Segmentation-Free, Area-Based Articulated Object Tracking

Daniel Mohr and Gabriel Zachmann

Clausthal University

Abstract. We propose a novel, model-based approach for articulated object detection and pose estimation that does not need any low-level feature extraction or foreground segmentation and thus eliminates this error-prone step. Our approach works directly on the input color image and is based on a new kind of divergence of the color distribution between an object hypothesis and its background. Consequently, we get a color distribution of the target object for free.

We further propose a coarse-to-fine and hierarchical algorithm for fast object localization and pose estimation. Our approach works significantly better than segmentation-based approaches in cases where the segmentation is noisy or fails, e.g. scenes with skin-colored backgrounds or bad illumination that distorts the skin color.

We also present results by applying our novel approach to markerless hand tracking.

1 Introduction

Today, tracking articulated objects (e.g. human bodies or hands) is of more interest than ever before. Consequently, robust detection and recognition of articulated objects in uncontrolled environments is an active research area and still a challenging task in computer vision. Applications can be found in more and more areas, such as games (e.g. the Kinect), gesture recognition as next generation "touchless" touchscreen, gesture recognition in mobile devices to improve application control, or rehabilitation, to mention only some of them.

When utilizing the human hand as an input device (as opposed to the whole human body), cameras of very high resolution are mandatory. Furthermore, the hand has about 26 DOF. Thus, smart algorithms and a lot of computation power is needed to handle the large configuration space.

Consequently, one has to find a good compromise between accuracy and computation time. On the one hand, the approach has to be robust enough, so we want to eliminate as many error sources as possible (like edge detection or segmentation). On the other hand, the approach still needs to be computationally very efficient. Many tracking systems utilize temporal coherence in order to save computation time: only the close neighborhood of the object's position and pose from the previous frame are scanned for updating position and pose. But this has the serious disadvantage that it often leads to drifting and typically, after

G. Bebis et al. (Eds.): ISVC 2011, Part I, LNCS 6938, pp. 112–123, 2011.

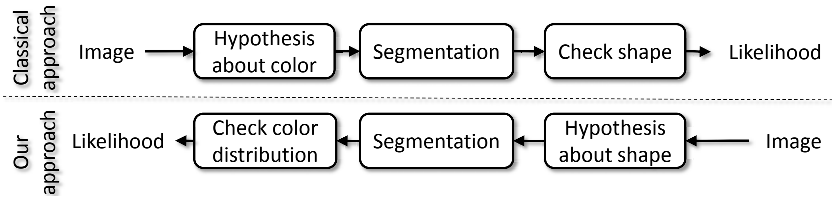

Fig. 1. We turn the classical approach upside-down and first test different shape hypotheses and then check the color distribution while classical approaches first estimate the color distribution (which is error-prone because it is not known well for real images) and then check the shapes

some hundred frames, the object is lost. In order to achieve better robustness, we propose to do tracking by per-frame detection.

Most approaches need some kind of low-level feature extraction as a preprocessing step, e.g. edge detection or foreground segmentation. There exists a large body of previous work to reduce the disadvantages of such a feature extraction.

Our novel method works, as an overview, as follows. For each possible pose and orientation of the model, we precompute its fore- and background information. This is then transformed into an object descriptor, which we will refer to as *template* in the following. It is crucial to use a very robust similarity measure between the templates and the input image. We propose a new similarity measure based on the color distribution divergence. Our similarity measure works directly on the color image. The basic idea is to compare the color distribution of the foreground and background regions for each hypothesis. The more divergent the color distributions are, the higher is the probability that the hypothesis is observed. In a way, our novel approach can be viewed as turning the classical matching approach upside-down: instead of first forming a hypothesis about the color distribution of the object and then checking whether or not it fits the expected shape, we first form a hypothesis about the shape and then check whether or not the color distribution fits the expectation (illustrated in Fig. 1).

Due to the large configuration space of the human hand, we need to speed up the localization and recognition. We propose a new coarse-to-fine approach where the step size depends on the template shapes and is computed offline during template generation. In addition, we combine a template hierarchy with this approach to further reduce the computation time from $O(\text{n})$ to $O(\text{log n})$, n = #templates. Basically, we coarsely scan the input image with the root node of the hierarchy and use the k best matches for further tree traversal.

