Fast Viewpoint-Invariant Articulated Hand Detection Combining Curve and Graph Matching

Luigi Cinque, Marco Cupelli and Enver Sangineto
Computer Science Department, University of Rome “Sapienza”,
via Salaria 113, 00198, Rome, Italy
{cinque, cupelli, sangineto}@di.uniroma1.it

Abstract
We present an approach to viewpoint invariant hand detection which merges model based representation of shape and exact curve matching with graph search in order to achieve a very low false alarm rate system able to work in real time. The method proposed makes few assumptions on the articulated object nature and can be applied to recognize other articulated objects as well.

1. Introduction
Detection and tracking of articulated objects such as human body and its limbs is a challenging task in computer vision with important applications such as human-computer interfaces, domotic systems, video-surveillance or sign language interpretation. Significant progresses have been done in this century in generic object detection using machine learning techniques (e.g., [11]). Nevertheless, successful object detectors usually assume to work with rigid objects. Difficulties with articulated objects arise from the fact that the large variability of appearances of a generic object class becomes larger in dealing with articulated object classes due to the variable positions of each object’s sub-part w.r.t. the others. Solutions proposed so far to this problem usually impose fixed configurations to the rigid components of the object to recognize (e.g., in the case of hand, fixed positions of each finger w.r.t. the others) or constraints to the application domain, such as to the image/video background nature. As a matter of fact, recognizing articulated objects with full degrees of freedom (DOF) in cluttered background images is still an open problem.

In [6], Kolsh and Turk use the Viola-Jones object detector [11] in order to detect hands in fixed postures. They select a set of postures (e.g., open hand, closed hand, fist, etc.) and train a different classifier for each posture. The system is able to recognize hands in images with a difficult cluttered background but the hand is constrained to have a fixed position w.r.t. the rest of the image and a fixed posture: fingers cannot move the one w.r.t. the others because they must have a configuration similar to the one used at training time. Cipolla and colleagues [10] use a set of different thousands of shape models of a given hand posture, organized in a three level tree, exhaustively matching these models to the binary edge image using the chamfer distance. Each possible discrete value of the hand’s position parameters is represented by a specific 2D hand model, synthesized offline using a 3D model. Despite the hierarchical structure in which the models are clustered, the number of model-image comparisons required is quite huge, leading to about a 2-3 seconds execution time per image on a standard PC [10]. Moreover, like in the work of Kolsh and Turk, the finger configuration is fixed.

The possibility to separately search for each object sub-part in the image and then composing the rigid detection results has been explored by Huttenlocher and colleagues [4]. In [4] the authors describe an articulated object (e.g., the human body) by means of rectangular patches organized in a tree-structure. They extend distance transform techniques in order to deal with “template and spring” [5] metrics and reduce the object rigid components’ assembling time. Nevertheless, the proposed approach is still slow, requiring about 10 seconds on a standard PC to recognize an 11-component object in a very low-resolution image. Moreover, spatial relations described by a “template and spring” metric assume a “preferred” configuration of the object’s components w.r.t. the other possible configurations, which is not always realistic. Sigal et al. [9] represent the conditional probability of the position of a given human body limb w.r.t. its neighboring limb’s position using a more flexible, non-parametric formulation. The proposed system relies on a multiple view of the scene given by 4 calibrated cameras to perform limb detection.

In this paper we present an approach to articulated object detection applied to the problem of hand detection which is based on two phases. In the first phase we separately search for the object rigid sub-parts (i.e., the fingers in our case)
using a curve matching technique. Curve matching is preferred to chamfer distance because it is more accurate, as shown in Section 3. However, point-by-point comparison needs to deal with only connected lines, while edge detection is an unstable process usually producing noisy images with missing components (e.g., see Figure 2 (a)). In order to cope with these problems, we pre-organize edge information using a graph structure. In the second phase, the relative positions of the detected fingers is analyzed in order to validate the correct detection of a hand.

The system is able to detect a hand in cluttered images, with different relative positions of a finger w.r.t. the others and with variable positions of the whole hand w.r.t. the scene. Minor out-of-plane rotations of the hand are tolerated as well.

