# Automatic 2D Hand Tracking in Video Sequences

Quan Yuan Stan Sclaroff Vassilis Athitsos*
*Computer Science Department*
*Boston University*
*Boston, MA 02215*

## Abstract

*In gesture and sign language video sequences, hand motion tends to be rapid, and hands frequently appear in front of each other or in front of the face. Thus, hand location is often ambiguous, and naive color-based hand tracking is insufficient. To improve tracking accuracy, some methods employ a prediction-update framework, but such methods require careful initialization of model parameters, and tend to drift and lose track in extended sequences. In this paper, a temporal filtering framework for hand tracking is proposed that can initialize and reset itself without human intervention. In each frame, simple features like color and motion residue are exploited to identify multiple candidate hand locations. The temporal filter then uses the Viterbi algorithm to select among the candidates from frame to frame. The resulting tracking system can automatically identify video trajectories of unambiguous hand motion, and detect frames where tracking becomes ambiguous because of occlusions or overlaps. Experiments on video sequences of several hundred frames in duration demonstrate the system's ability to track hands robustly, to detect and handle tracking ambiguities, and to extract the trajectories of unambiguous hand motion.*

## 1. Introduction

Accurate detection and tracking of moving hands is a challenging problem, with applications in sign language recognition, human-computer interfaces and virtual reality environments. Magnetic trackers can capture hand motion accurately, but they are expensive and intrusive (users have to wear special equipment).

Vision systems are less intrusive to human users, but in monocular gesture and sign language video sequences the hand motion is highly non-rigid, and there are frequent occlusions and overlaps of hands. These conditions make hand tracking a challenging problem.

We propose a temporal filtering method for 2D hand tracking, i.e. for identifying the bounding box of the two hands in each frame of a video sequence. In each video frame a small number of candidate hand locations is detected using features based on color and motion residue. Then a temporal filter selects the most likely trajectory of hand locations. The observation probability is estimated using the average of posterior skin color probability over the candidate region. The transition probability is estimated using features based on hand location, hand velocity and normalized cross-correlation of two hand regions. We also propose a probabilistic method for identifying the beginning and end of trajectories where the hand location is unambiguous. Identifying such trajectories allows the tracker to stop and reset itself when there are occlusions, overlaps, or other events that make tracking ambiguous. We believe that identifying unambiguous trajectories and automatic resetting are very useful properties of the system. The tracker can track for long periods of time, and automatic resetting addresses the problems of drifting and losing track. Furthermore, video trajectories of unambiguous hand locations can be extracted and passed on for higher level processing tasks, like sign language recognition.

## 2. Related Work

Methods that employ the Prediction-Update framework are often applied to general tracking problems. The Kalman filter [1] and particle filters [18] are examples of such methods that have been employed to track moving hands. Isard and Blake [4] introduced a statistical factored sampling algorithm known as CONDENSATION to track hand contours in a cluttered background. This method was extended to track both hands by Mammen [14]. In that method, large perturbations in the measurement due to occlusion or clutter are modeled. However, these methods need manual initialization of model parameters before tracking, and they cannot tell when the tracker has lost track.

In Yang and Ahuja [8], hand gestures are tracked and recognized using a time-delay neural network, with features based on motion estimation of multiple regions. That approach did not address the cases when a substantial part of the hand is occluded or when two hands overlap. In Martin [9], 2D hand tracking is achieved by combining cues from

*This research was funded in part by NSF grants CNS-0202067, IIS-0208876, IIS-0308213, and IIS-0329009, and ONR N00014-03-1-0108.

skin detection and image differencing. These cues can be insufficient in some domains, for example if no color information is available, or when image differencing picks up motion from clothes, the face, or other objects.

Rasmussen and Hager [11] proposed a tracking method for multiple objects by using a joint PDAF, which extends the probabilistic data association filter(PDAF) [15] in Multiple Hypothesis Tracking. Their algorithm enforces a probabilistic exclusion principle to prevent two trackers from latching onto the same target. However, that system still needs initialization and an arbitrary minimum separation between targets.

A significant amount of work has focused on articulated hand tracking. The goal in such methods is to track in detail the motion of each finger and the palm, which is a much harder task than merely tracking the 2D location or bounding rectangles of hands. Rehg and Kanade [3] introduced the use of a highly articulated 3D hand model in their DigitEyes hand tracking system. Stenger [10] used a hand model of 39 quadrics and applied the unscented Kalman filter to track hand motion. Lu [12] used edge and optical flow to fit 3D models to 2D images and then track the hand motion. Sudderth [13] used nonparametric belief propagation and geometric hand model to track hand motion. Articulated models require accurate initialization. They are also sensitive to large occlusions of hands, which happen frequently in human gestures and sign languages.

