In:
"Virtual Realities", Lecture Notes in Computer Science Volume 8844, 2015, pp 108-129

Hand Pose Recognition — Overview and Current Research

Daniel Mohr and Gabriel Zachmann

University of Bremen,
{mohr,zach}@cs.uni-bremen.de

Abstract. Vision-based markerless hand tracking has many applications, for instance in virtual prototyping, navigation in virtual environments, tele- and robot-surgery and video games. It is a very challenging task, due to the real-time requirements, 26 degrees-of-freedom, high appearance variability, and frequent self-occlusions. Because of that, and because of the many desirable applications, it has received increasing attention in the computer vision community of the past years. A lot of approaches have been proposed to (partially) solve the problem, but no system has been presented yet that can solve the full-DOF hand pose estimation problem robustly in real-time.

The purpose of this article is to present an overview of the approaches that have been presented so far and where future research of hand tracking probably will go.

First, we will explain the challenges in more detail. Second, we will classify the approaches; third, we will describe the most important approaches, and finally we will show the future directions and give a short overview of our current work.

1 Motivation and Applications

The task of vision-based hand tracking is to estimate the human hand pose based on one or multiple cameras. The scientific interest in this task is very high, and the importance of hand tracking is larger than ever due to the increasing interest in natural user interfaces.

The applications for hand tracking are manifold. On the one hand, there are a lot of professional applications such as assembly simulation, motion capture, virtual prototyping, navigation in virtual environments, and rehabilitation. Hand tracking also has a high potential in medical applications, e.g. for sterile interaction with patient related data or tele-surgery. On the other hand, the interest in hand gesture driven game control is increasing strongly. For example, human motion tracking found its way to the consumer market through Nintendo Wii, Sony Move, and Microsoft Kinect. The goal of all three products is to track the human body. The Kinect is the first markerless vision-based consumer product. It is able to track the whole body with fairly high accuracy. The next consequent step is the precise tracking of the human hand, which can significantly improve

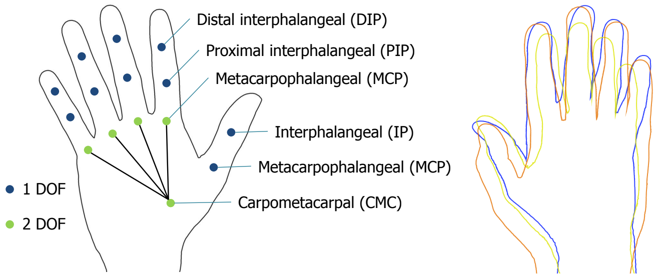
Distal interphalangeal (DIP)
Proximal interphalangeal (PIP)
Metacarpophalangeal (MCP)
Interphalangeal (IP)
Metacarpophalangeal (MCP)
Carpometacarpal (CMC)
● 1 DOF
● 2 DOF

Fig. 1. a) The left figure illustrates the degrees of freedom (DOF) of the human hand. The valid hand poses form a manifold in the 20-dim space.
b) The right figure illustrates five different hand shapes we captured from our group and some students. Any hand pose recognition approach has to take into account this shape variability.

the interaction with many game genres and desktop applications. The most recent application with strongly increasing interest in hand tracking are mobile devices to improve the natural interaction with them.

These are only a few of the numerous applications for hand tracking. Obviously, most of them need the hand to be tracked precisely and in real-time. Thus, algorithms to achieve this are an enabling technology. But robust hand detection and recognition in uncontrolled environments is still a challenging task in computer vision, especially on mobile devices due to their limited hardware resources.

1.1 Challenges of Hand Tracking

The main challenges of camera-based hand tracking are the high-dimensional hand configuration space, the high appearance variation, the limitations of cameras, and the potentially disturbing environment. In the following, the challenges are described in detail.

High-dimensional Configuration Space The problem dimension to estimate the full-DOF hand pose is very high. Figure 1a illustrates the articulations. Each finger has 4 degrees of freedom (DOF) which yield in 20 local DOF for the hand pose. Often, an additional DOF for axial rotation is modeled the thumb. With the 6 DOF for the global position and orientation the problem space has 26 dimensions.

Hand Motion and Appearance Variation The human hand to be tracked varies strongly from person to person. The skin color for example depends on the ethnic origins and the skin browning. The geometry of the hands are also very different, e.g. thickness and length of the fingers, and width of the hand to mention only

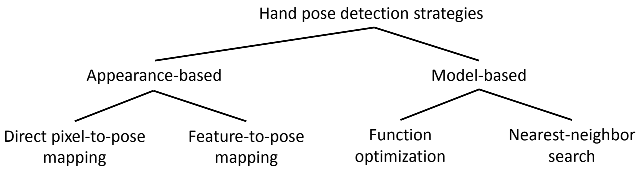

Fig. 2. Hand tracking approaches can be classified into appearance and model-based approaches. Appearance-based approaches use direct mapping techniques. In contrast, model-based approaches fit a hand model to the input image to estimate the pose. Pose estimation can be formulated as function optimization (the hand pose is the parameter set to be optimized) or nearest-neighbor search (find the hand pose most similar to the observed hand).

some of the varying parameters. Even the kinematic can vary slightly between human beings.

Additionally, the appearance variability of the hand is very high, and thus, it is challenging to detect the hand in an input image because neither its appearance nor its position are known in advance.

Unconstrained Background To be able to detect the hand in an input image, one first has to identify the image region corresponding to the hand by applying a segmentation algorithm (e.g. skin color segmentation or background subtraction) or extract features whose distribution on the hand and the background are sufficiently different (e.g. edges). The more complex the background the less likely those features can be used to discriminate between hand and background. For example skin colored regions in the background will heavily disturb a skin color segmentation. Moving object in the background are an error source for background subtraction and textured regions (consider for example a keyboard) will produce a lot of edges in the background that heavily disturbs any edge-based matching.

Camera Limitations Current camera technology is limited in its capturing capability. In most real setups there are over- and/or underexposed regions due to the low dynamic range of the cameras. Even HDR-cameras have a by some orders of magnitude lower dynamic range than the human eye has. Furthermore, most cameras capture only the usual three color channels and not the whole spectrum of light.

Real-time Tracking Condition Most hand tracking applications need the hand to be tracked in real-time i.e. at least 25 full pose estimations per second. This is a very strong condition in particular due to the high dimensional search space. This condition is particularly challenging for tracking on mobile devices.

2 Classification and Overview of Approaches Up to Now

Due to the aforementioned challenges, hand tracking is a very interesting and active research area. A lot of approaches have been presented in the past, using different algorithms, ranging from neural networks over hashing to hand pose hierarchies. The motivation of this article is to give an overview and classification of the various approaches. We hope to help both new researchers in this area, who want to get familiar with hand tracking, and advanced researchers, who want to get a different point of view on the research area.