2. Pre-Processing

Hand movement in human gestures is difficult to predict and common tracking algorithms based on a known motion model (e.g., the Kalman filter) usually show very low performances in this domain [10]. For this reason, like in [10] we perform hand detection on every frame of the video sequence. To speed-up detection, we first apply to each frame a set of pre-processing operations aiming at selecting a region of interest (ROI) in which the subsequent detection process can focus the attention. However, some of the pre-processing operations are not strictly necessary since the system is able to detect a hand in still images or just after a sudden scene change due to a camera movement.

Essentially, we perform combined motion and skin detection in order to select a ROI into the current frame. Motion detection is realized by means of standard single frame difference and background difference. To speed-up, we first apply to each frame a set of pre-processing operations aiming at selecting a region of interest (ROI) in which the subsequent detection process can focus the attention. However, some of the pre-processing operations are not strictly necessary since the system is able to detect a hand in still images or just after a sudden scene change due to a camera movement.

Essentially, we perform combined motion and skin detection in order to select a ROI into the current frame. Motion detection is realized by means of standard single frame difference and background difference. To deal with the foreground aperture problem of the single frame difference phase, we select rectangular ROIs around connected blobs of foreground pixels and we check each rectangle using background difference w.r.t. a background model dynamically updated. The advantage w.r.t. performing only background difference techniques is evident in situations in which the camera has just been moving for a while. In fact, in this case, background difference alone will signal all the frame pixels as different w.r.t. the old background model while single frame difference allows to select a subset of the image which is currently moving.

On the foreground pixel resulting from the motion analysis above mentioned, we perform a rule-based skin detection on an R-G projection of the RGB space (see [7] for more details). Let us call $R_t$ a binary mask representing the ROI of frame $t$, defined as:

$$R_t(p) = \begin{cases} 
1 & \text{if } p \text{ is a skin-colored foreground pixel} \\
0 & \text{otherwise}
\end{cases}$$  \hspace{1cm} (1)

Finally, on the ROI so selected, we perform the Canny edge detection [2] and we prune small edge lines. Let us call $D_t$ the pruned (binary) edge map of $I_t$. From now on we will omit the subscript $t$ when it is clear the reference to the current frame $t$. Figures 1 and 2 (a) show, respectively, the ROI and the edge map of a same input image.

3. The finger model

As a first step in our proposed hand detection approach, we search for a possible match between the image and an affine-transformed representation of the finger’s appearance and shape model. The finger’s appearance model $M^{(A)}$ is a simple positional and color model but it could be extended to include texture information (e.g., in dealing with detection of human dressed figures). $M^{(A)}$ is given by a set of points describing the inner part of a finger (see Figure 2 (b)), represented in a 2D reference frame and projected into the image to verify the presence of skin and motion pixels and the absence of noisy edge pixels.

More formally, let $T$ be a rigid affine transformation from the model reference frame to the image reference frame and $\{M_s^{(A)}\}_{s=1}^{\infty}$ a set of appearance models offline computed for different discrete scale value $s$. In turn
is a set of points: \( M_s^{(A)} = \{ r_1, \ldots, r_h \} \). If we assume that the movements of the whole hand and of the single fingers happen in a plane parallel to the camera plane, we can restrict our attention to similarity transformations. Hence, \( T \) can be defined by a 4 parameters vector \( v = (t_x, t_y, \theta, s) \), being \((t_x, t_y)\) the translation offset, \( \theta \) the rotation angle and \( s \) the scaling factor.

With these assumptions, the appearance error (\( E^{(A)} \)) obtained in projecting the finger model \( M_s^{(A)} \) into the image using transformation \( T_v \) (being \( v = (t_x, t_y, \theta, s) \)) is given by:

\[
E^{(A)} = E^{(R)} + \lambda E^{(D)},
\]

where \( E^{(R)} \) is the error computed on the ROI mask, \( E^{(D)} \) is the error on the edge image and \( \lambda \) a weighting factor:

\[
E^{(R)} = \sum_{i=1}^{h_s} [1 - R(T_v(r_i))] = h_s - \sum_{i=1}^{h_s} R(T_v(r_i))
\]

\[
E^{(D)} = \sum_{i=1}^{h_s} D(T_v(r_i))
\]