## 3. Overview

Given a video sequence of a person performing a gesture, our goal is to find the bounding square of each hand at each frame. A good bounding square catches as many hand pixels as possible and as few background pixels as possible.

Given motion and color information about a single frame, our single-frame hand detector generates a small number of candidate hand locations, i.e., candidate bounding squares of hands. At most two of those locations are correct, since there are only two hands, and sometimes they overlap, so they are at the same location. However, motion and color information about a single frame are not sufficient to unambiguously identify the two hands. To account for that, the single-frame detector generates more than two candidate locations (five locations are generated in our experiments), in order to ensure that the true hand locations will almost always be among the candidates.

Given hand location candidates in multiple consecutive frames, we apply a temporal filtering method to refine the results of the single-frame detector. The hand location candidates that correspond to the same hand (say the right hand) in different frames are expected to be consistent with each other, in the sense that their position, velocity and appearance will not change much from frame to frame. We formulate a probabilistic criterion that identifies, among all possible trajectories of candidate locations, the top few trajectories that are the most likely to consist of candidates that correspond to a single hand.

Sometimes hand location becomes ambiguous, for example when hands overlap each other. The system identifies cases in which, for a trajectory of candidate hand locations, no candidate in the current frame is a good match. In such cases the tracker simply stops the trajectory, and tries to start a new trajectory. In this way, every trajectory output by the system consists of hand locations that are consistent with their previous and their next locations.

## 4. Detecting Candidate Hand Locations in a Single Frame

Most existing hand detection methods detect hands using some of the following assumptions: hands are skin-colored, the background is known and can be subtracted, and hands are moving faster than other objects. While these assumptions can be very useful in some domains, there also exist many cases where these assumptions are violated. An example is shown in Fig.1: The sequence is gray-scale, so no color information is available, background subtraction would also pick up the body and face of the subject, and motion detection also picks up the shirt, which is heavily textured. It is therefore beneficial to utilize additional features that can improve the hand detection in such cases.

In this paper we propose a novel feature, based on motion residue. Hands typically undergo non-rigid motion, because they are articulated objects. This means that hand appearance changes more frequently from frame to frame, compared to appearance of the clothes, the face, and background objects. We can use this property to detect hands, by identifying regions in each frame that have no good matches (in terms of appearance) among regions in the next frame.

For every two consecutive frames, the first frame is partitioned into blocks and then we try to find best match of each block in the next frame by translation. The block matching process is performed in a Gaussian pyramid which propagates the velocity of blocks from low-resolution levels to high-resolution levels. Using image pyramids, the matching process can be done efficiently.

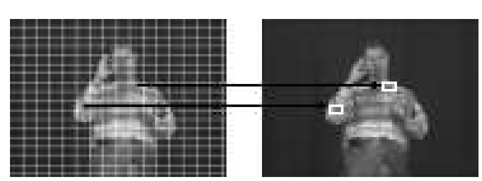

Figure 1: Find best match of each block in the next frame. The partitioned current frame is shown on the left, and the next frame is shown on the right.

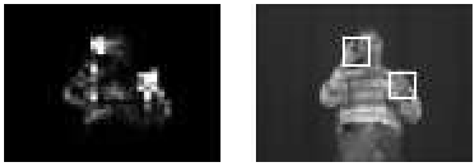

Figure 2: On the left is the motion residue of two frames in Figure 1. On the right is the located hands in current frame

Based on the best match of each block, we make an image of "block flow" to estimate the motion of regions. For every block we also estimate the residue, which is simply the average of absolute difference in intensity level between the block and its best match in the next frame. Because hands move nonrigidly in most cases, the blocks in a hand region tend to have high residues, and therefore we can use residue as a feature for detecting hands.

Hand candidates are identified as square areas with the largest residue value. In color sequences, we use skin detection to further improve detection accuracy. A skin color likelihood distribution and a non-skin color distribution are proposed in [7], in which the color space is in RGB but quantized to $32 \times 32 \times 32$ values. For each color, the probability of being skin is determined by the values for its skin likelihood and non-skin likelihood. We obtain a skin mask by thresholding the skin probability of each pixel. This mask is applied to the residue image, before searching for maxima in the residue image. In this way, no maxima will be found in image areas that are not skin-colored.

