A Survey of Vision-Based Markerless Hand Tracking Approaches

Daniel Mohr, Gabriel Zachmann

University of Bremen, Germany

Abstract

Vision-based markerless hand tracking has many applications such as virtual prototyping, navigation in virtual environments, tele- and robot-surgery and video games. Due to the real-time condition and the high complexity of the hand, which has its main reasons in the 27 degree of freedom and the high appearance variability, hand tracking is a very challenging task with increasing attention in the computer vision community. 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 in real-time.

The purpose of this survey is to give an overview of the approaches presented. First, we will explain the challenges in more detail. Second, we will classify the approaches, and finally, provide the most important approaches.

Keywords: Articulated Object Tracking, Hand Tracking, Hand Pose Estimation

1. Introduction

The purpose of this survey is not to give an exhaustive overview of all approaches but of the most important ones, and provide an overview of the research area and the different methods to solve the hand pose estimation task.

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 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 resource.

Email address: {mohr,zach}@cs.uni-bremen.de (Daniel Mohr, Gabriel Zachmann)

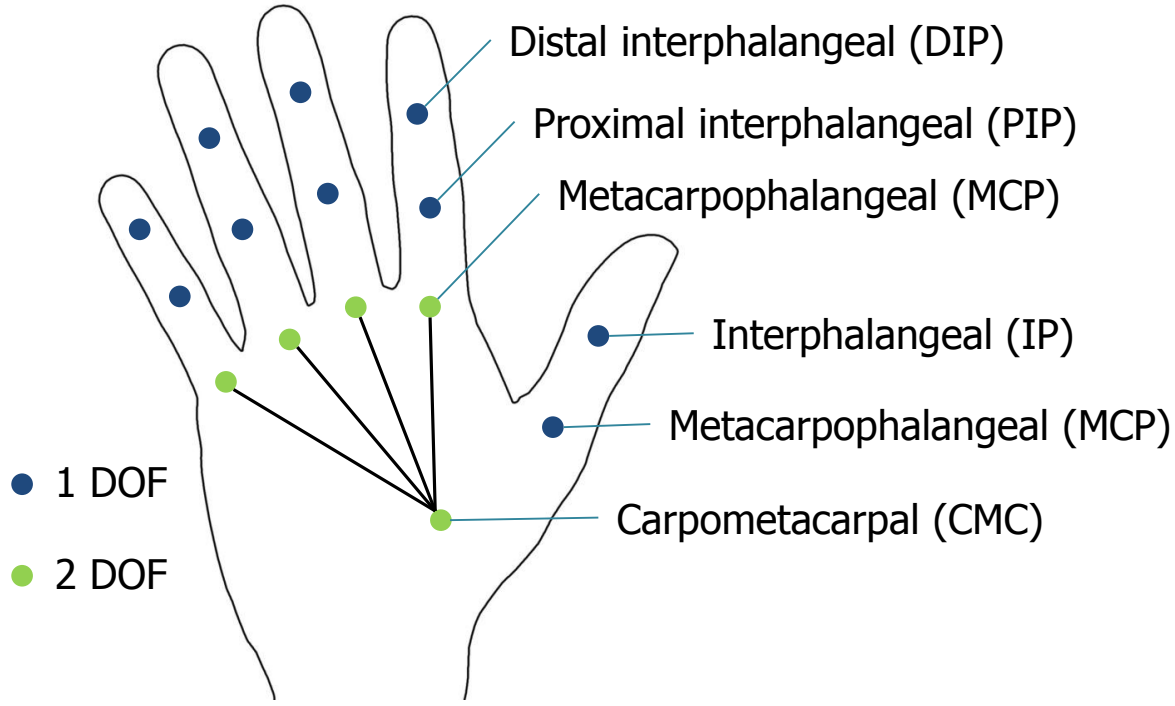

Figure 1: Illustration of the degrees of freedom (DOF) of the human hand. The valid hand poses form a manifold in 21-dimensional space. With the additional 6 global DOFs (translation and rotation), we arrive at 27 DOFs. The names of all joints can be found on the right-hand side of the illustration.

Thus, markerless vision-based hand tracking in real-time is an active research area. In the recent years, the research speed has increased due to the high interest in this topic. Consequently, to get an overview of this large research area, a survey is essential. To our knowledge, the most recent surveys on computer vision based hand tracking are 4 years [1] and 6 years [2] ago. This is half an eternity in such an active research area. Thus, a more up-to-date survey is necessary, which we will provide with this paper.

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.

1.1.1. High-dimensional Configuration Space

The problem dimension to estimate the foll-DOF hand pose is very high. Figure 1 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 27 dimensions.