In the following, we will first classify the approaches and then explain many approaches in more detail.

There are several ways to categorize hand tracking approaches [1, 2] e.g. gesture of full-DOF pose recognition, approaches that need automatic or manual initialization and so forth. The most popular categorization is: appearance-based vs model-based (Fig. 2). The term model-based means that a 3D hand model is fitted somehow against the input image. Model-based approaches can either be formulated as optimization or nearest neighbor search. The idea behind the optimization is simple: based on a initial match, the model is adapted and fitted again until convergence. The nearest neighbor formulation considers a database with all possible hand poses, which have to be tracked. Then, the goal is to find the most similar hand pose and the corresponding position in the input image.

By contrast, appearance-based approaches try to learn a direct mapping from the input image to the hand pose space. Most of them use fairly low-level features (e.g. edges or color blobs) or even no features at all (e.g. artificial neural networks). Thus, such approaches do not need to search the whole configuration space because the information of the hand poses is encoded in the learned mapping. This typically makes them computationally less expensive. On the other hand, they suffer from accuracy and stability due to poor handling of noise and partial occlusion in the input image. Of course, appearance-based approaches need to contain the information about the hand model in some way, too. For example, in a neural network-based approach, which maps the image pixels to the pose, a hand model is implicitly stored in the neural network itself. Figure 3 visually compares the idea of model- and appearance-based approaches.

Appearance-based Approaches A typical appearance-based approach is used in [3, 4] to detect the hand position in a gray-scale image. In a training step, multiple hand poses are trained. During tracking, “attention images” are used for segmentation. Basically, the image pixels are directly used as input vector and a principal component analysis (PCA) is applied for dimension reduction. A hand pose is successfully segmented by validating a training image to be close enough in the low-dimensional space. Nearest neighbor search is performed using a Voronoi diagram. The hand segmentation probability is evaluated using kernel density estimation.

A set of specialized mappings is trained based on data obtained by a Cyberglove in [5]. After a skin segmentation, moment-based features are computed

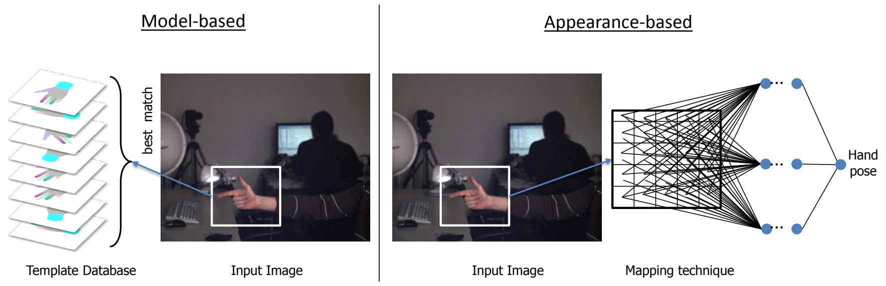

Fig. 3. Model-based approaches (left) use an object model (here the human hand) and match the templates, each representing a hand pose, to the input image. In contrast, appearance-based approaches (right) try to learn a direct mapping from the image space to the pose space.

and used as weak mapping functions. This mapping functions are combined to get a strong classification function.

Another classical appearance-based approach for hand tracking is used in [6]. They used a so-called Eigentracker to be able to detect a maximum of two hands. Color and motion cues are used for initialization. The eigenspace is updated online to incorporate new viewpoints. Illumination variations are handled by a neural network.

In [7] skin-colored blobs are detected to localize the hand position. Next, the hand pose is estimated by detecting the finger tips. The blobs are detected using a Bayesian classifier. Color changes during time are handled by an iterative training algorithm.

[8] detect the hand position in the image using Camshift. A contour in Fourier space is computed to obtain a scale and rotation invariant hand descriptor. After locating the hand position, the finger tips are determined by a semicircle detector. Particle filtering is used to find finger tip location candidates. A k-means clustering is applied to the candidates. The cluster centers (prototypes) are used as the final finger positions.

One of the main disadvantages of appearance-based approaches is their high sensitiveness to noise, feature extraction errors, and partial occlusion. For example, if a finger tip is occluded, but not necessarily the remainder of the finger, the above approaches will fail to detect the finger. It is not even easy to determine which fingers are occluded. A promising alternative are model-based approaches.

Model-based Approaches Model-based approaches search in the large configuration space to find the best matching hypothesis. Basically, a descriptor, optimized for fast and accurate matching, is defined first. Then for all hand poses to be tracked, the corresponding *template* is generated. During tracking, the hand poses are compared to the input image by computing the similarity between the corresponding templates and the (preprocessed) input image.

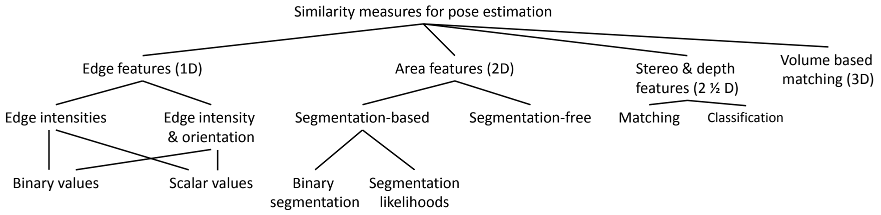

Fig. 4. Similarity measures for hand pose estimation can be categorized into four different types. The categorization is based on the input modality used for matching. Most often used are edges, color (segmentation) and depth images. Edge and color segmentation likelihood can either be used directly for matching or binarized before matching. Area-based features also allows to use the shape hypothesis directly without applying a segmentation. Depth image can be used for segmentation or direct matching.

Depending on the needs of the approach the templates are precomputed or generated online during tracking. The main differences between the approaches is the method to compute the similarity between hypothesis and input image, how to compute each similarity evaluation as fast as possible, and acceleration data structures to avoid as many similarity measure evaluations as possible.

The advantage of model-based approaches compared to appearance-based approaches is that arbitrary hand poses can be modeled including self occlusion. Partial occlusion by other objects can be handled robustly as well because the similarity measure between a hypothesis and an input image is only affected by a limited amount.

In the next section, we will first provide an overview of the different categories of similarity measures. In the section following that we will categorize and describe the acceleration data structures.

2.1 Similarity Measures

In the area of hand tracking the most often used features in the past are skin color and edges. Edges and silhouette area (e.g. extracted using skin color) are complementary features and often combined into a single measure using weighting functions. In the recent years, depth cameras (e.g. Kinect, Mesa SR 4000) became available and popular. The hand (and other objects) can more easily be distinguished from other objects. Depth images, in some way, provide area information (continuously changing depth values) and edge information (strongly changing depth values). Thus, depth images became very popular in the area of hand tracking in the most recent years. Volume reconstruction-based approaches [9–12] directly work on a 3D volume, but these approaches need a lot of cameras and get a very coarse volumetric representation of the hand. Only a few approaches exist using volume reconstruction. Due to space limitations, we will not present any such approaches in detail. Fig. 4 gives an overview of the different similarity measure classes.

