ICP iteration for solving $\mathbf{M}[k]$. When divide-and-conquer iteration converges, we update $SH[k]$ and $BH[k]$ according to the current hand region.

[5]Since likelihood is sufficient statistics for classification [15]. The likelihood ratio edge gradient represents the normal direction of the boundary between skin and nonskin regions.

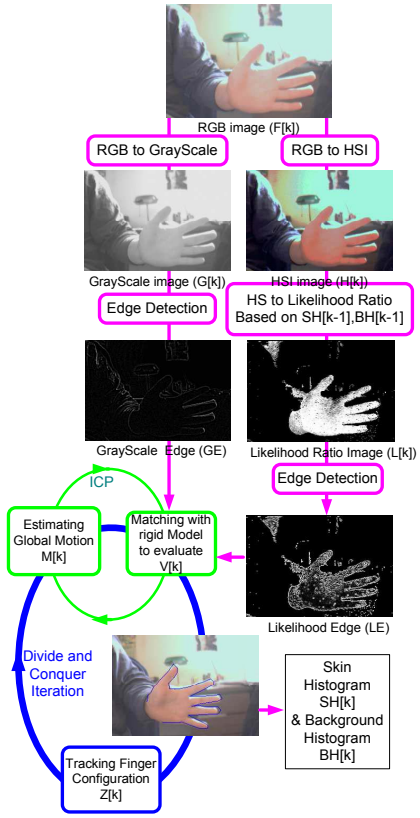

Figure 3. The flow chart for Bayesian hand tracking as inference within one frame of the DBN.

The bridge between step (2) and (3) is a 2D shape model shown in Figure 5. It is generated by rendering a 3D geometric hand model from current global pose and taking sample points along its silhouette. The points are given as $\{s_i = (m_i, dm_i), i = 1 \ldots M\}$, where $m_i = (u_i, v_i, 1)^T$ is the 2D homogeneous coordinate, $dm_i = (du_i, dv_i)^T$ is the normal direction (pointing from inner region to outer region) of the silhouette at m_i. Given the 2D shape model, there are many methods to solve for both correspondence and transformation [10]. Among them, the iterative closest point (ICP) algorithm[3] [36] is widely used because of its efficiency and guaranteed convergence to a local maximum. ICP iterates between assigning a binary correspondence based on nearest-neighbor relationship and estimating a transformation based on the correspondence. In our case, we search for the binary matching between the warped sample points and edge points as $F_E(i) = \arg\max_{(u,v)\in\wp}\{\psi(s_i, GE(u,v), LE(u,v))\}$, where $\wp$ denotes the neighborhood region. The similarity measure $\psi(s_i, GE(u,v), LE(u,v)) = dm_i^T GE(u,v) + dm_i^T LE(u,v)$ is the inner product between the normal di-

rection of the hand silhouette s_i and the edge gradient. When the ICP iteration converges, we use the four-point algorithm [7] to solve for the 3D homography H from $m_i[k] = Hm_i[1], i = 1 \dots M$ where $m_i[1]$ is all the sample points in the very first frame and $m_i[k]$ is those in the current frame. From H, we can uniquely decide [17] 3D rotation $R[k]$ and translation $T[k]$ in $\mathbf{M}[k] = [R[k]|T[k]]$.

6 Inference: Estimating Finger Configuration

Solving for MAP estimate $\mathbf{E}_{MAP}\{\mathbf{Z}[k]|V[k]\}$ (Step(3) in Figure 4) is accomplished by factored sampling in EDA space.

6.1 Sampling in EDA space

Since the mapping between $\mathbf{Z}[k]$ and the image feature $V[k]$ is nonlinear, the posterior probability density $p(\mathbf{Z}[k]|V[k], \mathbf{Z}[k-1])$ is multi-modal and cannot be approximated as normal. We therefore adopt a stochastic estimation technique based on factored sampling [30] to find the MAP estimate of $\mathbf{Z}[k]$.