Each candidate hand location returned by the single-frame detector is a square bounding box centered at one of the local maxima in the residue image. The size of the bounding box is determined by the bounding box of the face, which is detected using a standard face detector [6].

# 5. Temporal Filtering

In each frame, we have hand candidates corresponding to real hand locations, and we also have false detections. Generally the real hand candidates are consistent from frame to frame, while the false detections lack such temporal consistency. Through temporal filtering, we can remove the false detections in each frame and extract the real hand trajectories.

First we introduce some notation:

$P(A)$: Probability of A, where A is an observation or a hypothesis. Similarly, $P(A|B)$ denotes the probability of $A$ given $B$, where B can similarly be an observation or a hypothesis.

$o_t$: Feature vector of a hand candidate at frame $t$. The feature vector contains information about appearance, skin likelihood, location, and velocity.

$H$: The hypothesis that a feature vector (or every vector of a trajectory of feature vectors) corresponds to a hand. For example, $P(H|o_1, \ldots, o_T)$ is the probability that every single $o_i$ in $\{o_1, \ldots, o_T\}$ is a hand.

$H_S$: The hypothesis that every vector of a trajectory of feature vectors corresponds to the same hand (i.e. the subject's left or right hand).

$s_t$: Part of $o_t$, that contains all the features that we need in order to estimate the probability that $o_t$ is a hand. Formally, $s_t$ is a subset of features from $o_t$ that satisfies that $P(H|s_t) = P(H|o_t)$.

$\Delta(o_t, o_{t-1})$: A feature vector extracted from two hand candidates $o_t$ and $o_{t-1}$, that captures all the information that we need to know to estimate the probability that $o_t$ and $o_{t-1}$ correspond to the same hand (given the knowledge that both $o_t$ and $o_{t-1}$ are actually hands). Formally, $\Delta(o_t, o_{t-1})$ is such that $P(H_S|\Delta(o_t, o_{t-1}), H) = P(H_S|o_t, o_{t-1}, H)$.

## 5.1. Optimization Criterion

Overall, we want our system to identify a trajectory $o_1, \ldots, o_T$ that maximize $P(H_S|o_1, \ldots, o_T)$. That is, we want to pick from each frame $t$ a candidate $o_t$, so that each $o_t$ is likely to be a hand, and each pair $o_{t-1}, o_t$ is likely to correspond to the same hand.

This subsection proves that, under some assumptions, the following equation holds:

$$P(H_S|o_1, \ldots, o_T) \propto \prod_{t=2}^{T} P(\Delta(o_t, o_{t-1})|H_S) \prod_{t=1}^{T} P(s_t|H) \quad (1)$$

This equation will allow us to construct optimal trajectories using the Viterbi algorithm [2]. The remainder of this subsection proves Eq. 1 and can be skipped if the reader is not interested in the mathematical details.

We assume that the probability that each of $o_1, \ldots, o_T$ is a hand is simply the product of the individual probabilities that each one of them is a hand. This assumption, together with the definition of $s_t$, can be summarized as

$$P(H|o_1, \ldots, o_T) = \prod_{t=1}^{T} P(H|o_t) = \prod_{t=1}^{T} P(H|s_t) \quad (2)$$

We also assume that, if all $o_1, \ldots, o_t$ are hands, the probability that they correspond to the same hand is simply the product of the probabilities of every pair $o_{t-1}, o_t$ corresponding to the same hand. This assumption, together with the definition of $\Delta(o_t, o_{t-1})$ can be summarized as:

$$P(H_S|o_1, \ldots, o_T, H) = \prod_{t=2}^{T} P(H_S|\Delta(o_t, o_{t-1}), H) \quad (3)$$

Overall, we want our system to construct a trajectory $o_1, \ldots, o_T$ that maximizes $P(H_S|o_1, \ldots, o_T)$. We can expand $P(H_S|o_1, \ldots, o_T)$ as follows:

$$\begin{aligned} & P(H_S|o_1, \ldots, o_T) \\ = \; & P(H_S, H|o_1, \ldots, o_T) \\ = \; & P(H_S|o_1, \ldots, o_T, H)P(H|o_1, \ldots, o_T) \end{aligned} \quad (4)$$

In the first step of Eq.4 we used the fact that the hypothesis $(H_S, H)$ is the same as the hypothesis $H_S$. In other words, if, for a trajectory of candidate hand locations, we know that all of them correspond to the same hand, then we also know that all those locations correspond to hands.