We will first present approaches using classical area based similarity measures based on color images, then approaches using depth images, and finally, depth image based approaches.

Silhouette Area-Based Similarity Measures Silhouette-area based similarity measures are very effective and fast for articulated object tracking. The measure is continuous with respect to changes in pose space and robust to noise. Basically, the segmented silhouette of the input image is compared to a hypothesis (also represented by its silhouette area). The more similar both silhouettes are, the higher the matching probability is.

Silhouette area-based approaches can be divided into two categories. The first category needs a binary silhouette of both the model and the query image. The second category compares the binary model silhouette area with the likelihood map of the query image. To our knowledge there are no approaches using a non-binary model silhouette. All approaches presented use a fixed hand model and the hand model contains no noise, thus there is no information gain using a non-binary representation.

A simple method belonging to the first class is proposed in [13]. They assume that the hand is in front of a homogeneous, uniformly colored background. First, they apply skin segmentation to extract the foreground. Based on the segmented region, hand size, center, and differences between particular pixels on the boundary are used to detect the hand position. This information is used to recognize some simple gestures, e.g. an open hand or a fist.

A more robust approach is proposed in [14] and [15]. First, the difference d between the model silhouette and the segmented foreground area in the query image is computed. Then, the exponential of the negative squared difference is used as silhouette matching probability P i.e. $P = \exp(-d^2)$. A slightly different measure is used by Kato et al. [16]. First, they define the model silhouette area A_M, the segmented area A_I and the intersecting area $A_O = A_I \cap A_M$. The differences $A_I - A_O$ and $A_M - A_O$ are integrated into the overall measure in the same way as described above.

In [17], the non-overlapping area of the model and the segmented silhouettes are integrated into classical optimization methods, e.g. Levenberg-Marquardt or downhill simplex. [18] first compute the distance transform of both the input image and the model silhouettes. Regarding the distance transformed images as vectors, they compute the normalized scalar product of these vectors. Additionally, the model is divided into meaningful parts. Next, for each part, the area overlap between the part and the segmented input image is computed. Then, a weighted sum of the quotient between this overlap and the area of the corresponding model part is computed. The final similarity is the sum of the scalar product and the weighted sum.

All the aforementioned approaches have the same drawback: to ensure that the algorithms work, a binary segmentation of the input image of high quality is a pre-requisite. Binarization thresholds are often difficult to determine, and even an optimal threshold often yields a loss of important information about pixel-belonging-to-hand probabilities. To overcome this problem, approaches have been presented that work directly on the segmentation likelihood map. In [19] 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. However, in many cases, this leads to a very noisy edge image.

In [20–22], the skin color likelihood map is directly compared to the hand silhouette. Given a hypothesis, the silhouette foreground area of the corresponding hand pose and the neighboring rectangular background of a given size are used to compute the similarity measure. In the skin likelihood map, the joint matching probability for foreground and background are computed and combined into one similarity measure. Stenger et al. [20, 21] proposed a method to compute the joint probability in linear time w.r.t. the contour length.

[23] further reduced the computational complexity to compute the joint probability as similarity measure proposed by [20, 21] from linear to near-constant time. Consequently the computation of a similarity is resolution-independent. For this purpose they used the integral image and a novel representation of silhouette-areas based on axis-aligned rectangles.

Segmentation-based approaches have two main drawbacks. The first one is the segmentation itself: it is an error-prone step because the segmentation is based on an assumption about the color distribution of the foreground. The second drawback of segmentation-based approaches is the silhouette area, which is a projection of the 3D hand to 2D and, thus, a lot of important information about the hand shape is lost.

[24] proposed a novel similarity measure that does not need any kind of segmentation at all. The idea is to compute the color distributions in the input image that correspond to the shape of the hand and the corresponding background described by the template. They used the rectangle-based template representation from [23] to be able to compute the similarity measure efficiently. A further advantage of this similarity measure is that it trivially can be extended from color images to other input modalities such as range images.

The most important disadvantage of area-based approaches in general, and using a monocular camera in particular, is that several hand poses can be hardly distinguished. The reason is that the silhouettes are too similar from a specific point of view, i.e., a silhouette-based representation introduces a lot of ambiguities. Such cases are, for example, fingers in front of the palm with a moderate flexion, as shown in Figure 5.

Edge-Based Similarity Measures Edge gradient features are complementary to silhouette area-based features. While the silhouettes information utilizes the hand foreground and background, the idea of edge features is the border between fore- and background and even more important the separation of the fingers from the palm. The idea is to disambiguate hand poses that are unable to be distinguished using the silhouette. A further advantage of edge features is that they are fairly robust against illumination changes and varying object color.

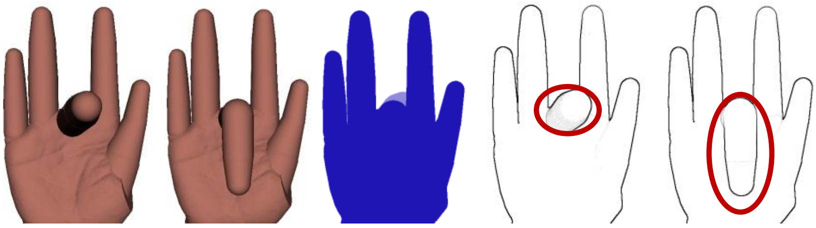

Fig. 5. Silhouette area-based similarity measures cannot resolve all ambiguities. Two example hand poses (left) illustrate the problem. Using the silhouette-area as feature does not allow to distinguish between both poses because the difference area (middle image, light blue area) is by far to small. In contrast, edge features allows us to distinguish both poses because the edges significantly change (right, highlighted by a red ellipsoid) between the two poses.

However, edges are not completely independent of illumination, color, texture, and camera parameters. Therefore, smart algorithms are still essential.

Most of the edge-based approaches need binary edges, i.e. an edge extraction is applied to both a projection of the hand model and the input image. Next, a distance measure between the edges is defined to compute the similarity between a hypothesis and the input image. As distance measures, for example the Hausdorff distance can be used. But much more popular is the Chamfer distance [25], [26] $\mathcal{C}$, which is a modification of the Hausdorff distance. Chamfer matching for tracking of articulated objects is, for example, used by [20], [27], [28], [29], [30], [22], [16] and [31]. A disadvantage of the chamfer distance is its sensitivity to outliers.