When likelihood $p(V[k]|\mathbf{Z}[k])$ can be evaluated pointwise but it is infeasible to generate sample from, factored sampling can be used to approximate the MAP estimation. It draws N random samples $Z[k]_n$ $(n = 1 \dots N)$ from the prior $p(\mathbf{Z}[k]|\mathbf{Z}[k-1])$ and assigns to each sample a weight

$$\pi[k]_n = \frac{p(V[k]|\mathbf{Z}[k] = Z[k]_n)}{\sum_{m=1}^{N} p(V[k]|\mathbf{Z}[k] = Z[k]_m)} \tag{6}$$

It has been shown that $\lim_{N\to\infty} \sum_{n=1}^{N} Z[k]_n \pi[k]_n = \mathbf{E}_{MAP}\{\mathbf{Z}[k]|V[k]\}$ [30]. The speed of convergence dependents on how well the samples $Z[k]_n$ are generated with respect to the unknown posterior $p(\mathbf{Z}[k]|V[k])$. Since evaluating the likelihood is computationally expensive, it is preferable to draw samples from areas where the likelihood $p(V[k]|Z[k]_n)$ is very large, instead of adding up samples with negligible $\pi[k]_n$, which leads to importance sampling [2]. Given the true prior $f(Z[k]_n) = p(\mathbf{Z}[k] = Z[k]_n|\mathbf{Z}[k-1])$ and a function $g_k(\mathbf{Z}[k])$ which resembles the unknown posterior, we can draw samples $Z[k]_n$ from $g_k(\mathbf{Z}[k])$. In order to reflect the use of a different sampling distribution, we need to add a correction term in Equation (6) and get:

$$\pi[k]_n = \frac{p(V[k]|\mathbf{Z}[k] = Z[k]_n) f(Z[k]_n)/g(Z[k]_n)}{\sum_{m=1}^{N} p(V[k]|\mathbf{Z}[k] = Z[k]_m) f(Z[k]_m)/g(Z[k]_m))} \tag{7}$$

To draw a new sample at time k, we generate a random walk along the trajectory induced by the state transition equation of $\hat{\mathbf{S}}^{(i)}$:

$$X^{(i)}[k]_n = (A^{(i)})^{\lfloor w_n \rfloor} \mathbf{E}\{\mathbf{X}^{(i)}[k-1]\} \tag{8}$$

where $\mathbf{E}\{\mathbf{X}^{(i)}[k-1]\} = \sum_{m=1}^{N} \pi[k-1]_m X^{(i)}[k-1]_m$ and $w_n \sim \mathbf{N}(0, \sigma^{(i)})$. Since the observation noise u_n is Gaussian, we generate $Z[k]_n$ from $X^{(i)}[k]_n$ by

$$\begin{aligned} \widetilde{Z}[k]_n &= (U^T \textstyle\sum_{i\in I} C^{(i)} X^{(i)}[k]_n) + \mathbf{E}\{\mathbf{Z}\} \\ Z[k]_n &= \widetilde{Z}[k]_n + u_n \ \ where \ \ u_n \sim \mathbf{N}(0, \Sigma) \end{aligned} \tag{9}$$

Since the mapping $X^{(i)}[k]_n \rightarrow Y^{(i)}[k]_n \rightarrow Y[k]_n \rightarrow Z[k]_n$ is linear, the additive Gaussian noise in each stage can be transformed to equivalent Gaussian noise in $\Re^{20}$.Therefore, the importance function is:

$$g(Z[k]_n) = \prod_{i=1}^{5} p(X^{(i)}[k]_n|\mathbf{X}^{(i)}[k-1]) p(Z[k]_n|\widetilde{Z}[k]_n)$$

$$= \prod_{i=1}^{5} \frac{1}{\sigma^{(i)}} \exp\{-\frac{(w^{(i)}[k]_n)^2}{2{\sigma^{(i)}}^2}\} \frac{1}{|\Sigma|^{1/2}} \exp\{\frac{1}{2} u^T[k]_n \Sigma^{-1} u[k]_n\} \tag{10}$$

This sampling scheme based on EDA stratifies the sampling space into: 1) a manifold due to the uncertainty in the position of each of the five inner states along its own trajectories, which is parameterized as the random walk $w_n^{(i)}$ in

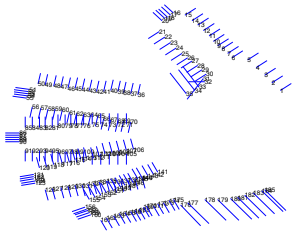

Figure 5. An example of the rigid planar model for global tracking. Each number stands for a sample point, while the blue line is its normal direction.