Eq. (3) says that the error in projecting \( M^{(A)} \) into the ROI of the current frame is given by the number of pixels for which \( R(T_v(r_i)) = 0 \). Conversely, in Eq. (4) the error is proportional to the number of noisy edge pixels possibly present in the inner part of the finger (\( D(T_v(r_i)) = 1 \)). While in principle we could have only one appearance model \( M^{(A)} \), we prefer to compute off-line different versions \( M_{s_1}^{(A)}, M_{s_2}^{(A)}, \ldots. \) each having its own cardinality \( (h_{s_1}, h_{s_2}, \ldots) \) in order to avoid computational expensive runtime scaling operations.

The shape model \( M^{(S)} = \{ q_1, \ldots, q_m \} \) is a set of pairwise adjacent points forming an open curve which represents the finger silhouette (see Figure 2 (b)). During the finger detection process we search for a rigid transformation \( T_v \) and a curve \( C \) in \( D \) such that \( T_v(M^{(S)}) \) and \( C \) are perceptually similar. If \( m \) is the cardinality of \( M^{(S)} \), then similarity between \( C \) and \( T_v(M^{(S)}) \) can be checked by uniformly subsampling \( C \) in order to extract from it the same number \( m \) of points of \( M^{(S)} \) and then comparing corresponding points in the two resulting curves. If \( C' = \{ p_1, \ldots, p_m \} \) is obtained by subsampling \( C \), then the shape error between \( C \) and \( T_v(M^{(S)}) \) is given by:

\[
E^{(S)}_{TOT} = \sum_{i=1}^{m} ||p_i - T_v(q_i)||^2
\]

and:

\[
E^{(S)}_{PUNCTUAL} = \sum_{i=1}^{m} \phi(p_i, T_v(q_i), v),
\]

where:

\[
\phi(p, q, v) = \begin{cases} 
1 & \text{if } ||p - q||^2 > \gamma_p s \\
0 & \text{otherwise.}
\end{cases}
\]

By imposing \( E^{(S)}_{PUNCTUAL} = 0 \), all the points of \( C' \) are constrained to be at most \( \gamma_p \times s \) pixels far for the projection of the model points into \( D \), where \( \gamma_p \) is a prefixed threshold and \( v = (t_x, t_y, \theta, s) \).

Shape comparison performed using point-to-point squared distance as in Eq. (5)-(6) is much more accurate than common model based shape verification techniques (e.g., chamfer distance or Hausdorff distance). In fact, we do not only check that the generic point \( q_j \), once projected into \( D(T(q_j)) \), is sufficiently close to any edge point \( p' \) of \( D \). We also impose that the distance must be computed exactly with the j-th point \( p_j \) of \( C' \). This ordering implicitly introduces context information in the point matching procedure. Conversely, chamfer distance and other common methods for model-based shape verification are context-free, which in images with thick textures and cluttered backgrounds can bring to a high number of false positives [1]. Our experimental results (Section 6) confirm the low number of false positives obtained with this technique.

In order to be effective, searching for all the possible transformations \( T_v \) in the parameter space and curve \( C \) in \( D \) must be performed efficiently. In the next section we show how this can be done.

4. Efficient curve matching mixing topological and geometric information

From the assumption to deal with only similarity transformations, the correspondence between 2 pairs of points (e.g., \( \hat{p}_1, \hat{p}_2 \in D \) and \( q_1, q_2 \in M^{(S)} \)) is sufficient to uniquely determine the values of the parameter vector \( v \) characterizing \( T_v \). In fact, from \( T_v(q_j) = \hat{p}_i \) (\( i = 1, 2 \)), we obtain the following linear system:

\[
x_1 = t_x + x_1' \cdot s \cdot \cos \theta - y_1' \cdot s \cdot \sin \theta
\]

\[
y_1 = t_y + x_1' \cdot s \cdot \sin \theta + y_1' \cdot s \cdot \cos \theta
\]