Both, chamfer and Hausdorff distance can be modified to take edge orientation into account, albeit with limited accuracy. One way to do this is to split the template and query images into several separate images, each containing only edge pixels within a predefined orientation interval [32], [20]. To achieve some robustness against outliers, [20] additionally limited the nearest neighbor distance from template to image edges to a predefined upper bound. A disadvantage of these approaches is, of course, the discretization of the edge orientations, which can cause wrong edge distance estimations.

[33] integrated edge orientation into the Hausdorff distance. They modeled each pixel as a 3D-vector. The first two components contain the pixel coordinates, the third component the edge orientation. The maximum norm is used to calculate the pixel-to-pixel distance. [22] presented a similar approach to incorporate edge position and orientation into chamfer distances.

Edge orientation information is also used by [34] as a distance measure between templates. They discretized the orientation into four intervals and then generated an orientation histogram. Because they do not take the edge intensity into account, the weight of edge orientations resulting from noise is equal to that of object edges, which results in a very noise sensitive algorithm.

In [35] the templates are stored as a set of line segments, each line contains information of its position, orientation, and length. In the input image, the line extraction thresholds are set such that most lines belonging to the target object are found. This results in very low thresholds, which has the disadvantage that many edges caused by noise are extracted, too. Consequently, the image becomes highly cluttered. Matching is formulated as finding the best correspondences between template and input lines. Because a large number of edges, produced by noise, are processed in the line matching step, the probability of false matching is highly dependent on the input image quality and background.

[36] avoided thesholding or discretization of the edge gradient. For this purpose they first mapped the edge gradients such that they can directly be compared using scalar product by keeping invariance with respect to the sign of the gradient. The similarity measure can be formulated as convolution and its computation time further reduced using Fourier transform. The similarity measure is designed to behave robust in noisy regions by integrating the edge gradient of the neighborhood of the template edges. But, as most other approaches, they still have to cope with varying edge intensities of important object edges.

The general problem with edge-based approaches is that they depend on the edge detection operators quality. There always is a trade-off between high clutter and missing object edges, i.e. often, the hand pose cannot be uniquely identified or not detected at all. Thus, if the input images have a cluttered background, edges is not the best choice. An additional problem are edge responses of wrinkles on the hand itself. They are hard to be modeled correctly by the artificial hand model, typically used for matching. Thus, they have to be treated as noise, and disturb edge-based similarity measures.

Depth images do not have the problems of false positive edges due to hand wrinkles or texture. Additionally, they do not need any color segmentation, which often is error prone. In the following, we will present approaches using depth images.

Depth Images Based Similarity Measures [37] used projective geometry to match the hand template to the depth image. First, a part of the 26-dimensional configuration space was sampled and a dimension reduction using PCA performed. A particle filter in the low-dimensional space was used to find the hand pose in the next frame. Their distance measure between the hand hypothesis and the input image uses both image coordinates and depth information.

In [38], the hand pose is estimated using a ToF camera. The depth information was primarily used to segment the hand from the background. Features like finger tips, finger-likeness and palm candidates are extracted. A graph is built based on the features and the candidates/nodes best meeting some specific conditions are considered as finger tips and palm. Additionally, the knowledge of the palm pose in the previous frame is taken into account. The approach is able to detect two hands simultaneously.

[39] integrated the Kinect into their hand tracking algorithm. The hand is localized conventionally through skin segmentation. Hand pose estimation is for-

mulated as an optimization problem. The difference-of-the-depth-values between the hypotheses and the Kinect data are added to the objective function. They use Particle Swarm Optimization (PSO) as optimization function. In [40] they extended the approach to track two interacting hands.

A tracking by classification approach is proposed by [41]. They adapt the method of human body tracking [42] to hand tracking. In [42], the body is partitioned into 31 parts. Then, they train a decision forest to be able to classify each part. They use a simple but effective difference-of-two-depth-values classifier for each node in the trees. The feature is inspired by [43]. After classification of each position in the depth image, they compute the body part positions by the mean shift algorithm. [41] argue that for hand poses the random forests will become too large. To overcome this problem, they subdivided the hand pose estimation into two sub-problems. First, random forests for several hand poses are trained. Second, for each hand pose individual random forests for the finger parts are learned. Matching is performed by first classifying the hand pose and then detecting the finger poses for the most probable hand pose.

With the great advances in range cameras over the past years, depth images based matching seems to facilitate the most promising avenues for future research. One advantage is that the depth can be used to compute the size of the hand in image space. Additionally, a partial volume representation of the hand can be computed, which is a very useful information a color image does not provide. Furthermore, current depth cameras have their own NIR light source, and thus, are less dependent from the environmental lighting conditions. This yields a much more stable image, except in direct sunlight where depth cameras always fail due to the high amount of NIR light the sun emits. One drawback of depth images is that they cannot differentiate between a real hand (skin colored) and an artificial hand (e.g. made of plastic). But for practical use this is rarely relevant.

Similarity measures, in general, are often expensive and have to be computed very often for each frame due to the large hand shape variability (which yields a large number of templates). Thus, acceleration strategies are essential to achieve real-time hand pose estimation.

2.2 Fast Template Search Strategies

So far, we have discussed similarity measures for efficient hypothesis testing using template matching. The similarity measure, basically, is responsible for the quality of the pose estimation. For full-DOF hand tracking application, a huge number of templates have to be matched. To be able to perform hand pose estimation and tracking in real-time, one has to avoid as many similarity measure computations as possible to save computation time. For this purpose, several acceleration data structures for template matching have been proposed, which will be described in the following section.

Many approaches avoid the problem of simultaneous object detection and pose estimation by a manual initialization, or they assume a perfect image segmentation. Manual initialization, however, means that the approach needs to know the object location and pose from the previous frame.

In [44], an approach is proposed that needs both, manual initialization and a perfect segmentation. They convert the hand silhouette into a descriptor, which is used to compare the query silhouette against the database. Local PCA is applied to further reduce the dimension of the descriptor. To avoid an exhaustive search, they assume an initial guess and search for the best match in the low-dimensional neighborhood.

Manual initialization is also needed in [22]. They use nonparametric belief propagation, which is able to reduce the dimension of the posterior distribution over hand configurations. They integrate edge and color likelihood features into the similarity measure, and consequently, they do not need the hand to be perfectly segmented.

Similar preconditions are needed in [45][46]. The similarity measure is integrated into an objective function, which is then optimized by gradient descent methods. Hand texture and shading informations are used in [46] and skin color in [45].

[14] uses a two-stage Nelder-Mead (NM) simplex search to optimize the hand position. They sample the hand pose space using a CyberGlove. The first NM search is constrained to the samples to avoid getting invalid hand poses. The second NM stage is a refinement and performs an unconstrained search in the continuous configuration space. They employ edge and silhouette features to measure the likelihood of the hypothesis.