Step(1): Initialization
Use the EDA motion model to predict $\mathbf{M}[k]$ and $\mathbf{Z}[k]$ based on $\mathbf{M}[k-1]$ and $\mathbf{Z}[k-1]$.
Step(2): Solving for global motion
Recover $\mathbf{M}[k]$ with ICP, assuming the finger configuration $\mathbf{Z}[k]$ is fixed.
Step(3): Solving for finger configuration
Find MAP estimate of $\mathbf{Z}[k]$ with factored sampling, assuming $\mathbf{M}[k]$ is fixed.
Step(4): Testing convergence
If the likelihood is lower than a threshold, go back to Step (2); otherwise k := k + 1, process the next frame.

Figure 4. The divide-and-conquer iteration

the i^{th} subsystem; 2) the surrounding area due to observation noise in $\Re^{20}$, parameterized as a unimodal Gaussian $u_n \sim \mathbf{N}(0, \Sigma)$, with much smaller variance than that of directly sampling in $\Re^{20}$.

6.2 Evaluating the Likelihood Function

Given a sample $Z[k]_n$, we use a 3D geometric hand model to generate the silhouette from the current global view point. Taking sample points m_i along the silhouette, we find their corresponding edge point $F_E(i)$ at image position $e_{F_E(i)}$, by nearest neighbor search. Assuming the sum of its gray-scale edge strength and its likelihood edge strength is $\zeta(F_E(i))$, we define the likelihood function as

$$p(V[k]|\mathbf{Z}[k] = Z[k]_n) = \sum_{i=1}^{M} \exp\{-\frac{\| m_i - e_{F_E(i)} \|^2}{\zeta(F_E(i))}\} \tag{11}$$

which assumes the pixel coordinates of the edge points around sample point m_i have independent Gaussian distribution, with mean at m_i and the variance being the strength of the edge.

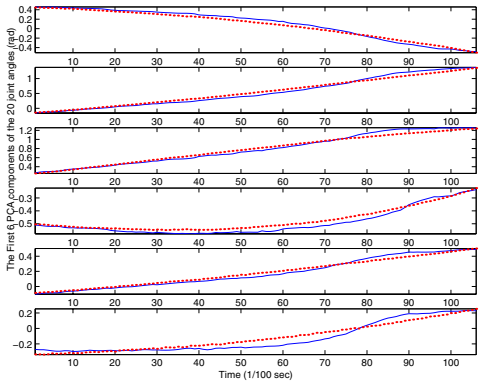

Figure 6. Trajectory of the 6 PCA components in flexing the ring finger. The doted red lines show the collected data, and the green lines show the recovered motion.

7 Experimental Results

7.1 Quantitative Results for Finger Articulation

In the simulation experiments, we use a 3D geometric hand model to render an image sequence from with the data collected by *CyberGlove*. To measure the performance of finger tracking without introducing the noise due to global tracker, we assume the ground truth global motion is known. We collect quantitative results of tracking five sequences of different types of motion (labelled as $D_i, i = 1 \ldots 5$, corresponding to flexing each of the five fingers). Figure 6 shows the results for the motion D_4 (flexing the ring finger). Table 4 shows relative mean square error (MSE) in the six dimensional PCA space Υ. Relative MSE is defined as the ratio between the absolute MSE and the range of motion (ROM). We choose not to enumerate the MSE in the configuration space Ç, because it is very lengthy and filled with a lot of insignificant entries corresponding to very small ROM. Since the mapping between Υ and Ç is linear, the relative MSE in Υ is sufficient for evaluating the performance of the finger tracking algorithm.