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

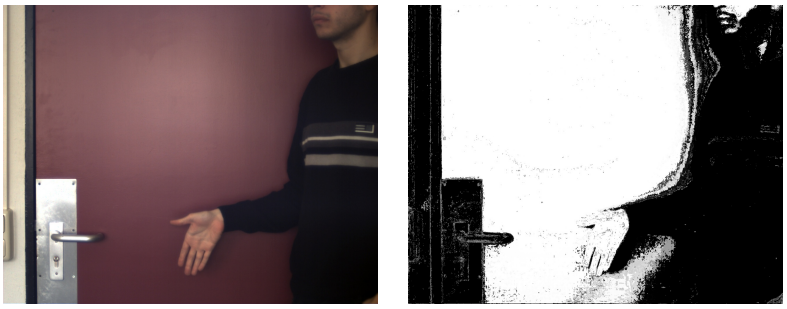

Figure 2: A skin color-based hand tracking approach will fail in the example image (left) due to the red door in the background. The reason is that the skin segmentation (right) will classify most of the door as skin.

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.

1.1.3. 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 (Fig. 2) 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 or a picture as shown in Figure 3) will produce a lot of edges in the background that heavily disturbs any edge-based matching.

1.1.4. 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.[1]

[1]Having the whole light spectrum, on could detect skin more reliable, and consequently would simplify the hand detection task significantly.

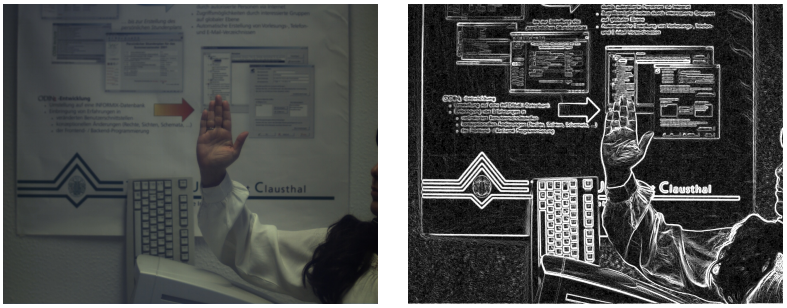

Figure 3: An edge-based hand tracking approach will yield a low matching quality in the left image due to the large amount of edges (right image) in the background.

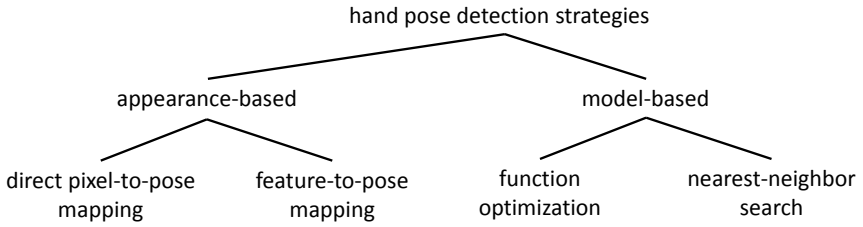

Figure 4: Hand tracking approaches can be classified into appearance and model-based approaches. Appearance-based approaches use direct mapping techniques. In contrast, model-based approaches fit a hand model to the input image to estimate the pose. Pose estimation can be formulated as function optimization (the hand pose is the parameter set to be optimized) or nearest-neighbor search (find the hand pose most similar to the observed hand).

1.1.5. 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.

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 survey 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 researches, 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.

2. Classification of Approaches

In this section, we will present ways to organize all hand tracking approaches in a taxonomy. Actually, there are several ways to categorize the approaches [3, 2]. A lot of publications in the area of hand tracking focus on the classification of a fixed number of gestures, others try to estimate the full DOFs including all finger joint angles. Hand gesture classification can be done efficiently through classification algorithms, e.g. support vector machines (SVM) or random trees. Recently, in the area of whole body tracking, an approach for full pose estimation through classification algorithms has been presented [4, 5]. But it is very questionable whether the application of this approach to the problem of hand tracking would work; this is mainly due to the larger appearance variability of the hand compared to the whole body.

Most of the hand tracking approaches today use some kind of fitting, i.e., the hand model or parts like fingers or finger tips are matched against the input image. This leads us to another way to categorize hand tracking approaches: classification or fitting-based approaches.

One can also differentiate between approaches that are able to automatically initialize the pose and approaches that need a manual initialization. Approaches with automatic initialization use a global search of the hand pose in the configuration space. By contrast, approaches with manual initialization apply only a local search in the neighborhood of the pose in the previous frame (trying to exploit temporal coherence).