Our *main contributions* are:

1. A novel and robust similarity measure based on a new kind of color distribution divergence, which does not need any segmentation or other error-prone feature extraction.

2. Our approach can be trivially extended to include other features (e.g. depth values using ToF-cameras or the Kinect) because our approach is based on

comparing multivariate distributions between object foreground and background, and not any color specific properties.

3. A coarse-to-fine and hierarchical approach for fast object detection and recognition. Based on the template description, we estimate the smallest possible distance between two local maxima in the confidence map. We use this knowledge to determine the scan step size and combine it with a multi-hypothesis and hierarchical template matching.

2 Related Work

Most approaches for articulated object tracking use edge features or a foreground segmentation. These features are used to define a similarity measure between the target object and the observation (input image). Most approaches are model-based. Basically, for each parameter set (pose) of the object a template is generated and used for matching. Typical edge-based approaches to template matching use the chamfer [1],[2] or Hausdorff [3]. Chamfer matching for tracking of articulated objects is, for example, used by [4], [5], [6],[7], [8], [9],[10] and [11], the Hausdorff distance by [12],[13] and [14]. The generalized Hausdorff distance is more robust to outliers. The computation of the chamfer distance can be accelerated by using the distance transform of the input image edges. Both, chamfer and Hausdorff distance can be modified to take edge orientation into account [12][8][15]. The main problems of edge-based tracking are the edge responses in the input image. Either the approach needs binary edges or works with intensities itself. In the first case, thresholds for binarization have to be chosen carefully and are not easy to be determined. In the second case, intensity normalization has to be performed.

Segmentation-based approaches apply color or background segmentation. The segmentation result is then compared to the object silhouette. [16,17] uses the difference between the model silhouette and segmented foreground area in the query image as similarity measure. A similarity measure is used by Kato et al. [9]. They use the differences between template foreground, segmentation and intersection of template and segmentation. In [18], the non-overlapping area of the model and the segmented silhouettes are integrated into classical function optimization methods. In [19,20] a compact description of the hand model is generated. Vectors from the gravity center to sample points on the silhouette boundary, normalized by the square root of the silhouette area, are used as hand representation. During tracking, the same transformations are performed to the binary input image and the vector is compared to the database. A completely different approach is proposed by Zhou and Huang [21]. They use local features extracted from the silhouette boundary obtained by a binary foreground segmentation. Each silhouette is described by a set of feature points. The chamfer distance between the feature points is used as similarity measure.

In [22] the skin-color likelihood is used. For further matching, new features, called likelihood edges, are generated by applying an edge operator to the likelihood ratio image. In [23,24,8], the skin-color likelihood map is directly compared with hand silhouettes. The product of all skin probabilities at the silhouette foreground is multiplied with the product of all background probabilities in the template background.

Similar to edge-based matching, segmentation-based approaches either need a binary segmentation, or directly work with the segmentation likelihood map. Thus, the disadvantages of these approaches are binarization errors or false segmentation, i.e. classifying background regions as foreground.

Wang [25] uses a completely different segmentation-based approach by requiring users to wear colored gloves. Each of the colors correspond to a specific part of the hand. A distance measure between two arbitrary hand poses, based on the color coding, is defined. In a preprocessing step, a large database, containing hand descriptors-based on the color glove coding, is generated. During tracking, this database is compared to the hand observed in the input image using a Hausdorff-like distance between the centers of the color regions. The disadvantages of the approach are that a homogeneous background is needed and a special glove is necessary.

We propose an approach that does not need such features like edges or a segmentation. Our method directly works on the color input image. The basic idea is to estimate color distributions of the fore- and background. Here, the foreground/background regions are given by the template descriptor and the corresponding color values in the input image. In other words, our algorithm simultaneously performs shape matching and the target object color distribution estimation.

3 Matching Object Templates

In the following, we will explain our proposed similarity measure between the target object and the input image. A similarity measure, in general, is used to compute the probability that, at a given position and scale in the input image, the target object in a given pose is observed. Henceforth, we assume that for any object pose an object silhouette area is given which is denoted by *template*. The goal is to estimate the probability for the observation of a template at a specific scale in the image. For detection, this can be done at each position and scale in the image.