Table 4. The relative MSE in percentage for simulation using 2nd order LDS as the eigen-dynamics model

Motion \ PCA Dim	1	2	3	4	5	6
D_1 flexing thumb	5.1	4.4	4.5	3.5	3.6	9.4
D_2 flexing index	3.0	13.3	7.6	4.6	4.2	5.2
D_3 flexing middle	4.1	5.8	4.1	4.3	4.8	21.2
D_4 flexing ring	4.5	4.0	4.5	7.6	4.6	10.5
D_5 flexing pinky	9.7	3.7	6.4	3.1	11.4	8.4

Table 5. The ROM in each PCA component

	1	2	3	4	5	6
D_1	0.5783	0.5143	0.6809	1.4920	1.0173	0.2707
D_2	1.9592	0.2168	0.3278	0.4140	0.9302	0.2499
D_3	1.5827	0.4117	0.6401	1.1943	0.6834	0.0879
D_4	0.9446	1.5356	0.9936	0.3419	0.5903	0.5289
D_5	0.2459	1.7761	0.7135	0.8787	0.1192	0.3934

In Table 4, some entries are very large, e.g. the relative MSE reaches 21.2% in the 6^{th} PCA component of motion D_3 (extending/flexing middle finger). However, considering the ROM for the 6^{th} component is also quite small[6], only 0.0879 as shown in Table 5, the absolute MSE for that entry is reasonably small.

Table 5 shows the ROM in Υ. Table 6 shows the error while tracking finger motion with 1st order LDS as the model of the six eigen-dynamics, which is much larger than that of the 2nd order LDS'es. Using 1st order LDS is similar to the straight-line-fitting method used in [34]), except they assume finger motion is constrained along one of the lines between 28 predefined basis configurations at each time.

Table 6. The relative MSE (in percentage) for simulation using 1st order LDS as the eigen-dynamics model

	1	2	3	4	5	6
D_1	143.2	92.2	38.6	37.3	35.5	49.9
D_2	44.3	188.5	137.2	75.4	38.0	42.0
D_3	34.2	72.5	36.2	33.7	80.7	42.5
D_4	35.0	31.1	49.9	106.6	30.5	44.1
D_5	156.8	28.7	59.1	29.2	77.3	47.6

[6]The 6^{th} component corresponds to the smallest eigen value among those of the six components.

7.2 Quantitative Results for Global Motion

To analyze the performance of global tracking alone, we assume the ground truth finger configuration is known. The trajectory of translation and rotation parameters are shown in Figure 7. The mean square error (MSE) in each dimension is shown in Table 7, where $R(X), R(Y), R(Z)$ denotes rotation along X, Y, Z axis respectively and $T(X), T(Y), T(Z)$ denotes translation along X, Y, Z axis.

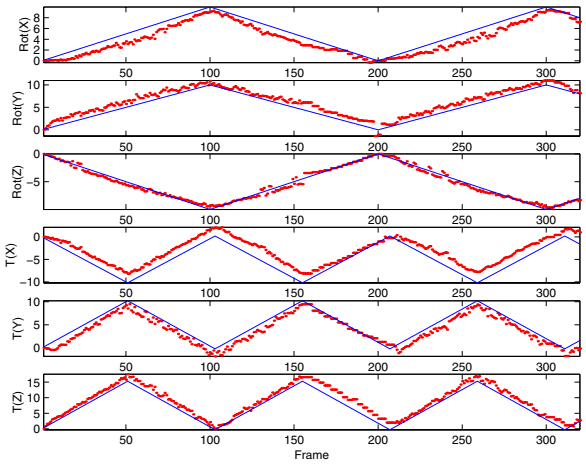

Figure 7. Trajectory of translation and rotation parameters. The green lines show the original transformation used to synthesis image sequence, the red dots show the global tracking results.

Table 7. MSE, ROM and relative MSE in percentage.

	$R(X)$	$R(Y)$	$R(Z)$	$T(X)$	$T(Y)$	$T(Z)$
MSE	1.22	1.13	0.40	1.78	1.11	1.06
ROM	10.4	10.4	15.6	10	10	10
RMSE(%)	11.73	10.87	2.56	17.75	11.06	10.60

The jittering effects occur when the nearest neighborhood search region covers the edges belonging to two adjacent fingers with very similar gradients. They can be alleviated by applying a smoothing filter when necessary.

7.3 Demonstration Using Real-world Data

The global tracker executes at 30 frames per second on an entry level processor (Pentium3 1.0GHz). Combining the local tracker, it slows down to 8 frames per second, partly because we have not optimize the implementation. Figure 8 and Figure 9 show some snapshots from various video clips. These and other video sequences are available at http://www.ifp.uiuc.edu/∼hzhou/EDA.