Another widely followed categorization divides hand tracking into the following two categories: appearance-based and model-based (Fig. 4). 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

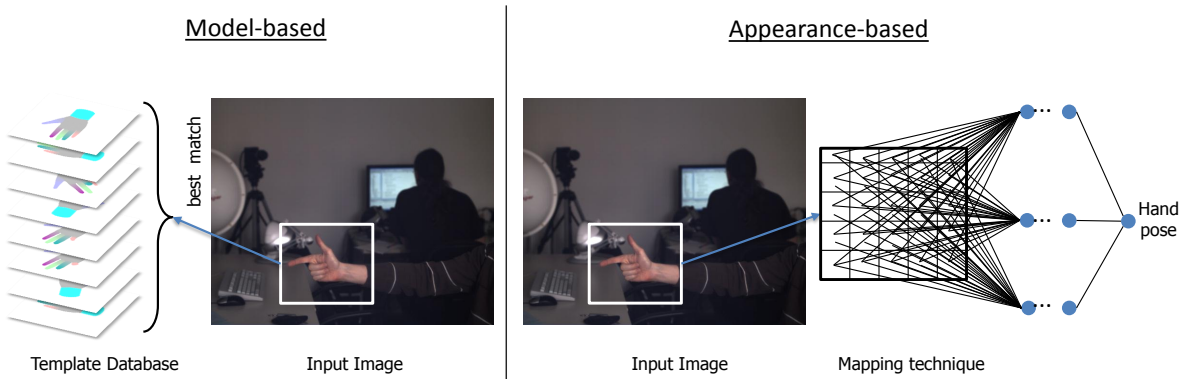

Figure 5: Model-based approaches (left) use an object model (here the human hand) and match the templates, each representing a hand pose, to the input image. In contrast, appearance-based approaches (right) try to learn a direct mapping from the image space to the pose space.

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 abound 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 5 visually compares the idea of model- and appearance-based approaches.

2.1. Appearance-based Approaches

A typical appearance-based approach is used in [6, 7] 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 [8]. After a skin segmentation, moment-based features are computed 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 [9]. 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 [10] 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.

[11] 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.

Appearance-based approaches are also popular in the area of human body tracking. Basically similar approaches as used for hand tracking can be applied but, compared to the hand, the appearance variability for the whole body is by far lower. In [12] a statistical body segmentation is applied and low-level features

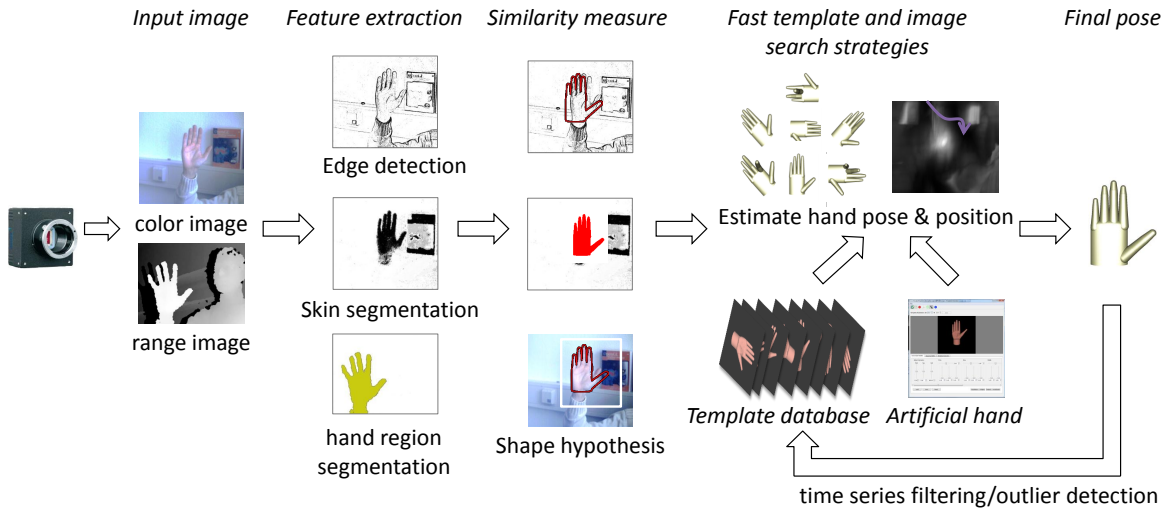

Figure 6: A typical model-based hand tracking pipeline has 3 core methods: feature extraction, similarity measure between hand hypothesis (typically denoted as templates) and input image, and a (fast) search strategy to minimize the cost for the simultaneous hand position and pose estimation.

extracted. A mapping from this low-level features to the 2D body pose is trained using a set of examples. This is done by first applying the Expectation Maximization (EM) algorithm to the examples. A mapping function from the resulting clusters to the 2D pose space is trained. Given a new visual feature, a mapping from each cluster is performed and the most likely chosen to be the most probable body pose.