\[
x_2 = t_x + x_2' \cdot s \cdot \cos \theta - y_2' \cdot s \cdot \sin \theta
\]

\[
y_2 = t_y + x_2' \cdot s \cdot \sin \theta + y_2' \cdot s \cdot \cos \theta
\]

where \( \hat{p}_1 = (x_1, y_1) \), \( \hat{p}_2 = (x_2, y_2) \), \( q_1 = (x_1', y_1') \), \( q_2 = (x_2', y_2') \). We fix \( q_1 = \hat{q}_1 \) and \( q_2 = \hat{q}_2 \) (remember that \( M^{(S)} = \{ q_1, \ldots, q_m \} \) and we search for high curvature points in \( D \) as candidate points for the base of the finger.

We use the Kanade, Lucas and Tomasi features (KLT features) [8] to select in the current frame \( I \) those points having at least two strong edge directions (we refer to [8] for more details). Such “salient” corner points do not necessarily correspond to edge points of \( D \), but are usually at most 2-3 pixels far from the edge points of \( D \). It is then quite easy to project KLT points onto \( D \) points by selecting for each KLT point \( p \) the closest edge point \( p' \) in a \( 5 \times 5 \) local neighborhood of \( p \). Let us call \( K \) the set of all such
p' points in D (see Figure 3). By matching a pair of points of K with the pair \((q_1, q_m)\) makes it possible to find the parameters \(v\) of transformation \(T_v\) using Eq. (8-11).

Concerning \(C\), we observe that it is composed of consecutively adjacent points of \(D\), as well as \(M^{(S)}\) is a set of pairwise adjacent points represented in the model reference frame. For this reason, candidate \(C\) curves to be matched with \(M^{(S)}\) can be found by following edge adjacency relation among points in \(D\). However, it is well known that the edge detection process is quite unstable, produces a lot of noise and, most important, edge line segments are very often interrupted due to low contrast in the image (e.g., see Figures 2 (a) and 3 (b)). Nevertheless, searching for \(C\) can be efficiently achieved by adequately structuring edge information. To this aim we construct a graph \(G = (N, A)\) representing the topological structure of \(D\): salient edge points are associated with nodes of the graph \(N\) and connected lines with arcs \(A\). More formally, let us call an endpoint an edge point which is adjacent to only one other edge point (i.e., it is the end of a line) and a junction point an edge point adjacent to more than two points (hence, it is the conjunction of two or more lines). If \(S\) is the set of such topologically salient edge points of \(D\), we can construct a graph \(G = (N, A)\) by biuniquely associating each node in \(N\) with a point in \(K \cup S\). Moreover, if \(l\) is a line (i.e., a set of pairwise adjacent edge points) connecting \(p_1 \in K \cup S\) with \(p_2 \in K \cup S\), and \(\nu \in N\) is associated with \(p_1\) and \(\nu \in N\) is associated with \(p_2\), then the arc \(e = (\nu, \nu)\) is added to \(A\) and the set of points of \(l\) is associated with \(e\). The length of \(l\) (number of points of \(l\)) is associated with \(e\) as well.

To deal with missing edge points due to low contrast in the image, we add virtual arcs to \(A\). Given two endpoints \(p_1\) and \(p_2\), respectively associated with \(\nu\) and \(\nu\), and whose Euclidean distance \(d = |p_1 - p_2|\) is such that \(d \leq \gamma_w\), if it happens that there is not any path in \(G\) connecting \(\nu\) with \(\nu\), composed of only real arcs and with cumulative length less than \(kd\), then the virtual arc \(w = (\nu, \nu)\) is added to \(A\) and associated with length \(d\). \(\gamma_w\) and \(k\) are two thresholds respectively used to check that \(p_1\) and \(p_2\) are close enough and that a possible path in \(D\) connecting \(p_1\) and \(p_2\) is composed of lines whose length is sufficiently greater than \(d\). Finally, \(G\) is partitioned in connected components \(G_1, G_2, \ldots\).

\(G\) and its connected components can be constructed in \(O(n + n^2_k\cdot n)\), where \(n\) and \(n_k\) are, respectively, the total number of edge points and the cardinality of \(K \cup S\). Nevertheless, their help in lower down the computational costs of finding a candidate curve \(C\) in \(D\) for comparison with \(M^{(S)}\) is considerable.

