Real-time articulated hand detection and pose estimation

Giorgio Panin, Sebastian Klose, Alois Knoll

Technische Universität München, Fakultät für Informatik
Boltzmannstrasse 3, 85748 Garching bei München, Germany
{panin,kloses,knoll}@in.tum.de

Abstract. We propose a novel method for planar hand detection from a single uncalibrated image, with the purpose of estimating the articulated pose of a generic model, roughly adapted to the current hand shape. The proposed method combines line and point correspondences, associated to finger tips, lines and concavities, extracted from color and intensity edges. The method robustly solves for ambiguous association issues, and refines the pose estimation through nonlinear optimization. The result can be used in order to initialize a contour-based tracking algorithm, as well as a model adaptation procedure.

1 Introduction

Hand tracking is an important and still challenging task in computer vision, for many desirable applications such as gesture recognition for natural Human-Computer Interfaces (HCI), virtual devices, and tele-manipulation tasks (see for example the review work [4, Chap. 2]).

In order to reduce the problem complexity, dedicated devices (*data gloves*) have been developed, directly providing the required measurements for pose estimation. However, such devices somehow constrain the field of applicability as well as the motion freedom of the user, at the same time requiring a carefully calibrated and often expensive setup (particularly when infrared cameras and markers are involved).

In a purely *markerless* context, [10] employs a *flock of features* for tracking, while detection is performed by an AdaBoost classifier [11] trained on Haar features [17]; however, although showing nice robustness properties, both procedures do not provide any articulated pose information, but only the approximate location over the image. The most well-known approaches to articulated tracking in 2D and 3D [14, 15, 6, 2, 13, 5] are instead based on *contours*, which provide a rich and precise visual cue, and profit from a large pool of *predicted features* (contour points and lines) from the previous frame, through dynamical data association and local search.

However, all of these tracking approaches assume at least a partially manual initialization (hand *detection*), providing an initial localization of the hand. Hand detection amounts to a global search in a high-dimensional parameter space,

using purely *static* data association [12, 1] and fusion mechanisms, that strongly limit the amount of distinctive features that can be reliably matched to the model.

This paper deals with the problem of fully automatic, articulated hand detection, using static feature correspondences (points and lines) extracted from two complimentary modalities, namely skin color and intensity edges.

The paper is organized as follows: in Section 2 we first describe the visual cues, and the association criteria, used in order to obtain the geometric feature correspondences to the generic model. Afterwards, Section 3 describes the articulated pose estimation procedure. Experimental results are provided in Section 4, together with a discussion of possible development roads.

2 Visual features for hand detection

From the input image, we detect two kinds of features: fingertips and concavities, obtained from skin color segmentation, and finger lines, detected along the intensity edges.

2.1 Point features from color segmentation

The input image is first converted to HSV color space, which is well-suited for skin color segmentation, and pixels are classified through a 2D Gaussian Mixture Model (GMM) in the Hue and Saturation channels [18].

Afterwards, we compute the *convex hull* of the main connected component (blob), and note that most of the time, fingertips and concavities are approximately located, respectively, on the convex hull vertices and *concavity defects* [3]: the latter are defined as the maximum-distance points to the respective hull segments (left side of Fig. 1).

In order to identify the fingertips among the hull vertices, all vertices and defects have to be properly thresholded and classified. In particular, the overall palm scale is estimated by r_{palm}, the radius of the maximum-inscribed circle (MIC) within the color blob; this proves to be robust with respect to the fingers configuration, and the center provides also a rough position estimate. The MIC is quickly computed, by maximizing the distance transform [7].

Fig. 1 also illustrates the scheme used to identify the hull points representing fingertips and the concavities representing palm points. For this purpose, for each hull segment we consider two scale-independent and dimensionless indices, related to the palm size r: the maximum concavity depth D/r, and the length L/r; with these values, we classify segments according to four cases indicated in the picture.

The next step consists in *merging* the fingertip points, by removing too short segments from the sequence (cases 3 and 4 with $L/r < t_L^-$), and averaging their endpoints. If a sequence of 5 fingertips is obtained, we identify the thumb and the small finger, by looking for the largest, clock-wise angle between fingertips, measured around the palm center c. Otherwise, the algorithm recognizes the

case of insufficient information, and returns a detection failure, thus avoiding any attempt to further processing and pose estimation.