[47] proposed a hand tracking approach that is designed to handle interactions with simple objects like cylinders and spheres. They manually initialize the hand pose and then optimize the objective function using the particle swarm optimization (PSO) algorithm. The objective function consists of two parts. The first part contains the incremental fitting of the hand model to the input image. This is done using the chamfer distance between binary edges, and the overlapping area between the hand silhouette and the binary segmentation. The second part penalizes self-penetration of the hand and penetration of the hand with the object the hand is interacting with.

Often, in real applications, neither a perfect segmentation nor an initial pose is given. A manual initialization is always tedious or not possible at all. Thus, several approaches are developed to search in the whole configuration space to be able to estimate the object pose in (near) real-time. This is even more challenging if the position of the object has to be detected as well. Particularly for objects with a high shape variability such as the human hand, localization and detection cannot be done separately because neither the appearance nor the location is known in advance.

Hashing [48, 49] is also used by [50] for hand pose classification. Binary hash functions are built from pairs of training examples, each pair building a line in the pose space. The hash values are in $\{0, 1\}$ depending on whether the projection of the input to the line is between two predefined thresholds or not. The projection is computed using only distances between objects (e.g. hand pose images). The binary hash functions are used to construct multiple multibit hash tables.

The idea to convert the evaluation of similarity measures to vector distances is used in [29, 51]. They used a Euclidean embedding technique to accelerate the template database indexing. A large number of 1D embedding is generated. An 1D embedding is characterized by a template pair. AdaBoost is used to combine many 1D embeddings into a multidimensional embedding. A database retrieval is performed by embedding the query image, and then, comparing the vector in the embedded Euclidean space to all database elements. Each embedding needs the similarity computation between the input image and all pairs of templates characterizing the high-dimensional embedding.

Thayananthan et al. [32] used a relevance vector machine (RVM). They used skin segmentation to localize the hand. The RVM's are trained using an EM type algorithm to learn the one-to-many mapping from binary image edges to pose space. From a training set of 10.000 hand templates, 455 are retained.

[52] used a hierarchical approach for hand gesture tracking with application to finger spelling. They use a small database consisting of real hand images. The hand silhouette is extracted utilizing skin segmentation. Applying a Fourier Transform to the silhouette, they obtain a high-dimensional feature vector. They build a hierarchy by recursively applying PCA-based vector quantization to the vectors.

[20] proposed an approach that hierarchically partitions the hand pose space. "The state space is partitioned using a multi-resolution grid". The nodes at each level are associated with non-overlapping sets of hand poses in the state space. "Tracking is formulated as a Bayesian inference problem". During tracking, they process only the sub-trees yielding a high posterior probability.

In contrast to the pose space hierarchy of [20], [23] used a feature space hierarchy to be able to build a deeper template tree, which allows for faster matching. Their hierarchy is based on the silhouette area of the templates. Inner nodes represent the intersection area of their child nodes. Leaves represent hand poses and inner nodes represent the hand poses of all leaves in the sub-tree. Matching is performed through traversal. During the traversal of the tree from the root node to a leaf, the hand silhouettes are getting closer to a hand pose.

In [28], cascading [53] is used for hand shape classification. Four different classifiers are employed, based on edge locations, edge orientations, finger locations, and geometric moments. "Database retrieval is done hierarchically by quickly rejecting the vast majority of all database views" using finger and moment-based features. They reported that they could reject 99% of the database in this step. Then, the remaining candidates are ranked by a combination of all four classifiers.

2.3 Conclusions

In this section, we provided an overview on the area of hand pose estimation. A detailed description about the main challenges of vision-based hand pose estimation is given. Clearly, in the past decade a lot of approaches have been

Family of approach	Initialization method	Reliability of results
Function optimization [14, 47, 45]	manual	high
Dim reduction and NNS [44, 22]	manual	medium–high
Hashing [50]	automatic	medium
Hierarchies: pose [20] and image space [52, 23]	automatic/manual	med
Machine learning [29, 51, 32]	automatic	med–high
Cascading [28]	automatic	med–high

Fig. 6. Hand pose estimation approaches can be categorized in the above six families. Some families are able to initially detect the hand pose and position themselves, others are not. The reliability means how often the estimated pose of the approach is (close to) the true hand position. For all approaches, any similarity measure could be used.

presented that tried to solve the problem. Many approaches make an important contribution to robust real-time hand tracking. Several similarity measures and input modalities can, of course, be combined to increase the robustness, e.g. edge features, color-based features, and depth information.

In the following section, a new direction for tackling the problem of hand tracking is proposed. We believe it to be very promising, and our preliminary results prove its great potential. The new approach uses depth images for similarity measure computation and machine learning to learn the hand poses as well as invariance to hand geometry e.g. finger length. The approach explained in the following is our current research in progress. Similar methods for human pose estimation exist but its application to hand pose estimation is challenging due to the high self-similarity of the fingers and much more unconstrained hand movement. We will start with the motivation for choosing machine learning for hand pose recognition.

3 Our RF–Based Hand Pose Recognition

The survey given in the previous section shows that the most often used and promising approaches are model-based. Almost all model-based approaches use similarity measures defined by experts to compare a pose hypothesis with the observation, i.e., an expert manually designs the kinds of features such as hand silhouette or edges to be used and how exactly the hypothesis is compared against the observation (e.g., non-overlapping area or chamfer distance). This has two drawbacks: first, manually defined kinds of features and similarity measures between features may not necessarily be optimal. They cannot be learned from examples. Second, similarity measures from experts use a fixed (mostly artificial) hand model. But, the more the real hand to be tracked differs in shape from the artificial hand model, the more inaccurate the pose estimation will be. Coping with the variability of hand geometry is often hard or impossible with such similarity measures. Of course, one could add them as additional hand poses to the template database, but they would all have to be matched to the input image, which would dramatically increase the execution time of the recognition system, and they would dramatically increase the memory needed for storing the template data base.

The approach introduced in this section uses a machine learning approach based on random forests[1] (RF) that can learn to be invariant to different hand geometries as well as achieve a high pose recognition rate. Hand pose estimation, which is a regression problem, is mapped to a classification problem in a natural way. This enables to use the well-proven random forest techniques for classification.

3.1 Random Forests

This section gives a quick recap of the general idea of random forests. A random forest is a set of decision trees. Decision trees are a common technique to make any kind of decision based on a set of individual test functions called weak classifier. Each weak classifier yields a small amount of information gain. One of the main problems of decision trees is that they tend to overfit to the training data, and consequently, do not generalize well. To get rid of this problem, Leo Breiman [54] proposed to use a set of decision trees. Each individual tree is trained using for each tree a random subset of both, the training data and weak classifiers. The idea behind random forest is that some individual decision trees can make a wrong decision but the majority of the trees will make the right decision. Random forests have been widely used and proved to outperform many other machine learning approaches and generalize very well. Note that RF's can also be used to solve other tasks e.g. regression or density function estimation.