3.1 Color Divergence-Based Similarity Measure

Our similarity measure is based on the idea that the target object has a different color distribution than its surrounding background. On a very general level, this is similar to classical approaches based on image segmentation. However, in our approach, the only a priori knowledge are the template descriptors of the object shapes in each pose. We do not need a priori knowledge about the object color distribution.

Given a template and a position in the input image, the pixels that correspond to the foreground and the background in the template, resp., are determined.

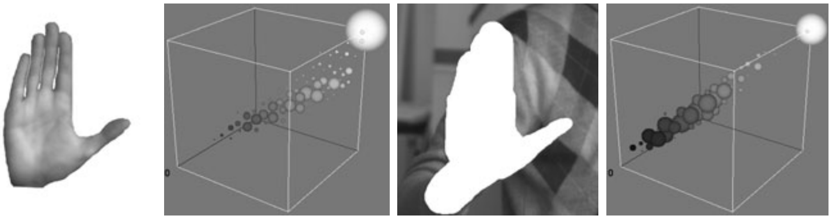

Fig. 2. Example of the color distribution of a human hand and a background. The image is decomposed into the hand and the background. The first two images show the hand and the corresponding 3D color histogram. The last two images show the surrounding background and its color distribution. The color distributions of the hand and the background are quite different and can be used as similarity measure for hand shape matching.

These form a hypothesis about the object pose. If the object pose, represented by the template, is actually found at the given position, the foreground and background color distributions must be different. If a different, or no shape is found there, the color distributions of foreground and background must overlap each other significantly. For illustration, Fig. 2 shows an example. Consequently, the dissimilarity between the two distributions can be used as a measure for template similarity.

Obviously, the color distribution has to be done as fast as possible. The Kullback-Leibler divergence is not feasible, because the computation of the histograms would take to long. A second disadvantage of a histogram-based representation is that there are possibly not enough color pixels to densely fill all histogram bins belonging to the inherent color distribution.

Representing the color distributions by a normal distribution does not have these disadvantages. We use one multivariate Gaussian to represent foreground and background, resp.. In our application to hand-tracking we observed that the approximation of the color distribution of the human hand by one Gaussian is sufficient in most cases. Of course, the background region should not be chosen too large, in order to obtain an appropriate approximation by one Gaussian.

Assume the means, μ_{fg}, μ_{bg}, and the covariances of the foreground and background regions, Σ_{fg}, Σ_{bg}, given in color space. Then we use the following color distribution similarity:

$$D = \frac{G(\mu_{bg}|\mu_{fg}, \Sigma_{fg}) + G(\mu_{fg}|\mu_{bg}, \Sigma_{bg})}{2} \tag{1}$$

with the unnormalized Gaussian function $G(\mathbf{x}|\mu, \Sigma) = |(2\pi)^3\Sigma)|^{1/2}\mathcal{N}(\mathbf{x}|\mu, \Sigma)$. Using the normal distribution itself in Eq. 1 would result in lower dissimilarity values for higher covariances while having the same separability of the distributions.

3.2 Fast Color Distribution Estimation

To reduce the computation time, we build upon the approach proposed by [26] for the template representation. They approximate each template silhouette by a set of axis-aligned rectangles. Then, they utilize that, using the integral image, the sum over the whole foreground and background region can be computed by $O(\#rectangles)$.

In the following, we will explain the computation of the mean and covariance, irrespective of the region being a foreground or background in the template. The same calculations are, of course, used to compute either of the two. Before we can explain the computation of mean and covariance matrix, we need to make some definitions: I is a color input image; II the integral image of I; $\mathcal{R} = \{R_i\}_{i=1\cdots n}$ a set of rectangles representing a template region and $II(R_i) = \sum_{\mathbf{x}\in R_i} I(\mathbf{x})$ the sum of all pixels over the rectangle R_i in I.

The mean μ can be trivially computed by:

$$\mu \propto \sum_{R_i\in\mathcal{R}} II(R_i) \tag{2}$$

The covariance matrix cannot be computed exactly using the rectangle representation because the off-diagonal entries cannot be computed using II. We could, of course, compute the integral image of I^2 with $I^2(\mathbf{p}) = I(\mathbf{p})I(\mathbf{p})^\mathsf{T}$. But this would need 6 additional integral images (6 and not 9, because $I(\mathbf{p})I(\mathbf{p})^\mathsf{T}$ is a symmetric matrix) and thus, result in too much memory accesses and a high latency per frame. Therefore, we have decided to estimate the covariance matrix in the following way.