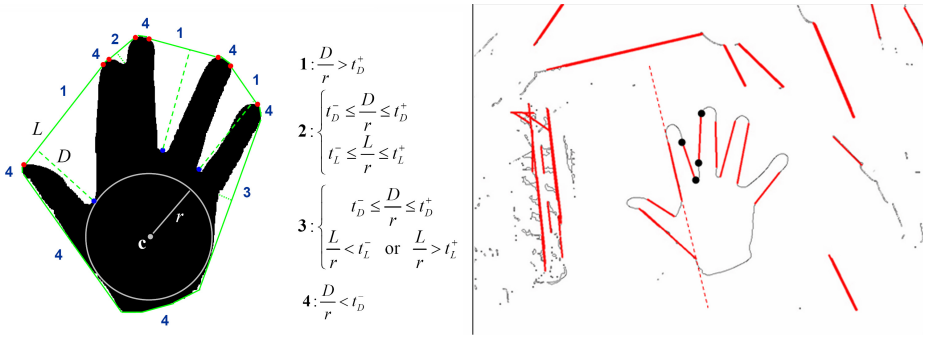

Fig. 1. Left: Using the convex hull for fingertip and concavity detection. Right: Segments detected via the probabilistic Hough transform.

2.2 Line features from intensity edges

As a second modality, we use intensity edges. In particular, from the Canny edge map we detect straight line segments, through a probabilistic Hough transform [9], that can be matched to the model lines. The right side of Fig. 1 shows an example of line detection.

A segment correspondence in principle provides 2 point correspondences (i.e. 4 measurements). However, as we can see from Fig. 1 (right side), the endpoints of the segment are not as well localized as the *line* itself (in terms of direction and distance to the origin); therefore, the most reliable matching can be obtained by pure *line* correspondences.

A line is described in homogeneous coordinates by a 3-vector $\mathbf{l} = (a, b, d)^T$, defined by the equation

$$ax + by + d = 0 \Leftrightarrow \mathbf{l}^T\mathbf{x} = 0 \tag{1}$$

with $x = (x, y, 1)^T$ the homogeneous coordinates of a point belonging to $\mathbf{l}$. We also assume, without loss of generality, that the orthogonal vector $\mathbf{n} = (a, b)^T$ is normalized ($a^2 + b^2 = 1$), so that the third component d represents the distance of the line to the origin.

2.3 Data association to a generic model

For pose estimation, the detected features have to be associated to the correct model features from a generic model.

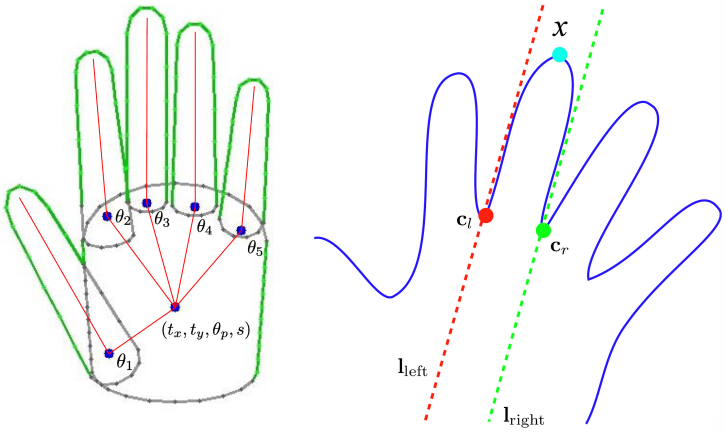

Fig. 2. Left: a simple shape model, made up of ellipses and rectangles; red lines represent the 2D skeleton; green contour lines are used for matching. Right: Definition of model line and point features.

For this purpose, we abstractly describe a *finger* model as a triplet of points and two lines (right side of Fig. 2): the fingertip ($\mathbf{x}$), the left and the right concavity points ($\mathbf{c}_l$, $\mathbf{c}_r$), two lines representing the left and the right edges ($\mathbf{l}_{\text{l}}$, $\mathbf{l}_{\text{r}}$), and a set of flags, signalling if the respective feature has been detected in the image.