8 Conclusions

The main contributions of our work are:

1) We proposed an EDA method to learn a high order LDS depicting the dynamics of natural finger articulation. By imposing a structure tailored for hand motion, EDA separates the intrinsic biomechanical nonlinearity from the nonlinearity due to the asynchrony between different fingers, thus reduces the search space for tracking.

2) We proposed a new feature called *likelihood edge*, which combines color histogram and edge feature in the observation level.

3) Based on EDA, we implemented a system for tracking both articulation and 3D global hand motion. As the experiments on the synthesized and real-world data show, the system is accurate and robust against cluttered variant background and can handle partial occlusion.

Enumerated below are the limitations of the system:

1) Under certain finger articulations, self-occlusion will affect the evaluation of the likelihood function, such that the MAP estimation given by factored sampling may be far from the actual configuration. In that case, the divide-and-conquer iteration will take a long time to converge.

2) Under extreme out-plane rotation, the palm can not be approximated as a planar object. In such cases, the noise in point matching will introduce considerable error into the pose recovering results.

Figure 8. Demonstrations of the performance of the global hand tracker. The first two rows show the tracking results under inplane/outplane rotation and 3D translation. The third row shows that the tracker is robust against complex variant background. The fourth row demonstrates that starting from a rough initialization, the system can automatically converge to the actual hand pose. The fifth row shows the robustness against partial occlusion.

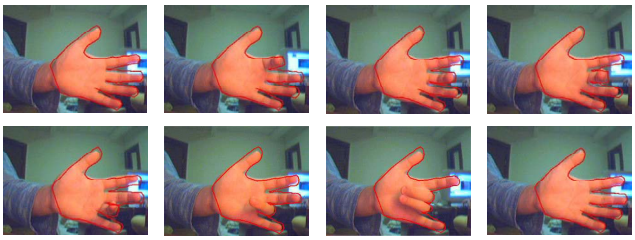

Figure 9. Demonstrations of the performance of the finger motion tracker.

Acknowledgements This research was supported in part by National Science Foundation, under Grant IIS 01-38965 and Alliance Program. The authors thank Ying Wu, John Lin, Dennis Lin and Bjorn Stenger for the inspiring discussions and selfless help. Special thanks for Ziyou Xiong's proofreading.

References

[1] Vassilis Athitsos and Stan Sclaroff. Estimating 3d hand pose from a cluttered image. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Madson, Wisconsin, June 2003.

[2] B.D.Ripley. *Stochastic Simulation*. Wiley, New York, 1987.

[3] P. J. Besl and N. D. Mckay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14:239–256, 1992.

[4] M. Black and A. Jepson. Eigentracking: Robust matching and tracking of articulated object using a view-based representation. In *Proc. European Conference on Computer Vision*, volume 1, pages 343–356, 1996.

[5] Chi-Tsong Chen. *Linear System Theory and Design*. Oxford University Press, New York, 1999.

[6] Quentin Delamarre and Olivier D. Faugeras. Finding pose of hand in video images: a stereo-based approach. In *Proc. IEEE International Conference on Automatic Face and Gesture Recognition*, pages 585–590, 1998.

[7] O. Faugeras, Q.-T. Luong, and T. Papadopoulo. *Geometry of multiple images*. MIT press, Cambridge, 2001.

[8] Andrew Fitzgibbon. Stochastic rigidity: Image registration for Nowhere-Static scenes. In *Proc. IEEE International Conference on Computer Vision*, volume 1, pages 662–669, 2001.

[9] Zoubin Ghahramani and Geoffrey E. Hinton. Switching state-space models. Technical report, Univ. of Toronto, 6 King's College Road, Toronto M5S 3H5, Canada, 1998.

[10] S. Gold, Anand Rangarajan, C. P. Lu, S. Pappu, and E. Mjolsness. New algorithms for 2-d and 3-d point matching: pose estimation and correspondence. *Pattern Recognition*, 31:1019–1031, 1998.

[11] T. Heap and D. Hogg. Towards 3D hand tracking using a deformable model. In *Proc. IEEE International Conference on Automatic Face and Gesture Recognition*, pages 140–145, Killington, VT, 1996.

[12] Tony Heap and David. Hogg. Wormholes in shape space: Tracking through discontinuous changes in shape. In *Proc. IEEE International Conference on Computer Vision*, pages 350–355, 1998.