We let each point inside a rectangle R_i be represented by the mean $\mu_i = II(R_i)/|R_i|$ of the rectangle. The covariance can now be estimated by

$$\Sigma \propto \sum_{\mathbf{x}\in\mathcal{R}} \mathbf{x}\mathbf{x}^\mathsf{T} - \mu\mu^\mathsf{T} = \sum_{R_i\in\mathcal{R}} \sum_{\mathbf{x}\in R_i} \mathbf{x}\mathbf{x}^\mathsf{T} - \mu\mu^\mathsf{T} \approx \sum_{R_i\in\mathcal{R}} |R_i|\cdot\mu_i\mu_i^\mathsf{T} - \mu\mu^\mathsf{T} \tag{3}$$

To avoid obtaining too crude an approximation of the covariance matrix, we subdivide big rectangles and use the mean values of the subregions to compute the covariance matrix. We have testes two alternatives. The first one is a simple subdivision of the rectangle into rectangular blocks of equal size. The second method is an adaptive subdivision: we subdivide a rectangle successively until the covariance estimated by all subrectangles does not significantly change anymore. We expected the second method to yield better results, but it has the disadvantage that the covariance matrix is fluctuating when slightly moving the template position in the input image. This disturbs the mode finding (i.e. detecting the most probable matching position in the input image) by our method described in Sec. 4. Hence, the simpler subdivision method could work better for us.

Please note that our algorithm could also take image segmentation results into account. More generally, we are not limited to any specific dimensionality of the input, i.e. we could easily incorporate other modalities such as depth values.

Algorithm 1. objectDetection(I, H, k)

```
Input: $H$ = template hierarchy, $I$ = input image, $k$ best hypothesis
Output: $M$ = $k$ best matches, each containing a target object position and pose
coarsely scan $I$ with root($H$), take $k$ best matches, $\rightarrow$ match candidates $C$
apply local optimization to each candidate $\in C \rightarrow$ new set $C$          // we use [27]
while $C$ not empty do
    foreach $c \in C$ do
        if c.template is leaf in $H$ then
            $M \cup \{c\} \rightarrow M$
        else                                        // note: c.template is a node in H
            $C_{new} \cup \{$ ( $c$.pos,templ) | templ $\in$ children of c.template $\} \rightarrow C_{new}$

    apply local optimization to $C_{new} \rightarrow C$
    $k$ best matches of $C \rightarrow C$
$k$ best matches of $M \rightarrow M$
```

4 Coarse-to-Fine and Hierarchical Object Detection

So far, we have discussed how to efficiently compute the probability that an object in a specific pose is observed at a position in the input image by a similarity measure between object template and input image. In this section, we will describe, how to use this similarity measure to detect the target object position, size, and pose.

To be able to do this in real-time, we save a large number of similarity computations by combining fast local optimization with a temlate hierarchy (which reduces the time complexity from $O(n)$ to $O(\log n)$) as follows.

We match the template that represents the root node in the template hierarchy with the input image with a large, template-dependent step size. For the k best matches in the image, we use a hill climbing method to find the local maximum in the likelihood map, i.e. the position in the input image, that matches best to the template. Next, we replace the root template by the templates of its child nodes and perform a local optimization again. Again, we keep the k best child nodes. We apply hill climbing again and so on, until we reach the leaves of the template hierarchy. Finally, the best match obtained by comparing a leaf template determines the final object pose and position in the input image.

We build on the approach proposed by [26] to construct the hierarchy with the following improvement: in each inner node, we only cover a region by rectangles, if it is foreground/background for *all*, and not only most, of the templates. This ensures, that only regions that correspond to foreground/background for all templates in the node are used to compute the fore-/background color distributions. In addition, to ensure that the intersecting fore-/background area at each inner node (excluding the intersection of all ancester nodes) is not empty, we have to allow a dynamic number of child nodes. Consider, for example, a template set that is split into two subsets. But the solely intersecting area of each subset still is an intersecting area for all of the templates. We are not able to distinguish