Given a connected component \(G_j\) of \(G\), and the corresponding set of KLT features \(K_j\), for each pair of points \(a, b \in K_j\) if \(a\) and \(b\) are not too far and not too close w.r.t. the expected dimensions of a finger into the image, we match \(a\) with \(q_1\) and \(b\) with \(q_m\). Let \(v^{a,b} = (x_a^b, y_a^b, \theta_a^b, s_a^b)\) be the parameter vector defining the transformation \(T_{a,b}\) such that \(T_{a,v}(a) = q_1\) and \(T_{a,v}(b) = q_m\). \(v^{a,b}\) is found solving Eq. (8)-(11).

Using \(T_{a,v}\) we project the points of \(M^{(S)}\) into \(D\) and \(R\) and we compute \(E(R)\) and \(E(D)\). The hypothesis on the match between \(a\) and \(q_1\) and \(b\) and \(q_m\) is rejected if \(E(R)\) and \(E(D)\) are greater than two prefixed thresholds depending on \(h_{a,b}\). This first test using the appearance model helps in rejecting a lot of false matching hypotheses but it is clearly not sufficient in finding a finger shape without using \(M^{(S)}\). Then, if \(E(R)\) and \(E(D)\) are small enough, we search for a curve \(C\) in \(D\) to compare with \(M^{(S)}\).

Since \(a, b \in K_j\), there exist two nodes \(\nu, \nu\) which are respectively associated with \(a\) and \(b\). Hence, we perform a depth-first visit of \(G_j\), starting from \(\nu\), searching for a path \(x = \{e_1, e_2, \ldots\}\) which possibly connects \(\nu\) to \(\nu\) and such that the curve \(C\) given by the concatenation of the lines \(l_1, l_2, \ldots\) to \(C\), similar to \(M^{(S)}\).

Given a path \(x\) in \(G_j\) and the corresponding curve \(C\) in \(D\), we can uniformly subsample \(C\) to obtain \(C'\) with \(M^{(S)}\) using Eq. (5) and (6). The subsampling step which allows us to uniformly select points from \(C\) must be proportional to the scale factor \(s_a^b\). Let \(f^{a,b}\) be the subsampling step \(f^{a,b} \propto s_a^b\). If \(x'\) is a partial path in \(G_j\) starting from \(\nu\), the error introduced adding a new (real) arc \(e\) to \(x'\) can be computed by taking the line \(l\) corresponding to \(e\) and using \(f^{a,b}\) to obtain a subsampled set of points \(l'\). \(l'\) is compared with the corresponding “unexplored” subset of \(T_{a,v}^{-1}(M^{(S)})\) and \(e\) is added to \(x'\) if \(E^{(S)}_{T_{a,v}}\) remains below a given threshold and \(E^{(S)}_{PUNCTUAl}\) remains equal to 0. Vice versa, if \(e\) is a virtual arc whose length is \(d\), we can add \(e\) to \(x'\) if the cumulative length of the virtual part of \(x'\) remains below a given threshold.

The constraints above mentioned allow the system to rapidly prune the search space because very few lines in \(D\)
are perceptually similar to the searched for shape (a finger-like curve, in our case). Thus, wrong paths in exploring $G_j$ are cut out as soon as the accumulated error becomes excessive, which typically happens when we meet a point in $D$ which is farther than $n_{p}^{x, b} h$ pixels from $T_{\nu, \nu} (M^{(S)})$. The proposed geometric constraint satisfaction method is a yes-or-not approach which does not provide an estimation of the finger detection reliability, just like a “standard” classification method (e.g., [11]). In other works (e.g., [9]) rigid part detection is much less reliable and this must be taken into account using a probabilistic framework.

5. Searching for valid finger configurations

The process described in the previous section is extended to find the shape of a hand, composed of different fingers. Using the same terms of the previous section, once a valid path $x$, corresponding to curve $C$, is found between $\nu$ and $v$, we continue to search for other neighboring fingers in the same connected graph component $G_j$. The presence of neighboring fingers lowers down the likelihood that $x$ corresponds to a false positive because the shape which must be found in $D$ becomes more and more complex.