3.2 Learning Random Forests for Hand Pose Recognition

The RF–based approach is used for hand pose recognition in terms of estimating the hand orientation (3 degrees of freedom) and joint angles (20 degrees of freedom) with a high accuracy. This yields a 23 dimensional search space. The remaining 3 DOFs are the location in the image. The approach most similar to our method is [41]. Their approach depends on a clustering algorithm that assigns each hand pose to a gesture, and the number of gestures they use. The RF–based approach proposed in this section uses a more natural way, that directly maps the hand pose estimation to classification without the needs of additional error prone methods.

Description of the Kind of Input Dataset: The input data for our hand pose recognition method are depth images obtained using a time-of-flight camera. Depth images provide the distance from the camera for each pixel. Depth images are superior over color images because they are independent of color distortions,

[1] Random forests (sometimes also called decision forests, were first introduced by Leo Breiman [54]. Precursors were introduced, e.g., by [55][56].

lighting conditions and the depth information itself can solve a lot of ambiguities a color image is not able to.

Our Features Set: Crucial for the performance of any RF is the automatic choice of good features and the per-node weak classifiers, respectively. The choice of the features is specific to the task, which, here, is hand pose recognition from depth images.

Choice of the Features: Given an input image I and a candidate position $\mathbf{x}$ the hand is supposed to be located at, [42, 41] proposed differences between depth values of randomly selected pixels as features. This approach can be generalized to arbitrary rectangles R_i. Rectangles are a good choice because the sum of a rectangular area can be computed in constant time utilizing the integral image [53]: $f(I, \mathbf{x}) = d_I\left(\mathbf{x} + \frac{\mathbf{R}_i}{d_I(\mathbf{x})}\right) - d_I\left(\mathbf{x} + \frac{\mathbf{R}_j}{d_I(\mathbf{x})}\right)$. The notation $\frac{\mathbf{R}_i}{d}$ denotes a scaling of rectangle $\mathbf{R}_i$'s position and size relative to the hand distance, and $d_I(\mathbf{R}_i)$ denotes the mean of all distance values in $\mathbf{R}_i$. The choice of the features have two main advantages. First, range images from state-of-the-art hardware are very noisy. Using rectangles trivially allows for an averaging of the depth values. It could be much more robust than single pixel. Second, using a rectangle can provide more information to a weak classifier than a single pixel. The integral image as acceleration data structure allows constant time complexity for the computation time of the feature response.

Background Handling: Similar to [42], we use a large value ω for $d_I(\mathbf{y})$ if $\mathbf{y}$ belongs to the background. To determine whether a pixel $\mathbf{y}$ belongs to the background, we apply the threshold test $d_I(\mathbf{y}) - d_I(\mathbf{x}) \notin [\tau_m(\mathbf{x}), \tau_M(\mathbf{x})]$. The RF–based scheme uses more sophisticated, adaptive thresholds. During RF training, in each decision node depth images are used for training. $\tau_m(\mathbf{x}), \tau_M(\mathbf{x})$ is computed by recording the minimum/maximum distances of all pixels inside all hands relative to the average hand distance. Additionally, a small offset ε is added to the thresholds to increase the background detection robustness during tracking.

To ensure that our features are independent with respect to the overall depth of the hand i.e. $d_I(\mathbf{x})$, in the case where $\mathbf{y}$ is a background pixel, the constant ω $d_I(\mathbf{x})$ is added to the background .

Infinite Training Dataset Generation: An artificial hand model is used to train the RF. This is very advantageous because we are able to draw samples from an *arbitrarily large* ground truth database of hand poses for any pose and shape variability. This yields a very flexible and virtually inexhaustible source of ground truth data for RF training.

Of course, any approach should be highly robust against different hand geometries. In order to achieve that, the length and thickness of each bone has to be explicitly be parametrized in the hand model. Additionally, Perlin gradient noise is added to the hand geometry to model "curvy" hand silhouettes produced by skin, tissue, phalanx "imperfections", and camera noise. This allows to generate images with various shapes for our training set. In this way, the RF learns various hand geometries for each and every pose.

We denote a hand pose in pose space by $\theta \in \Theta$ and a specific hand geometry by $\gamma \in \Gamma$. The hand geometry space consists of parameters for the finger, palm, and forearm length and thickness, as well as the Perlin noise parameters to modify the bending of the hand. Overall, the rendered depth image $I_k(\theta, \gamma)$ depends on those two properties. In the following, we will explain how we use our artificial hand model to train the decision trees.

Mapping Hand Pose Recognition to RF Learning: The RF–based scheme uses an elegant mapping of hand pose recognition, which is a regression problem, to a random forest classification problem. This has the advantage that the well proven Shannon entropy-based information gain measure can be used. For a robust estimation, it is crucial to recognize different hand poses independently of the hand geometry (e.g., finger length and thickness) to some amount. Therefore, the mapping to a classification problem is as follows: random samples of uniform distribution are taken from the hand pose space Θ; each sample $\theta_i \in \Theta$ represents a class c_i our RF should recognize. Each hand pose sample θ_i is rendered at a different geometry $\gamma_j \in \Gamma$ taken randomly from uniform distribution, too; all of them are put in to the same class.

That way, a random forest for classification with the Shannon Entropy based information gain is trained. Consequently, the RF–based scheme learns to classify depth images of hand poses robustly against hand shapes.

Recognizing Hand Poses with RFs: Arriving at a leaf node of a decision tree, the predicted class labeled with a hand pose is obtained In case of discrete classes, typically a voting over all trees is done and the class with the most votes wins. However, the set of poses estimated by the decision trees in our RF can be considered as density function because the hand pose space is a continuous space. For this reason, the mode of the density function is used as the final estimated pose. Due to the high-dimensional pose space, the mode finding is applied to each degree of freedom separately. For maximum finding, the well known mean shift algorithm [57] is used.

For qualitative evaluation, a real hand with flexing and abducting 4 fingers (index, middle, ring, and pinky) is captured. For the real dataset no ground truth is available thus, we provide screenshots of a few selected frames that shows the power of our approach in Fig. 7.

4 Conclusions

In the near future it might be a good idea to use multi-modal sensors and features, such as depth and conventional camera images, most often denoted by RGB-D image. The depth image can be used to increase the robustness of the hand localization and rough pose estimation, and the color image could heavily improve the accuracy of the pose estimation. First approaches using depth images have been presented by [37–39].