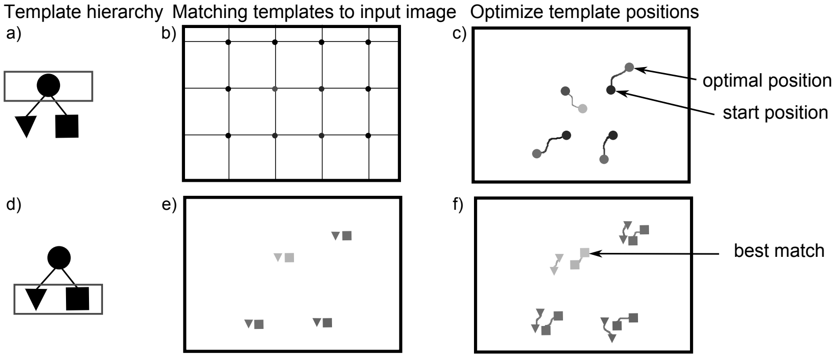

Fig. 3. Illustration of our coarse-to-fine and hierarchical detection approach. First, we coarsely match the root node of the template hierarchy (a) to the input image (b). For the best k matches (here 4), we perform a function optimization to find the best matching image position (c). Next, we use these matches as an estimate for the positions (e) for the child nodes in the template hierarchy (d) and search for the local maxima again. When arriving at the leaves of the hierarchy, we use the best match as final hand pose estimate (f).

it from the parent node because the intersecting areas are the same. In those cases, we split the template into three subsets or even more if necessary.

It remains to estimate the scan step size such that no local maximum in the confidence map is missed. This can be done offline, and depends only on the templates. Note that inner nodes in the hierarchy can be considered as templates, too, in that they describe the common properties of a set of templates. Therefore, the scan step size should be chosen not greater than the extend of the hill of a maximum in the likelihood map. This value can be determined by autocorrelating the template with itself. We do this not only in 2D image space, but also in scale space.

Algorithm 1 shows the above two approaches combined, and Figure 3 illustrates the method.

5 Results

We applied our approach to tracking of the human hand and evaluated it on several datasets. We captured video sequences with different hand movements and image backgrounds. Our approach is silhouette area-based and therefore best compared to segmentation based approaches. With the human hand, skin color has proven to work well in many cases. Consequently, we compared our approach with a skin segmentation-based approach.

We used three datasets, each with different skin segmentation quality. The first video sequence is a hand moving in front of a skin-colored background. Approaches based on skin color segmentation would completely fail under such conditions. The second dataset consists of a heterogeneous background. Several

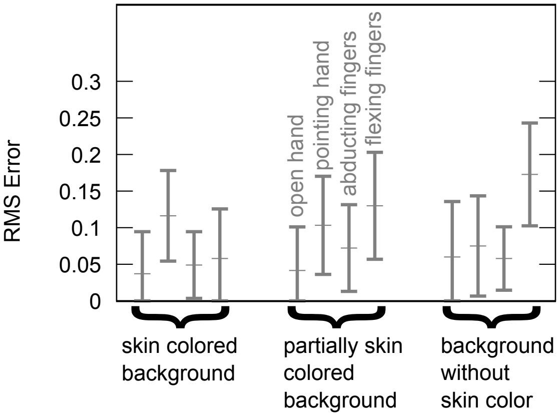

Fig. 4. Each group shows the results for a specific input dataset. Each bar within each group shows the mean and standard deviation of the RMS error between the brute-force and our coarse-to-fine detection is shown. An RMS error of 1 indicates the maximally possible error.

background regions of moderate size would be classified as skin, but in contrast to the first dataset, the hand silhouette often is clearly visible after skin segmentation. The third dataset contains almost no skin-colored background and, thus, is well suited for skin segmentation.

For each dataset, we tested four different hand movements, all of which include translation and rotation in the image plane. The four hand gestures are: an open hand, an open hand with additionally abducting the fingers, an open hand with additionally flexing the fingers, and a pointing hand. Overall, we have 12 different configurations.

Due to the lack of ground truth data, the quality is best evaluated by a human observer; computing an error measure (e.g. the RMS) between the results of two approaches does not make any sense. Therefore, we provide video sequences [1] taken under all above setups. In the video, the skin segmentation-based approach proposed by [26] and our approach are shown. Both approaches are tried with a brute-force dense sampling and our coarse-to-fine and hierarchical detection approach from Sec. 4. In the brute-force approach, we have chosen a scan step size of 12 pixels in x and y direction and 10 template scalings between 200 and 800 pixels. In all cases, the input image resolution is 1280 × 1024.