We have that

$$\begin{aligned} & P(H_S|o_1, \ldots, o_T, H) \\ = \; & P(H_S|\Delta(o_2, o_1), \ldots, \Delta(o_T, o_{T-1}), H) \\ = \; & \prod_{t=2}^{T} P(H_S|\Delta(o_t, o_{t-1}), H) \end{aligned} \quad (5)$$

Using Bayes rule, we get, for $t = 2, \ldots, T$ that

$$\begin{aligned} & P(H_S|\Delta(o_t, o_{t-1}), H) \\ = \; & \frac{P(\Delta(o_t, o_{t-1})|H_S)P(H_S|H)}{P(\Delta(o_t, o_{t-1}))} \end{aligned} \quad (6)$$

We assume that, as long as we do not know whether $o_t$ and $o_{t-1}$ are hands or not, $P(\Delta(o_t, o_{t-1}))$ can be approximated as a uniform distribution. This means that the denominator of Eq. 6 reduces to a constant. Along the same line, the $P(H_S|H)$ in the numerator is also a constant. Therefore, we get that

$$P(H_S|\Delta(o_t, o_{t-1}), H) \propto P(\Delta(o_t, o_{t-1})|H_S) \quad (7)$$

$$P(H_S|o_1, \ldots, o_T, H) \propto \prod_{t=2}^{T} P(\Delta(o_t, o_{t-1})|H_S) \quad (8)$$

Using similar manipulations, and by assuming that $P(s_t)$ is uniform, we get:

$$\begin{aligned} P(H|o_1, \ldots, o_T) \; &= \; P(H|s_1 \ldots, s_T) \\ = \prod_{t=1}^{T} P(H|s_t) \; &\propto \; \prod_{t=1}^{T} P(s_t|H) \end{aligned} \quad (9)$$

By combining all these results together, we get Eq. 1.

## 5.2. Estimation of Observation Probabilities

In order to maximize the right side of Eq. 1 we need to know $P(s_t|H)$ and $P(\Delta(o_t, o_{t-1})|H_S)$.

Feature $s_t$ is simply the mean, over all pixels in the candidate hand location $o_t$, of the probability of each pixel being skin. This probability is estimated as discussed in Sec. 4. We model $P(s_t|H)$ as a Gaussian distribution, with mean $u_h$ and variance $\sigma$:

$$P(s_t|H) \sim \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(s_t - u_h)^2}{2\sigma^2}} \; . \quad (10)$$

Quantities $u_h$ and $\sigma$ are estimated from a set of training samples.

Feature $\Delta(o_t, o_{t-1})$ is a five-dimensional vector describing how different $o_t$ and $o_{t-1}$ are in position, velocity and appearance. We define $(x_t, y_t)$ to be the center of candidate hand location $o_t$, and we define $(u_t, v_t)$ to be the observed velocity at that location. Velocity is estimated as a by-product of estimating the residual image (Sec. 4), where we find, for each block in one frame, its best matching block in the next frame. The velocity $(u_t, v_t)$ of $o_t$ is simply the average of the velocities of all the blocks that are inside the bounding square that specifies the location of $o_t$. We define the four-dimensional vector $\Delta'(o_t, o_{t-1})$ to be the vector $(x_t - x_{t-1}, y_t - y_{t-1}, u_t - u_{t-1}, v_t - v_{t-1})$.

Since we are not focusing on specific gestures, we assume that the change of position and velocity from $o_{t-1}$ to $o_t$ has a multi-dimensional Gaussian distribution, when $o_t$ and $o_{t-1}$ correspond indeed to the same hand:

$$P(\Delta'(o_t, o_{t-1})|H_S) \sim \mathcal{N}(\bar{0}; \Sigma) \; . \quad (11)$$

The covariance matrix $\Sigma$ is trained by sample trajectories.