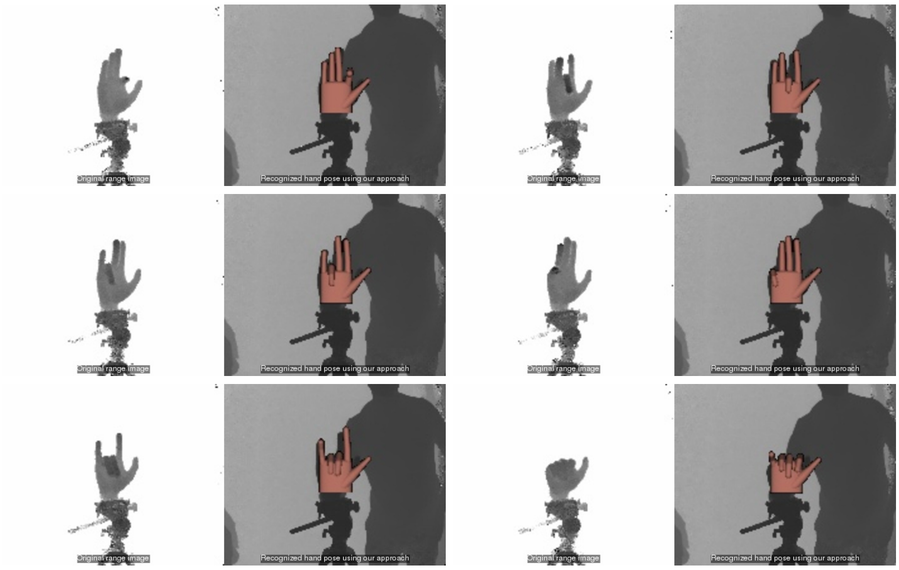

Fig. 7. The figure shows first results of the proposed RF–based method scheme using a random forest for classification for full-DOF hand pose recognition. The above images show just 6 frames taken from a larger dataset consisting of flexing and abducting the index, middle, ring and little finger in arbitrary combination. Each of the 6 frames consists of two panels. The left panel shows the depth image mapped such that you can clearly see the hand pose. On the right panel the image is mapped such that you can see the whole scene, superimposed with an artificial hand at the pose estimated by our proposed approach.

In the long-term future, when the sensor resolution of depth cameras will have been improved substantially, hand tracking can significantly benefit from depth information and the color image could become unnecessary in many situations. However, one can always find cases with many objects, one of them being the hand, that have similar distance from the camera, which make it hard to detect the hand using depth information only. In such cases, conventional color images could help a lot to detect and estimate the hand pose. Thus, the approaches using color images should be useful for the future, independent of the quality of upcoming depth cameras.

Hand tracking is a challenging task due to the high-dimensional pose space but also due to the highly non-linearity of the pose space and the high variability of the hand geometry of different persons. We took images of 5 people (Fig. 1b) and found that the palm and finger length and thickness differ a lot. Simple model-based approaches cannot handle this geometry variability appropriately because they have to use a particular hand- model. But machine learning-based approaches can cope with them. Hands with different geometry can be fed into machine learning algorithms much in the same way as different hand poses such that the learning algorithm (e.g. AdaBoosting, support vector machine or random forests) can learn them. For this reason our current work (Sec. 3) focuses on one of the most popular machine-learning approaches, the random forest. [32] and [41] also proposed to use machine learning for hand pose recognition.

Finally, we want to mention that in the past many approaches used local optimization for hand pose estimation and tracking. They need the hand pose and position to be known in the previous frame. Consequently, they need a *manual* initialization which is tedious and sometimes not practicable at all. Another consequence it that the hand motion speed is strongly limited by the computational power. Both limitations make such approaches unusable for any real applications. The future of hand tracking, thus, will focus tracking by detection approaches which estimate hand pose and position for each frame independently.

References

1. Moeslund, T.B., Hilton, A., Krüger, V.: A survey of advances in vision-based human motion capture and analysis. Computer Vision and Image Understanding **104** (2006) 90–126
2. Erol, A., Bebis, G., Nicolescu, M., Boyle, R.D., Twombly, X.: Vision-based hand pose estimation: A review. Computer Vision and Image Understanding **108** (2007) 52–73
3. Cui, Y., Weng, J.J.: Hand segmentation using learning-based prediction and verification for hand sign recognition. In: In Proc. IEEE Conf. Comp. Vision Pattern Recognition. (1996) 88–93
4. Cui, Y., Weng, J.: Appearance-based hand sign recognition from intensity image sequences. Computer Vision and Image Understanding **78** (2000) 157–176
5. Rosales, R., Athitsos, V., Sigal, L., Sclaroff, S.: 3d hand pose reconstruction using specialized mappings. In: International Conference on Computer Vision. (2001) 378–385
6. Barhate, K.A., Patwardhan, K.S., Roy, S.D., Chaudhuri, S., S.Chaudhury: robust shape based two hand tracker. In: IEEE International Conference on Image Processing. (2004) 1017–1020
7. Argyros, A., Lourakis, M.: Tracking multiple colored blobs with a moving camera. In: IEEE Conference on Computer Vision and Pattern Recognition. (2005) 1178
8. Wang, X., Zhang, X., Dai, G.: Tracking of deformable human hand in real time as continuous input for gesture-based interaction. In: International Conference on Intelligent User Interfaces. (2007) 235–242
9. John, C.: Volumetric hand reconstruction and tracking to support non-verbal communication in collaborative virtual environments. In: Dissertation submitted to the University of Otago, Dunedin, New Zealand. (2011)
10. Ueda, E., Matsumoto, Y., Imai, M., Ogasawara, T.: A hand-pose estimation for vision-based human interfaces. In: IEEE Transactions on Industrial Electronics. (2003) 676–684
11. Schlattmann, M., Klein, R.: Simultaneous 4 gestures 6 dof real-time two-hand tracking without any markers. In: ACM Symposium on Virtual Reality Software and Technology (VRST '07). (2007)
12. Schlattmann, M., Kahlesz, F., Sarlette, R., Klein, R.: Markerless 4 gestures 6 dof real-time visual tracking of the human hand with automatic initialization. Computer Graphics Forum **26** (2007) 467–476