In the brute-force approach, the whole template hierarchy (if any) is traversed at each position separately. This needs several minutes per frame. To achieve an acceptable detection rate, one can stop traversing the hierarchy if the matching probability is lower than a threshold τ. The risk with thresholding is that all matches could be below the threshold and the hand is not detected at all. We

[1] http://www.youtube.com/watch?v=ZuyKcSqpkkE,
http://cg.in.tu-clausthal.de/research/handtracking/videos/mohr_isvc2011.avi

have chosen $\tau = 0.7$, which works well for our datasets. Note that our coarse-to-fine detection approach does not need any threshold and consequently does not have this disadvantage.

The brute-force approach will have, of course, slightly higher quality, but it will cost significantly more computation time, depending on the scan step size and the threshold τ. In order to examine the error of our coarse-to-fine hierarchical matching, we compared it to the brute-force dense sampling approach.

Using both our novel method and the brute-force method, we determined the best match for each image in the video sequence. For ease of comparison, hand positions, orientations, and finger angles were normalized. These will be called configurations in the following. Then, we computed the RMS error between the two configurations over the whole video sequence.

Figure 4 shows the results for each data set. Obviously, our method performs better in the "open hand" and "abducting finger" sequence. The reason is that "pointing hand" and "moving finger" templates have a smaller intersection area in the root node. This increases chances that the tree traversal finds "good" matches for nodes close to the root in image areas were there is no hand at all. Consequently, fewer match candidates remain for the true hand position during hierarchy traversal.

We also measured the average computation time for each datasets (10 frames per dataset). The computation time to detect and recognize the hand in the input image is about $3.5s$ for the brute-force approach. This, of course, is only achieved by using a manually optimized threshold τ, such that, for most positions in the input image, only the root node or a small part of the hierarchy is traversed. Our coarse-to-fine approach needs about $1.6s$ per frame (and no thresholding).

6 Conclusions

In this paper, we have presented a novel, robust detection and recognition approach for articulated objects. We propose a color distribution divergence-based similarity measure that does not need any error-prone feature extraction. Thus, our method can adapt much better to changing conditions, such as lighting, different skin color, etc.

We have also presented a coarse-to-fine and hierarchical object detection approach using function optimization methods and multi-hypothesis tracking to reduce computation time with only a small loss in accuracy. In addition, there are no thresholds that need to be adjusted to a given condition. And finally, it is straight-forward to incorporate other input modalities into our similarity and detection approach, such as range images or HDR images.

In an application to hand tracking, we achieve good results in difficult setups, where, for example, skin segmentation approaches will completely fail. Compared to the brute-force detection approach (that uses thresholding during template tree traversal to significantly prune the tree), our approach is about 2.2 times faster and about as reliable as the brute-force approach.

In the future, we want to extend the color distribution model of the object background for better handling multi-colored backgrounds (e.g. by Mixture of

Gaussians). Additionally we want to replace the local optimization by a multi-grid approach. We are also currently working on an implementation of our method on a massively parallel architecture (GPU) to reduce the computation time.