To measure how similar the appearance of $o_t$ and $o_{t-1}$ is, we take the maximum normalized cross-correlation coefficient $\mathrm{Corr}(o_t, o_{t-1})$ of the two image windows corresponding to $o_t$ and $o_{t-1}$. We make a histogram $\mathrm{Hist}(\mathrm{Corr}(o_t, o_{t-1}))$ using training trajectories in which we manually identify, in consecutive frames, pairs of windows in consecutive frames that correspond to the same hand. We normalize the histogram so that the sum of its entries is 1, so that the histogram describes the probability of getting a correlation value given that $o_t$ and $o_{t-1}$ are indeed hands and correspond to the same hand:

$$\mathrm{Hist}(\mathrm{Corr}(o_t, o_{t-1})) \simeq P(\mathrm{Corr}(o_t, o_{t-1})|H_S) \; . \quad (12)$$

Now we can finally define feature $\Delta(o_t, o_{t-1})$ as $(\mathrm{Corr}(o_t, o_{t-1}), \Delta'(o_t, o_{t-1}))$. Assuming that the probability of the appearance-based correlation is independent of the probability of the combined position and velocity changes, we have:

$$\begin{aligned} & P(\Delta(o_t, o_{t-1})|H_S) \\ = \; & P(\Delta'(o_t, o_{t-1})|H_S)P(\mathrm{Corr}(o_t, o_{t-1})|H_S) \end{aligned} \quad (13)$$

## 5.3. Measuring Consistency Between Consecutive Frames

In this subsection we establish a criterion for telling when two consecutive observations $o_{t-1}, o_t$ are likely to belong to the same hand. This criterion is useful for preventing the system from forming trajectories in which two consecutive locations are deemed to be inconsistent with each other. If, at previous frame $t-1$, the observation $o_{t-1}$ found by the Viterbi algorithm to have a link to the observation $o_t$ is inconsistent with $o_t$ , we don't add observation $o_t$ to that trajectory ending at $o_{t-1}$. If $o_{t-1}$ can't have any link to observations in frame $t$, the trajectory ending at $o_{t-1}$ stops.

We define this consistency criterion $P_S(H_S|o_t, o_{t-1})$ to be the probability, given that $o_{t-1}$ is a hand, that $o_t$ and $o_{t-1}$ correspond to the same hand. When $P_S$ is greater than 0.5, we consider observations $o_t, o_{t-1}$ to be consistent with each other. To derive what $P_S$ is equal to, first we observe that, based on Eq. 4,

$$P(H_S|o_t, o_{t-1}) = P(H_S|\Delta(o_t, o_{t-1}), H)P(H|o_t, o_{t-1}) \tag{14}$$

To save space, we define $k_1 = P(H_S|H)$, and $\Delta_t = \Delta(o_t, o_{t-1})$. We get that

$$\begin{aligned} & P_S(H_S|o_t, o_{t-1}) = \frac{P(H_S|o_t, o_{t-1})}{P(H|o_{t-1})} \\ = \; & P(H_S|\Delta_t, H)P(H|o_t) \\ = \; & \frac{P(\Delta_t|H_S)k_1}{k_1 P(\Delta_t|H_S) + (1-k_1)P(\Delta_t|\overline{H_S}, H)} \\ \times \; & \frac{P(o_t|H)P(H)}{P(H)P(o_t|H) + P(\overline{H})P(o_t|\overline{H})} \end{aligned} \tag{15}$$

Here we assume that $P(\Delta(o_t, o_{t-1})|\overline{H_S}, H)$ and $P(o_t|\overline{H})$ are uniformly distributed. Quantities $P(H)$, $P(\overline{H})$, $P(H_S|H)$, $P(\overline{H_S}|H)$ are constants. To simplify notation, we introduce the following new constants:

$k_2 = P(H)$,
$c_1 = P(\overline{H_S}|H)P(\Delta(o_t, o_{t-1})|\overline{H_S}, H)$,
$c_2 = P(\overline{H})P(o_t|\overline{H})$

So if $P_S(H_S|o_t, o_{t-1}) > 0.5$ then we have

$$\begin{aligned} & \frac{k_1 k_2 P(\Delta_t|H_S)P(o_t|H)}{(k_1 P(\Delta_t|H_S) + c_1)(k_2 P(o_t|H) + c_2)} > 0.5 \\ \Rightarrow \; & 2k_1 k_2 P(\Delta_t|H_S)P(o_t|H) > \\ & k_1 k_2 P(\Delta_t|H_S)P(o_t|H) + k_1 c_2 P(\Delta_t|H_S) + \\ & k_2 c_1 P(o_t|H) + c_1 c_2 \\ \Rightarrow \; & k_1 k_2 P(\Delta_t|H_S)P(o_t|H) - k_1 c_2 P(\Delta_t|H_S) - \\ & k_2 c_1 P(o_t|H) - c_1 c_2 > 0 \end{aligned} \tag{16}$$

Then, to tell whether $P_S(H_S|o_t, o_{t-1}) > 0.5$, we can build a linear discriminant $(a, b, c, d)$ such that

$aP(\Delta_t|H_S)P(o_t|H) + bP(\Delta_t|H_S) + cP(o_t|H) + d > 0$ when $o_t$ is the same hand as $o_{t-1}$ and

$aP(\Delta_t|H_S)P(o_t|H) + bP(\Delta_t|H_S) + cP(o_t|H) + d < 0$ when $o_t$ is not a hand or is not the same hand as $o_{t-1}$.