The candidate points obtained from the convex hull are first considered, by taking parallel neighboring lines $\mathbf{l}_{\text{left}}$ and $\mathbf{l}_{\text{right}}$, previously aligned in order to have the same normal directions, forming a *candidate finger* if:

$$|\mathbf{n}_l^T \mathbf{n}_r| > 1.0 - \cos \epsilon_{\text{angle}} \tag{2}$$

$$\left[\left(\mathbf{l}_{\text{l}}^T \mathbf{x} > 0\right) \wedge \left(\mathbf{l}_{\text{r}}^T \mathbf{x} < 0\right)\right] \vee \left[\left(\mathbf{l}_{\text{l}}^T \mathbf{x} < 0\right) \wedge \left(\mathbf{l}_{\text{r}}^T \mathbf{x} > 0\right)\right] \tag{3}$$

$$t^-_{\text{tipDist}} < \left|\mathbf{l}_{\text{l,r}}^T \mathbf{x}\right| < t^+_{\text{tipDist}} \tag{4}$$

$$\left(\left|\mathbf{l}_{\text{l}}^T \mathbf{c}_{\text{l}}\right| < t_{\text{concDist}}\right) \wedge \left(\left|\mathbf{l}_{\text{r}}^T \mathbf{c}_{\text{r}}\right| < t_{\text{concDist}}\right) \tag{5}$$

These conditions state that: the lines have to be approximately parallel, i.e. the angle between the two normals is checked against a threshold ϵ_{angle} (eq. (2)); the fingertip $\mathbf{x}$ should lie between both lines (eq. (3)); the fingertip $\mathbf{x}$ should be close enough to both lines $\mathbf{l}$ (but also not too close, eq. (4)); and finally, each concavity point $\mathbf{c}$ should be quite close to the respective line $\mathbf{l}$ (eq. (5)).

3 Articulated pose estimation from corresponding features

After establishing the correct data association, the next problem is to estimate the hand pose, minimizing the re-projection error of all detected model features (points and lines) with respect to their noisy measurements.

In our approach, the hand model is an articulated skeleton (left side of Fig. 2), composed of 6 rigid links (one for the palm and for each of the fingers). In particular, the palm undergoes a 2D similarity transform (roto-translation and uniform scale), with *4-dof*, while each finger carries an additional rotation angle θ_i, so that the 9 pose parameters $\mathbf{p}$ are

$$\mathbf{p} = (t_x, t_y, \theta_p, s, \theta_1, \theta_2, \theta_3, \theta_4, \theta_5)^T \tag{6}$$

Starting from the generic model, made up of simple shapes (ellipses and rectangles), we first compute the reference lines and points, by using the same procedure of Sec. 2.3, applied to a rendered image of the model. This has the advantage of keeping generality with respect to the model, at the same time providing the reference features in an automatic way, for a given shape.

3.1 Single-body pose estimation

The pose of each link of the hand in 2D can be represented by a (3×3) homogeneous transform T, projecting points from model to screen coordinates. Moreover, in order to keep generality for the articulated chain, a *parent transform* $\bar{T}$ pre-multiplying T may be present, considered constant for a single-body pose estimation, and possibly belonging to a different transformation group (for example, for each finger $\bar{T}$ may be a full similarity, while T is a single-axis rotation).

For our purposes, we restrict the attention to 2D similarities

$$\bar{T}T = \begin{bmatrix} sR & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \tag{7}$$

where s is a scale factor, R a (2×2) orthogonal matrix, and $\mathbf{t}$ the translation vector. This is a good model for planar hand estimation problems, where the distance to the camera center is large enough compared to the hand size.