13. Dhawale, P., Masoodian, M., Rogers, B.: Bare-hand 3D gesture input to interactive systems. In: 7th international conference on Computer-human interaction: design centered HCI. (2006) 25–32
14. 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. (2004) 693
15. Wu, Y., Lin, J.Y., Huang, T.S.: Capturing natural hand articulation. In: International Conference on Computer Vision. Volume 2. (2001) 426–432
16. Kato, M., Chen, Y.W., Xu, G.: Articulated hand tracking by pca-ica approach. In: International Conference on Automatic Face and Gesture Recognition. (2006) 329–334
17. Ouhaddi, H., Horain, P.: 3D hand gesture tracking by model registration. In: Workshop on Synthetic-Natural Hybrid Coding and Three Dimensional Imaging. (1999) 70–73
18. Nirei, K., Saito, H., Mochimaru, M., Ozawa, S.: Human hand tracking from binocular image sequences. In: 22th International Conference on Industrial Electronics, Control, and Instrumentation. (1996) 297–302
19. Zhou, H., Huang, T.: Tracking articulated hand motion with eigen dynamics analysis. In: IEEE International Conference on Computer Vision. Volume 2. (2003) 1102–1109
20. Stenger, B., Thayananthan, A., Torr, P.H.S., Cipolla, R.: Model-based hand tracking using a hierarchical bayesian filter. In: IEEE Transactions on Pattern Analysis and Machine Intelligence. Volume 28. (2006) 1372–1384
21. Stenger, B.D.R.: Model-based hand tracking using a hierarchical bayesian filter. In: Dissertation submitted to the University of Cambridge. (2004)
22. 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. Volume 12. (2004) 189
23. Mohr, D., Zachmann, G.: Fast: Fast adaptive silhouette area based template matching. In: Proceedings of the British Machine Vision Conference, BMVA Press (2010) 39.1–39.12 doi:10.5244/C.24.39.
24. Mohr, D., Zachmann, G.: Segmentation-free, area-based articulated object tracking. In Bebis, G., Boyle, R., Parvin, B., Koracin, D., Wang, S., Kyungnam, K., Benes, B., Moreland, K., Borst, C., DiVerdi, S., Yi-Jen, C., Ming, J., eds.: 7th International Symposium on Visual Computing. Volume 6938 of Lecture Notes in Computer Science., Springer (2011) 112–123
25. 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)
26. Borgefors, G.: Hierarchical chamfer matching: A parametric edge matching algorithm. In: IEEE Transaction on Pattern Analysis and Machine Intelligence. (1988)
27. 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)
28. 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)
29. 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)

30. Gavrila, D., Philomin, V.: Real-time object detection for "smart" vehicles. Volume 1., Los Alamitos, CA, USA, IEEE Computer Society (1999) 87
31. 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)
32. Thayananthan, A., Navaratnam, R., Stenger, B., Torr, P., Cipolla, R.: Multivariate relevance vector machines for tracking. In: European Conference on Computer Vision. (2006)
33. Olson, C.F., Huttenlocher, D.P.: Automatic target recognition by matching oriented edge pixels. In: IEEE Transactions on Image Processing. (1997)
34. Shaknarovich, G., Viola, P., Darrell, T.: Fast pose estimation with parameter-sensitive hashing. In: IEEE International Conference on Computer Vision. (2003)
35. Athitsos, V., Sclaroff, S.: Estimating 3D hand pose from a cluttered image. In: IEEE Conference on Computer Vision and Pattern Recognition. (2003)
36. Mohr, D., Zachmann, G.: Continuous edge gradient-based template matching for articulated objects. In: International Joint Conference on Computer Vision and Computer Graphics Theory and Applications. (2009)
37. Gudmundsson, S.A., Sveinsson, J.R., Pardas, M., Aanaes, H., Larsen, R.: Model-based hand gesture tracking in ToF image sequences. In: 6th International Conference of Articulated Motion and Deformable Objects. (2010) 118–127
38. Hackenberg, G., McCall, R., Broll, W.: Lightweight palm and finger tracking for real-time 3D gesture control. In: IEEE Virtual Reality Conference. (2011) 19–26
39. Oikonomidis, I., Kyriazis, N., Argyros, A.: Efficient model-based 3d tracking of hand articulations using kinect. In: BMVC 2011, BMVA (2011)
40. Oikonomidis, I., Kyriazis, N., Argyros, A.: Tracking the articulated motion of two strongly interacting hands. In: CVPR 2012, IEEE (2012) to appear.
41. Keskin, C., Kurac, F., Kara, Y.E., Akarun, L.: Hand pose estimation and hand shape classification using multi-layered randomized decision forests. In: Proceedings of the 12th European conference on Computer Vision - Volume Part VI. ECCV'12, Berlin, Heidelberg, Springer-Verlag (2012) 852–863
42. Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., Blake, A.: Real-Time Human Pose Recognition in Parts from Single Depth Images. (2011)
43. Lepetit, V., Lagger, P., Fua, P.: Randomized trees for real-time keypoint recognition. In: Computer Vision and Pattern Recognition. (2005) 775–781
44. 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. (2001) 23
45. de La Gorce, M., Paragios, N.: A variational approach to monocular hand-pose estimation. Computter Vision and Image Understanding **114** (2010) 363–372
46. de La Gorce, M., Paragios, N., Fleet, D.J.: Model-based hand tracking with texture, shading and self-occlusions. In: IEEE Conference on Computer Vision and Pattern Recognition. (2008)
47. Oikonomidis, I., Kyriazis, N., Argyros, A.: Full dof tracking of a hand interacting with an object by modeling occlusions and physical constraints. In: ICCV 2011, IEEE (2011)
48. Kulis, B., Darrell, T.: Learning to hash with binary reconstructive embeddings. In: In Proc. NIPS. (2009) 1042–1050
49. Kulis, B., Grauman, K.: Kernelized locality-sensitive hashing for scalable image search. In: IEEE International Conference on Computer Vision (ICCV. (2009)

50. Athitsos, V., Potamias, M., Papapetrou, P., Kollios, G.: Nearest neighbor retrieval using distance-based hashing. In: Proceedings of the 2008 IEEE 24th International Conference on Data Engineering. ICDE '08, Washington, DC, USA, IEEE Computer Society (2008) 327–336
51. Athitsos, V., Alon, J., Sclaroff, S., Kollios, G.: BoostMap: An Embedding Method for Efficient Nearest Neighbor Retrieval. Pattern Analysis and Machine Intelligence, IEEE Transactions on **30** (2008) 89–104
52. Tomasi, C., Petrov, S., Sastry, A.: 3d tracking = classification + interpolation. In: International Conference on Computer Vision. (2003) 1441–1448
53. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: IEEE Conference on Computer Vision and Pattern Recognition. Volume 1. (2001) I–511–I–518
54. Breiman, L.: Random forests. Machine Learning **45** (2001) 5–32
55. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1993)
56. Ho, T.K.: Random decision forests. In: Proceedings of the Third International Conference on Document Analysis and Recognition (Volume 1) - Volume 1. ICDAR '95, Washington, DC, USA, IEEE Computer Society (1995) 278–
57. Fukunaga, K., Hostetler, L.: The estimation of the gradient of a density function, with applications in pattern recognition. IEEE Trans. Inf. Theor. **21** (2006) 32–40