Finding $a, b, c, d$ is a standard problem of finding a linear discriminant for two classes. To solve it, we use the standard Minimum Squared-Error technique of the pseudo-inverse matrix [17]. We use some training trajectories to obtain positive and negative examples for the training.

## 5.4. Dynamic Programming Implementation

In Sec. 5.1, we use the Viterbi algorithm to extract the most likely hand trajectories. In a Viterbi net a node $N$ at current time $t$ corresponds to a candidate hand feature vector in frame $t$. Each node points to a node corresponding to time $t-1$, and by tracing these pointers starting at node $N$ we can recover the current best trajectory ending at $N$.

The complete procedure is given in Algorithm 1. To simplify notation we use $L(s_t|H)$ and $L_H(o_t, o_{t-1})$ to denote the logarithm of $P(s_t|H)$ and $P(\Delta(o_t, o_{t-1})|H_S)$ respectively.

**input** : A sequence of $T$ frames, with $n$ feature vectors $x_t^1 \dots x_t^n$ at each node $t$
**output** : a group of trajectories seq(1),seq(2),...
//Definitions
$\delta_t^i$: current score of candidate $o_t^i$ in frame $t$.
$\psi_t(i)$: index of candidates at frame $t-1$, which links to candidate $x_t^i$ at frame $t$.
$n$: the number of candidates in each frame.
$m$: the number of candidates we will extract in each frame .
**while** $\text{total number of extracted candidates} < mT$ **do**
  Find first node $\tau$with more than $n-m$ feature vectors.
  //Initialization
  **for** $i = 1 : \text{number of feature vectors at node } \tau$ **do**
    $\delta_\tau^i = L(x_\tau^i|H)$;
    $\psi_\tau(i) = 0$;
  **end**
  //Recursion
  $t = \tau + 1$
  **while** *(Not all $\delta_{t-1}^s = -\infty$) $\wedge$ ($t \leq T$)* **do**
    **for** $j = 1 : \text{number of feature vectors at } t$ **do**
      $\delta_t^j = \max_{1 \leq i \leq n}[\delta_{t-1}^i + L_H(o_t^j, o_{t-1}^i) + L(o_t^j|H)]$
      $\psi_t(j) = \underset{1 \leq i \leq n}{\operatorname{argmax}}[\delta_{t-1}^i + L_H(o_t^j, o_{t-1}^i) + L(o_t^j|H)]$
      **if** *($o_t^j$ and $o_{t-1}^{\psi_t(j)}$) does not satisfy Eq. 16* **then**
        $\delta_t^j = -\infty$
      **end**
    **end**
    $t = t + 1$
  **end**
  Output the trajectory of feature vectors from $\tau$ to $t-1$ by tracking back $\psi_t(j_m)$, where$j_m = \arg\max_{1 \leq j \leq n} \delta_{t-1}^j$
**end**

Algorithm 1: Modified Viterbi algorithm used in our system.

To decide whether the current best trajectory should end at frame $t$, we only need to check whether the current best trajectory can find a candidate hand square in frame $t+1$ that satisfies Eq. 16. So, different from the conventional Viterbi algorithm, we have an extra check at each time $t$. When all the trajectories cannot go beyond frame $t$, the best trajectory at frame $t$ is output. Then the algorithm finds the first frame with a sufficient number of candidates as a new start.

The time complexity of this modified Viterbi algorithm is O($mn^2T$), where m is the number of targets and n is the number of candidates in each frame.

In Algorithm 1, the parameter $m$ is the number of objects we expect to track in a video sequence. Because human faces have similar color and size as hands, we set $m$ = 3 in our hand tracking system. After the Viterbi algorithm we then remove face trajectories by a post-processing step: we compare the extracted trajectories with locations of detected faces in all the frames. If three of the hand locations in a trajectory overlap with face regions, this trajectory is regarded as a face trajectory and removed. Trajectories shorter than three frames are also discarded. After post-processing it is possible that in some frames we have 3 trajectories, but this error occurred rarely in our experiments.