Point correspondences Given N model points $\mathbf{X}$ and corresponding noisy image measurements $\mathbf{x}$ in homogeneous coordinates

$$\mathbf{X}_i = (X_i, Y_i, 1)^T; \ \mathbf{x}_i = (x_i, y_i, 1)^T \tag{8}$$

we look for the optimal transformation T^* that projects the model points $\mathbf{X}_i$ "as close as possible" to the measured points $\mathbf{x}_i$, i.e. satisfying

$$\bar{T}T \cdot \mathbf{X}_i = \mathbf{x}_i, \forall i \tag{9}$$

We can *pre-process* the data points $\mathbf{x}_i$, by writing

$$T \cdot \mathbf{X}_i = \bar{T}^{-1}\mathbf{x}_i \equiv \bar{\mathbf{x}}_i, \forall i \tag{10}$$

where $\bar{\mathbf{x}}_i = (\bar{x}_i, \bar{y}_i, 1)^T$ become the data values for pose estimation.

If $T(\mathbf{p})$ has a linear parametrization $\mathbf{p}$, then we have

$$T_{2\times 3}(\mathbf{p})\, \mathbf{X}_i = \hat{\mathbf{X}}_i \cdot \mathbf{p} \tag{11}$$

where the $(2 \times n)$ matrix $\hat{\mathbf{X}}_i$ is a function of $\mathbf{X}_i$, and $T_{2\times 3}$ is the upper (2×3) submatrix of T (non-homogeneous coordinates).

Line correspondences We formulate the line correspondence problem as follows [12]: given n_l model segments $(\mathbf{L}_i^1, \mathbf{L}_i^2)$ matched to image lines $\mathbf{l}_i = (\mathbf{n}_i, d_i)^T$, find T such that both projected endpoints lie on $\mathbf{l}_i$

$$\forall i : \mathbf{l}_i^T \left(\bar{T}T \cdot \mathbf{L}_i^1\right) = \mathbf{l}_i^T \left(\bar{T}T \cdot \mathbf{L}_i^2\right) = 0 \tag{12}$$

In the above equation, the term $\bar{T}$ can again be removed, by pre-processing the data lines $\mathbf{l}$

$$\bar{\mathbf{l}}^T = \mathbf{l}^T \bar{T} = (\bar{\mathbf{n}}^T, \bar{d}) \tag{13}$$

which can be seen as the *dual* version of (10).

Finally, if the parametrization is linear in $\mathbf{p}$, then the estimation problem becomes linear as well: by denoting with $\hat{\mathbf{L}}_i^1, \hat{\mathbf{L}}_i^2$ the equivalent matrices to $\mathbf{L}_i^1, \mathbf{L}_i^2$, respectively (as in the previous Section) we can write the two equations (12) in a more compact way

$$\hat{L}_i \cdot \mathbf{p} + \bar{\mathbf{d}}_i = \mathbf{0};\ \hat{L}_i = \begin{bmatrix} \bar{\mathbf{n}}_i^T \hat{\mathbf{L}}_i^1 \\ \bar{\mathbf{n}}_i^T \hat{\mathbf{L}}_i^2 \end{bmatrix};\ \bar{\mathbf{d}}_i = \begin{bmatrix} \bar{d}_i \\ \bar{d}_i \end{bmatrix} \tag{14}$$

Single-body pose estimation Under a linear parametrization $T(\mathbf{p})$, given n_l line and n_p point correspondences respectively, we can write the single-body LSE estimation problem

$$\mathbf{p}^* = \min_{\mathbf{p} \in \Re^d} \left(\sum_{i=1}^{n_l} \left\| \hat{L}_i \cdot \mathbf{p} + \bar{\mathbf{d}}_i \right\|^2 + \sum_{j=1}^{n_p} \left\| \hat{\mathbf{X}}_j \cdot \mathbf{p} - \bar{\mathbf{x}}_j \right\|^2 \right) \tag{15}$$

with $\hat{L}_i, \bar{\mathbf{d}}_i$ defined in (14), and the pre-processed measurements $\bar{\mathbf{l}}_i, \bar{\mathbf{x}}_j$ given by (13) and (10), respectively.

This problem is linear in $\mathbf{p}$, and can be solved in one step, via the singular value decomposition (SVD) technique.

3.2 Articulated pose estimation

Recovering articulated pose parameters is accomplished by a two-step procedure.