[13] Michael Isard and Andrew Blake. CONDENSATION — conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29:5–28, 1998.

[14] M. Jones and J. Rehg. Statistical color models with application to skin detection. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume I, pages 274–280, Fort Collins, 1999.

[15] K.Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, New York, second edition, 1990.

[16] James J. Kuch and T. S. Huang. Vision-based hand modeling and tracking for virtual teleconferencing and telecollaboration. In *Proc. of IEEE International Conf. on Computer Vision*, pages 666–671, Cambridge, MA, June 1995.

[17] Yi Ma, Jana Kosecka, Stefano Soatto, and Shankar Sastry. *An Invitation to 3-D Vision*. Springer-Verlag, New York, 2002.

[18] J.P. MacCormick and M. Isard. Partitioned sampling, articulated objects, and interface-quality hand tracking. In *Proc. European Conference on Computer Vision*, 2000.

[19] Ben North, Andrew Blake, Michael Isard, and Jens Rittscher. Learning and classification of complex dynamics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(9):1016–1034, 2000.

[20] Vladimir Pavlovic, Brendan Frey, and Thomas S. Huang. Time-series classification using mixed-state dynamic bayesian networks. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 609–615, June 1999.

[21] Vladimir Pavlovic and James M. Rehg. Impact of dynamic model learning on classification of human motion. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 788–795, Hilton Head Island, SC., June 2000.

[22] Vladimir Pavlovic, James M. Rehg, Tat-Jen Cham, and Kevin P. Murphy. A dynamic bayesian network approach to figure tracking using learned dynamic models. In *Proc. IEEE International Conference on Computer Vision*, pages 94–101, September 1999.

[23] J. Rehg and T. Kanade. Model-based tracking of self-occluding articulated objects. In *Proc. of IEEE International Conf. Computer Vision*, pages 612–617, 1995.

[24] J.M. Rehg. *Visual Analyssis of High DOF Articulated Objects with Application to Hand Tracking*. PhD thesis, Electrical and Computer Eng., Carnegie Mellon University, 1995.

[25] Romer Rosales, Stan Sclaroff, and Vassilis Athitsos. 3D hand pose reconstruction using specialized mappings. In *Proc. IEEE International Conference on Computer Vision*, Vancouver, Canada, July 2001.

[26] Y. Wu S. Soatto, G. Doretto. Dynamic textures. In *Proc. IEEE International Conference on Computer Vision*, volume 2, pages 439–446, 2001.

[27] Nobutaka Shimada, Kousuke Kimura, and Yoshiaki Shirai. Real-time 3-d hand posture estimation based on 2-d appearance retrieval using monocular camera. In *Proc. of Intl. Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Realtime Systems*, pages 23–30, Vancouver, Canada, July 2001.

[28] B. Stenger, A. Thayananthan, P. H. S. Torr, and R. Cipolla. Filtering using a tree-based estimator. Nice, France, October 2003. to appear.

[29] A. Thayananthan, B. Stenger, P. H. S. Torr, and R. Cipolla. Shape context and chamfer matching in cluttered scenes. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume I, pages 127–133, Madison, USA, June 2003.

[30] Y. U.Grenander. *Introduction to Statistical Pattern Recognition*. Academic Press, New York, second edition, 1990.

[31] Peter van Overschee and Bart De Moor. *Subspace Identification for Linear Systems: Theory, Implementation Applications*. Kluwer Academic Publishers, New York, 1996.

[32] Y. Wu and T. S. Huang. Capturing articulated human hand motion: A divide-and-conquer approach. In *Proc. IEEE International Conference on Computer Vision*, pages 606–611, Corfu, Greece, Sept. 1999.

[33] Y. Wu and T. S. Huang. View-independent recognition of hand postures. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume II, pages 88–94, Hilton Head Island, South Carolina, June 2000.

[34] Y. Wu, John Lin, and T. S. Huang. Capturing natural hand articulation. In *Proc. of International Conference on Computer Vision*, pages 426–432, Vancouver, July 2001.

[35] H.-Y. Shum Y. Li, T. Wang. Motion texture: a two-level statistical model for character motion synthesis. In *siggraph2002*, pages 465–472, 2002.

[36] Zhengyou Zhang. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, 13:119–152, 1994.