# 6. Experiments

The method is tested on sign language sequences. We evaluate the single-frame detector on gray-scale video sequences, on which skin-color detection is not applicable. We also evaluate the overall algorithm on color video sequences and compare with CONDENSATION[4]. All video sequences display subjects using sign language. None of the test sequences was used for training the system.

## 6.1. Single-Frame Detection Results

We tested the single-frame detector on a number of American sign language videos which have 1444 frames in total. These videos are 8-bit gray-level videos, so skin color detection cannot be applied to them. The ground-truth of hand locations is manually recorded. The size of blocks is 8 by 8 when the translation residue is computed. We use 3-level Gaussian pyramid which propagates the velocity of blocks from higher level to lower level. At each frame we pick up three maxima in the motion residue image as hand candidates and then compare them with the ground-truth. To measure accuracy we compute the distance between centers of hand candidates $(x_c, y_c)$ and centers of hand locations based on ground-truth $(x_g, y_g)$. Let $d$ be the distance between $(x_c, y_c)$ and $(x_g, y_g)$. When $d$ is smaller than the minimum radius of the bounding circles of the ground-truth and the candidate, then we consider the candidate "well-located". In these 1444 frames we get 1.8061 "well-located" candidates per frame. Naturally, a perfect result would be 2.0 "well-located" candidates per frame.

## 6.2. Temporal Filtering Results

We extracted four video sequences of 200 frames each. Two video sequences depict American Sign Language [19]. The other two depict Flemish sign language [20]. For ground truth we manually marked the hand locations in each frame by squares. When the center of an extracted hand is inside the hand square according to ground-truth, then the ground-truth hand is regarded as correctly detected (denoted as "detected hands" in Table 1).

Ideally the system would extract trajectories composed completely of locations of the same hand. However, an extracted trajectory sometimes includes locations of the other hand, or even non-hand locations. As a measure of system performance, we estimate for each extracted trajectory the percentage of locations that correspond to the same hand. We call this measure "consistency." For example, if a trajectory of 30 locations is composed of only left hands, the consistency score is 100%. If the trajectory contains 24 right hands, 4 left hands and 2 non-hand locations, the consistency score is 80% (i.e, 24/30). The final consistency score for a video is the weighted sum of the consistency scores of all extracted trajectories, where each trajectory is weighted by its length.

Table 1: *Test results on sign language data, the perfect value of Detected hands and Consistency are 100%.*

| % Detected hands | A | B | C | D |
|---|---|---|---|---|
| our method | 90% | 66.5% | 75.8% | 81.8% |
| CONDENSATION | 66% | 14.5% | 73% | 38.3% |
| **% Consistency** | A | B | C | D |
| Our method | 92.4% | 70.2% | 90.1% | 83% |
| CONDENSATION | 78.1% | 11.6% | 72.6% | 40% |

A: "DSP Introduction to a Story" in [19], frame 1 to frame 200.
B: "DSP Ski Trip Story" in [19], frame 4001 to frame 4200.
C: "NGT_AH_fab2_b.mpg" in [20], frame 1 to frame 200.
D: "NGT_AW_fab5_b.mpg" in [20], frame 501 to frame 700.

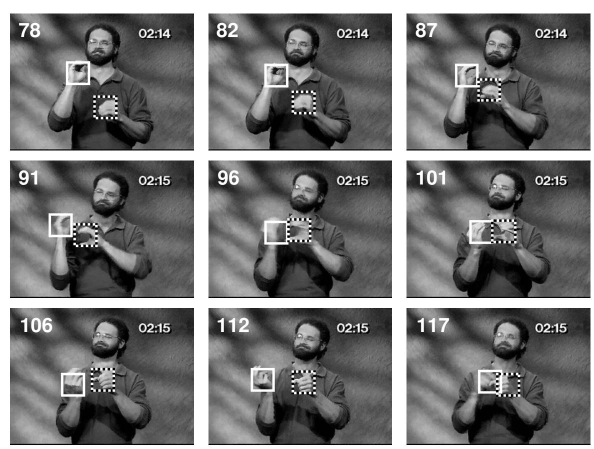


Figure 3: *Tracked hand locations (2 hand trajectories combined ) from the "Ski Trip Story" in ASLLRP SignStream Databases [19]. At the top-left corner is the frame number.*


Figure 4: *Tracked hand locations in the Flemish Sign Language [20] video "NGT_AH_fab2_b.mpg" .*

We compare our method with the CONDENSATION[4] algorithm. To apply CONDENSATION, we manually initialize the hand locations in the first frame and we keep 100 samples in each of the following frames. The prediction is based on the same Gaussian model of 2-D location as in our method, Eq.10. The update is based on skin probability, normalized correlation score and blockflow of the new sample, using the same model and parameter values as in our method. For the four sequences tested it was observed that CONDENSATION tends to drift when there are occlusions or hands are moving fast. When CONDENSATION loses track, it has no mechanism to reset itself. Table 1 compares the results of our algorithm versus CONDENSATION.