Initialization In order to initialize the articulated parameters, we use a hierarchical approach:

1. We examine the skeleton tree, starting from the root (i.e. the palm of the hand) and estimating its similarity parameters alone. For this purpose, two point correspondences (concavity defects, palm center) are sufficient to estimate the similarity parameters $p_1, ..., p_4$ [16].
2. Afterwards, for all child nodes (i.e. the fingers), we use the parent T estimate as a reference matrix $\bar{T}$ for each link, and employ all available point and line correspondences in order to estimate its pose as in (15)

This approach does not require any initial guess for $\mathbf{p}$, and usually provides a good initial estimate $\mathbf{p}_0$.

Nonlinear LSE refinement using contour points The geometric error of the articulated chain with respect to the global pose parameters $\mathbf{p}$ has an overall nonlinear form, due to the fact that intermediate T matrices are multiplied along the skeleton, in order to produce the finger transforms, and each of them is a function of a subset of pose parameters.

For this purpose, the measurements are obtained in a standard way [8], by sampling a set of m *contour points* $\mathbf{y}_i^f$ and image normals $\mathbf{n}_i^f$, uniformly over the articulated chain (Fig. 3). The contour points are re-projected and matched to the closest intensity edges $\mathbf{z}_i$ at each Gauss-Newton iteration, providing dynamic data association with a much larger set of measurements for the pose estimation problem.

By writing the *normal equations* (for sake of simplicity, with equal weights for all features), we have

$$\sum_{i=1}^{m} J_{\mathbf{y}_i}^T \mathbf{n}_i \mathbf{n}_i^T J_{\mathbf{y}_i} \delta\mathbf{p} = \sum_{i=1}^{m} J_{\mathbf{y}_i}^T \mathbf{n}_i^T \mathbf{e}_{\mathbf{y}_i} \tag{16}$$

where $\delta\mathbf{p}$ are the *incremental* pose parameters w.r.t. the previous iteration, and the (2×9) Jacobian matrices

$$J_{\mathbf{y}_i} = \left[\frac{\partial \mathbf{y}_i^f}{\partial p_1} \cdots \frac{\partial \mathbf{y}_i^f}{\partial p_9} \right] \tag{17}$$

provide the derivatives of screen projections w.r.t. the pose parameters, for each sample contour point. Fig. 3 shows an example of non-linear pose estimation.

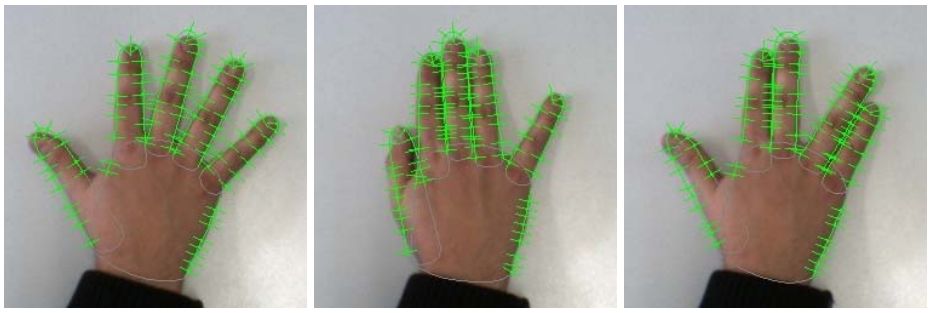

Fig. 3. Articulated pose estimation with contour points and normals, after Gauss-Newton optimization.

4 Experimenal results

We provide here some experiments, showing the performance of the detector for different, more or less crucial hand poses (with closed fingers).

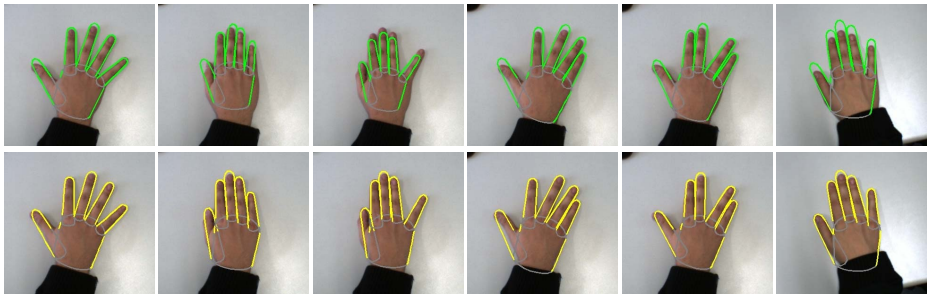

Fig. 4. Hand detection result for different postures. Top row: initialization; bottom row: nonlinear LSE refinement.