References

1. Barrow, H.G., Tenenbaum, J.M., Bolles, R.C., Wolf, H.C.: Parametric correspondence and chamfer matching: Two new techniques for image matching. In: International Joint Conference on Artificial Intelligence (1977)
2. Borgefors, G.: Hierarchical chamfer matching: A parametric edge matching algorithm. IEEE Transaction on Pattern Analysis and Machine Intelligence (1988)
3. Huttenlocher, D., Klanderman, G.A., Rucklidge, W.J.: Comparing images using the hausdorff distance. IEEE Transactions on Pattern Analysis and Machine Intelligence (1993)
4. Athitsos, V., Sclaroff, S.: 3d hand pose estimation by finding appearance-based matches in a large database of training views. In: IEEE Workshop on Cues in Communication (2001)
5. Athitsos, V., Sclaroff, S.: An appearance-based framework for 3d hand shape classification and camera viewpoint estimation. In: IEEE Conference on Automatic Face and Gesture Recognition (2002)
6. Athitsos, V., Alon, J., Sclaroff, S., Kollios, G.: Boostmap: A method for efficient approximate similarity rankings. In: IEEE Conference on Computer Vision and Pattern Recognition (2004)
7. Gavrila, D.M., Philomin, V.: Real-time object detection for smart vehicles. In: IEEE International Conference on Computer Vision (1999)
8. Sudderth, E.B., Mandel, M.I., Freeman, W.T., Willsky, A.S.: Visual hand tracking using nonparametric belief propagation. In: IEEE CVPR Workshop on Generative Model Based Vision, vol. 12, p. 189 (2004)
9. Kato, M., Chen, Y.W., Xu, G.: Articulated hand tracking by pca-ica approach. In: International Conference on Automatic Face and Gesture Recognition, pp. 329–334 (2006)
10. Toyama, K., Blake, A.: Probabilistic tracking with exemplars in a metric space. International Journal of Computer Vision (2002)
11. Lin, Z., Davis, L.S., Doermann, D., DeMenthon, D.: Hierarchical part-template matching for human detection and segmentation. In: IEEE International Conference on Computer Vision (2007)
12. Olson, C.F., Huttenlocher, D.P.: Automatic target recognition by matching oriented edge pixels. IEEE Transactions on Image Processing (1997)
13. Thayananthan, A., Navaratnam, R., Stenger, B., Torr, P., Cipolla, R.: Pose estimation and tracking using multivariate regression. Pattern Recognition Letters (2008)
14. Stenger, B., Thayananthan, A., Torr, P.H.S., Cipolla, R.: Hand pose estimation using hierarchical detection. In: International Workshop on Human-Computer Interaction (2004)
15. Shaknarovich, G., Viola, P., Darrell, T.: Fast pose estimation with parameter-sensitive hashing. In: IEEE International Conference on Computer Vision (2003)
16. Lin, J.Y., Wu, Y., Huang, T.S.: 3D model-based hand tracking using stochastic direct search method. In: International Conference on Automatic Face and Gesture Recognition, p. 693 (2004)

17. Wu, Y., Lin, J.Y., Huang, T.S.: Capturing natural hand articulation. In: International Conference on Computer Vision, vol. 2, pp. 426–432 (2001)
18. Ouhaddi, H., Horain, P.: 3D hand gesture tracking by model registration. In: Workshop on Synthetic-Natural Hybrid Coding and Three Dimensional Imaging, pp. 70–73 (1999)
19. Amai, A., Shimada, N., Shirai, Y.: 3-d hand posture recognition by training contour variation. In: IEEE Conference on Automatic Face and Gesture Recognition, pp. 895–900 (2004)
20. Shimada, N., Kimura, K., Shirai, Y.: Real-time 3-d hand posture estimation based on 2-d appearance retrieval using monocular camera. In: IEEE International Conference on Computer Vision, p. 23 (2001)
21. Zhou, H., Huang, T.: Okapi-chamfer matching for articulated object recognition. In: IEEE International Conference on Computer Vision, vol. 2, pp. 1026–1033 (2005)
22. Zhou, H., Huang, T.: Tracking articulated hand motion with eigen dynamics analysis. In: IEEE International Conference on Computer Vision, vol. 2, pp. 1102–1109 (2003)
23. Stenger, B., Thayananthan, A., Torr, P.H.S., Cipolla, R.: Model-based hand tracking using a hierarchical bayesian filter. IEEE Transactions on Pattern Analysis and Machine Intelligence 28, 1372–1384 (2006)
24. Stenger, B.D.R.: Model-based hand tracking using a hierarchical bayesian filter. Dissertation submitted to the University of Cambridge (2004)
25. Wang, R.Y., Popović, J.: Real-time hand-tracking with a color glove. ACM Transactions on Graphics 28 (2009)
26. Mohr, D., Zachmann, G.: Fast: Fast adaptive silhouette area based template matching. In: Proceedings of the British Machine Vision Conference, pp. 39.1– 39.12. BMVA Press (2010), doi:10.5244/C.24.39
27. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: Quasi-newton or variable metric methods in multidimensions. In: Numerical Recipes, The Art of Scientific Computing, pp. 521–526. Cambridge University Press, Cambridge (2007)