# 7. Conclusion

We have described a 2D hand tracking method that extracts trajectories of unambiguous hand locations. Candidate hand bounding squares are detected using a novel feature, based on motion residue. This feature is combined with skin detection in color video. A temporal filter employs the Viterbi algorithm to identify consistent hand trajectories. An additional consistency check is added to the Viterbi algorithm, to increase the likelihood that each extracted trajectory will contain hand locations corresponding to the same hand. The consistency check allows the system to stop tracking when there are ambiguous observations. The tracker can reset itself automatically after stopping the previous trajectory. Results on video sequences of sign languages demonstrate the robustness of our method.

# References

[1] R. E. Kalman,"A New Approach to Linear Filtering and Prediction Problems," *Trans. of the ASME–J. of Basic Engineering*, Vol.82, Series D, pp.35-45, 1960.

[2] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimal decoding algorithm, " *IEEE T. Info. Theory*, Vol. IT-13:260-269, 1967.

[3] J. M. Rehg and T. Kanade, "Visual tracking of high DOF articulated structures: an application to human hand tracking," *Proc. ECCV*, Vol.2, pp.35-46, 1994.

[4] M. Isard and A. Blake, "Condensation – conditional density propagation for visual tracking," *IJCV*, 29(1):5-28, 1998.

[5] M.J. Black and A.D. Jepson, "EigenTracking: Robust Matching and Tracking of Articulated Objects Using a View-Based Representation," *IJCV* , 26(1):63-84, 1998.

[6] H.A.Rowley, S.Baluja, and T.Kanade, "Neural Network-Based Face Detection, "*IEEE T-PAMI*, 20(1):23-38, 1998.

[7] M.J. Jones and J.M. Rehg, "Statistical color models with application to skin detection, " *CVPR*, I:1466-1472,1999.

[8] M.H. Yang and N. Ahuja, "Recognizing Hand Gesture Using Motion Trajectories," *Proc. CVPR*, I:1466-1472,1999.

[9] J. Martin, V. Devin and J.L. Crowley, "Active Hand Tracking," *Proc. IEEE Conf. on Face and Gesture*, pp.573-578, 1998.

[10] B. Stenger, P. Mendonca and R. Cipolla, "Model-Based 3D Tracking of an Articulated Hand," *Proc. CVPR*, II:310-315, 2001.

[11] C. Rasmussen and G. Hager, "Probabilistic data association methods for tracking complex visual objects," *IEEE T-PAMI*, 23(6):560-576, 2001.

[12] S. Lu, D. Metaxas, D. Samaras and J. Oliensis, "Using Multiple Cues for Hand Tracking and Model Refinement," *Proc. CVPR*, II:443-450, 2003.

[13] E.B. Sudderth, M.I. Mandel, W.T. Freeman, and A.S. Willsky, "Visual Hand Tracking Using Nonparametric Belief Propagation," *IEEE Workshop on Generative Model Based Vision*, 2004.

[14] J.P. Mammen, S. Chaudhuri and T. Agrawal, "Simultaneous Tracking of Both Hands by Estimation of Erroneous Observations, " *Proc. BMVC*, 2004.

[15] Y.Bar-Shalom and T.Fortmann, *Tracking and Data Association,* Academic Press, 1988.

[16] S.Blackman and Robert Popoli, *Design and Analysis of Modern Tracking Systems,* Artech House, 1999.

[17] Richard O. Duda, Peter E. Hart and David G. Stork, *Pattern Classification,* Second ed., Wiley-Interscience, 2000.

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

[19] http://www.bu.edu/asllrp/cd/cont-s3.html.

[20] O. Crasborn, E. van der Kooij, A. Nonhebel and W. Emmerik. " ECHO data set for Sign Language of the Netherlands (NGT), " Dept. of Linguistics, Univ. of Nijmegen, Netherlands, 2004. http://www.let.kun.nl/sign-lang/echo.