The first row of Fig. 4 show the result of the pose initialization algorithm of Sec. 3.2, obtained after detection of line and point features, with our data association procedure. It can be seen, that in all cases the detected pose of the hand is close to the correct one; however, the initialization privileges the palm parameters, and uses only the basic features from the detection step.

The second row of Fig. 4 show the result of the subsequent pose refinement (Sec. 3.2) after 10 Gauss-Newton iterations. In all cases the tracker converges to a correct pose estimate, despite the mismatched sizes of the model fingers to the real size of the subject. As already emphasized, this step achieves the correct overall scaling and matching, by uniformly optimizing over the full contour features (yellow lines), and ignoring the wrist and internal model lines (gray).

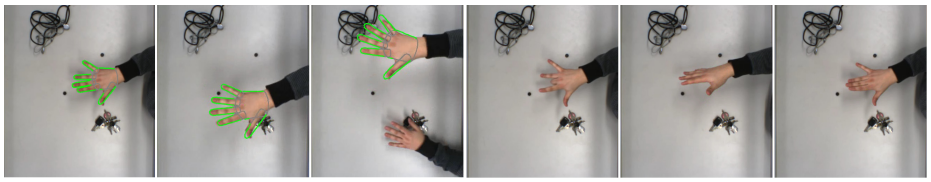

Fig. 5. Left: detection performance with background clutter. Right: detection failures.

By considering more challenging situations, in the first 3 frames of Fig. 5 we show the detection performance in presence of clutter, both conerning intensity edges and other skin-colored objects. The last 3 frames show some examples of data association failures: bent fingers, out-of-plane rotations, and too close fingers. In the whole sequence, all of these case are recognized by the detector, that does not attempt to perform any incorrect pose estimation.

In order to provide numerical evaluations, we tested the algorithm against ground-truth data, obtained by a manual alignment of the model to the above given images.

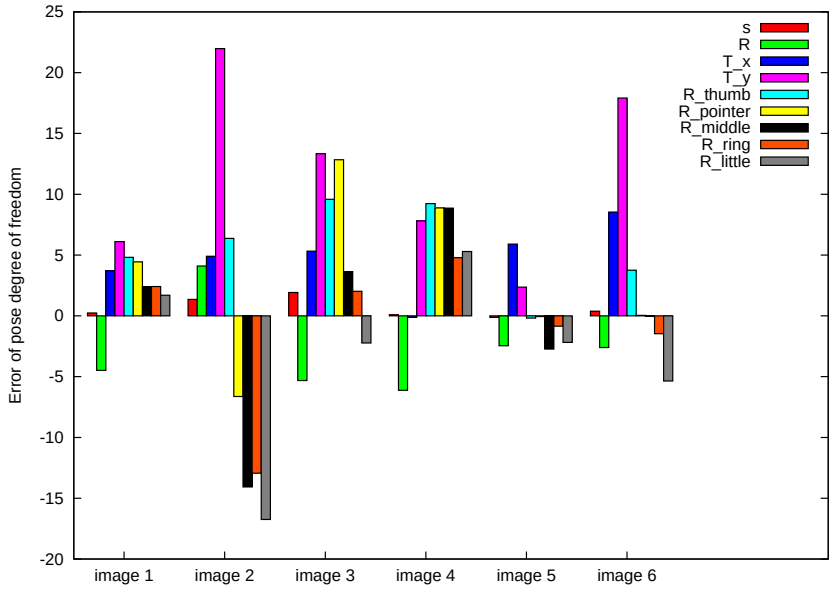

Fig. 6. Error of the pose detection w.r.t. the visually computed ground truth. (s, R, t_x, t_y) = palm scale, rotation and translation; R_{thumb} = rotation angle for the thumb, etc.