Assuming that $K P t (x)$ returns the set of the KLT points corresponding to the nodes of $x$, searching for a second finger is performed by repeating the process described in the previous section using a second pair of starting points $a', b'$ selected in such a way that:

$$a', b' \in K_{p} - K P t(x) \cup \{a, b\} \land \neg((a' = a) \land (b' = b)).$$

(12)

Note that in Eq. (12) we permit that one (but not both) between $a$ and $b$ can be reused in order to allow the system to detect adjacent fingers sharing a KLT point (Figure 3).

In order to take into account the relative positions of different fingers, we transform a search in parameter space in a constraint satisfaction problem. Essentially, instead of searching for all the possible (discrete) positions of a finger w.r.t. another, we check that groups of separately found fingers have a valid displacement the one w.r.t. the other fingers (e.g., see Figures 4). To deal with out-of-plane hand rotations we use different (shorter) shape models; other shape models are necessary for dealing with thumb. A total amount of only 9 shape models is used, iterating the recognition process for 9 different model curves $M_{1}^{(S)}, \ldots, M_{9}^{(S)}$.

No training is necessary to learn the shape or the appearance models and the system can deal with different human beings without any parameter tuning. Concerning the scenario features, our system is able to work with different lighting conditions, cluttered background, possible occlusions, presence in the scene of other moving objects or other skin-colored patches. The presence of objects with a skin-like color just behind the hand is not a problem as well but it usually deteriorates the detection rate since it degrades the image contrast, hence making worse the edge detection process. When pre-processing operations (see Section 2) makes it possible to select a ROI mask, detection speed is achieved at 10-20 fps with $320 \times 240$ images on a Pentium IV 2.4 Ghz. The hand detection process is performed in each frame of the input video sequence. However, results for past frames are used to improve detection rate. In fact, if no successful detection has been obtained in the current frame $I_t$ but the system signaled a hand detection in the majority of the past 5 frames $(I_{t-5}, \ldots, I_{t-1})$ then we signal a hand in frame $I_t$ as well, averaging its position using the past positions.

6. Results

We performed two types of tests: the first using a database of still images and the second with an annotated video sequence. In all the experiments a detection is considered valid if the center of the hand signaled by the system overlaps with the real hand represented in the input figure and the direction error is less than 45 degrees w.r.t. the real direction (see Figure 4 for some examples). In the still image test the detection rate is lower because we cannot use information about past detections to guess the presence of a hand.
in the current frame (see the previous section). However, it is the very low false alarm rate (FAR) which allows the system to use detections on past frames to deduce the presence of a hand in the current frame.

For testing the system’s classification accuracy with still images we used the well known Caltech 101 Dataset [3], taking 8994 images out of 9145 (we discarded only too large images). None of these contains hands. Then we added other 113 images containing hands in various positions, different postures, taken both in indoor and outdoor environments and with a nonuniform background. Table 1 shows detection rates and FAR achieved.

For testing the system with a video sequence, we grabbed a video containing 1645 frames in which scenes with the presence of a hand (989 frames) alternate with scenes without any hand (656 frames). Also in this case we used cluttered background, different lighting conditions and hands in different positions and postures. The results are shown in Table 1. Note that the simple technique to average w.r.t. the past 5 frames (Section 5) makes it possible to largely increase the detection rate.

### 7. Conclusions

We have presented a hand detection system working in real time and able to recognize hands independently of the person identity, hand position and which is robust to different lighting condition changes as well as cluttered background images. Different configurations of the hand can be recognized by searching for the model of a finger and then clustering detected fingers which satisfy anatomic geometric constraints. Few assumptions have been done on the articulated object nature and the proposed method can be applied to recognition of the whole human figure or other articulated objects.

### 8. Acknowledgments

This work was partially supported by the Italian Ministry of University and Scientific Research within the framework of the project ”Ambient Intelligence: event analysis, sensor reconfiguration and multimodal interfaces” (2006-2008).

### References