Denoting with $\mathbf{p}^T$ and $\mathbf{p}^E$ the true and estimated pose respectively, we compute the estimation error as $\mathbf{e}_i = \mathbf{p}_i^T - \mathbf{p}_i^E$. Fig. 6 shows the error components for each of the images. In particular, translations (t_x, t_y) are given in pixels, and rotation angles in 10^{-1} degrees.

The presented algorithm was tested on an *Intel Core 2 Duo* with 2,33GHz, 2GB RAM and a *NVIDIA 8600GT* GPU with 256MB graphics memory. As operating system we use *Ubuntu Linux 8.04*. For video input, an *AVT Guppy F033C* firewire camera was used to capture frames with a resolution of 656×494 at 25Hz. Using this setup the algorithm averagely performs at 5 FPS.

5 Conclusion and future work

In this paper, we presented a hand detection and pose estimation methodology, based on a generic model with articulated degrees of freedom, using geometric feature correspondences of points and lines. In particular, the method has been demonstrated for a planar case, with similarity transform and planar fingers motion.

A full 3D detection involves more complex issues, which can be best dealt with by using multiple views and related features association. However, the ideas

presented so far can serve as as a basis for a more complex approach, where multiple convex hulls are used in order to detect fingertips and palm concavities, while detected edge segments can be (at least in part) associated to individual finger links, by using the detected point information.

References

1. Y. Bar-Shalom. *Tracking and data association.* Academic Press Professional, Inc., San Diego, CA, USA, 1987.
2. C. Bregler and J. Malik. Tracking people with twists and exponential maps. In *CVPR '98: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, page 8, Washington, DC, USA, 1998. IEEE Computer Society.
3. Yaroslav Bulatov, Sachin Jambawalikar, Piyush Kumar, and Saurabh Sethia. Hand recognition using geometric classifiers. In *ICBA*, pages 753–759, 2004.
4. T. E. de Campos. *3D Visual Tracking of Articulated Objects and Hands.* PhD thesis, University of Oxford, 2006.
5. J. Deutscher and I. D. Reid. Articulated body motion capture by stochastic search. *Int'l Journal of Computer Vision*, 61(2), 2005.
6. Tom Drummond and Roberto Cipolla. Real-time tracking of highly articulated structures in the presence of noisy measurements. In *ICCV*, pages 315–320, 2001.
7. Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Distance transforms of sampled functions. Technical report, Cornell Computing and Information Science, September 2004.
8. Chris Harris. Tracking with rigid models. pages 59–73, 1993.
9. N. Kiryati, Y. Eldar, and A. M. Bruckstein. A probabilistic hough transform. *Pattern Recogn.*, 24(4):303–316, 1991.
10. Mathias Kölsch and Matthew Turk. Fast 2d hand tracking with flocks of features and multi-cue integration. *Computer Vision and Pattern Recognition Workshop*, 10:158, 2004.
11. Mathias Kölsch and Matthew Turk. Robust hand detection. In *FGR*, pages 614–619, 2004.
12. David G. Lowe. Fitting parameterized three-dimensional models to images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13(5):441–450, 1991.
13. John MacCormick and Michael Isard. Partitioned sampling, articulated objects, and interface-quality hand tracking. In *Proceedings of the European Conference on Computer Vision (ECCV'00) - volume 2*, number 1843 in Lecture Notes in Computer Science, pages 3–19, Dublin, Ireland, June 2000. Springer-Verlag.
14. James M. Rehg and Takeo Kanade. Digiteyes: Vision-based human hand tracking. Technical report, Pittsburgh, PA, USA, 1993.
15. B. Stenger, P. R. S. Mendona, and R. Cipolla. Model-based 3d tracking of an articulated hand. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2:310, 2001.
16. Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13(4):376–380, 1991.
17. Paul A. Viola and Michael J. Jones. Robust real-time face detection. In *ICCV*, page 747, 2001.
18. Z. Zivkovic. Improved adaptive Gaussian mixture model for background subtraction. In *Proceedings of the 17th International Conference on Pattern Recognition*, volume 2, 2004.