Felzenszwalb et al. [13] uses difference of Gaussians (DoG) as features. They build a tree-structured graph that roughly matches to the human body structure. Minimization is performed through the Viterbi algorithm. In an earlier work [14] they used the color mean and variance of rectangular regions as features.

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 rest of the finger, the above approaches will fail to detect the finger. It is not even easy to determine which of the fingers is occluded.

A promising alternative are model-based approaches.

2.2. 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. Depending on the needs of the approach (number of poses that have to be detected, computational power of target device) the templates are precomputed or generated online during tracking. Figure 6 gives an overview of a typical model-based pipeline. 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.

Most approaches for articulated object tracking use edge features and/or a foreground segmentation as a preprocessing step. Similarity measures between the target object and the input image are defined based on these features.

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

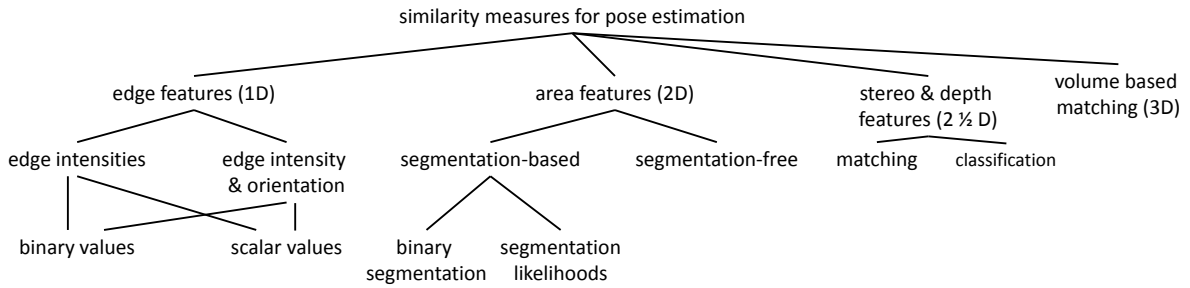

Figure 7: Similarity measures for hand pose estimation can be categorized into four different measure types. The categorization is based on the input modality used for matching. The most often used modalities are edges and the hand silhouette area. Edge intensities, edge gradient and the segmentation 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. The other two matching methods are based on depth information and visual hull reconstruction.

robustly as well because the similarity measure between a hypothesis and an input image is only affected by a limited amount.

A typical model-based pipeline is shown in Fig. 6. It has two *core* components:

1. *similarity measures* are used to test a hand pose hypothesis given an input image. The overall hand tracking quality heavily depends on the discriminating power of the similarity measure itself. Additionally, the computation time of a similarity measure linearly influences the overall matching time because hypothesis testing consumes most of the overall computation time.
2. smart *acceleration data structures for hypothesis testing* are essential for hand tracking due to the very large configuration space, and consequently the huge number of hypotheses that would have to be tested using a naïve approach. It is crucial to minimize the number of hand poses (i.e. hypotheses) that have to be matched against the input image.

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.

3. Similarity Measures

In the area of hand tracking the most often used features 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. Other similarity measures are based on different input modalities such as stereo or range images. Volume reconstruction-based approaches directly work on a 3D volume, but these approaches have a large drawback. We will discuss these approaches in Sec. 5. Fig. 7 gives an overview of the different similarity measure classes.

Skin color is used to segment the hand from the background, which yields the hand silhouette area. Based on the silhouette area, similarity measures are needed to compare the similarity/difference between the silhouette area of a hypothesis and the segmentation result of the input image.

Edge features are used in a similar way. First, edges are extracted from the hand hypothesis and the input image; then a distance measure between the resulting edge images is used for matching.

In the following, we will first take a look at silhouette area-based similarity measures; then we will give an overview of edge-based measures.

3.1. 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 modes and the hand model contains no noise, thus there is no information gain using a non-binary representation. [2]

A simple method belonging to the first class is proposed in [15]. They assume that the hand is in front of a homogeneous, uniformly colored background. First, they applying 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 [16] and [17]. 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. [18]. 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 [19], the non-overlapping area of the model and the segmented silhouettes are integrated into classical optimization methods, e.g. Levenberg-Marquardt or downhill simplex. [20] 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.

In [21, 22] 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 then the vector is compared to the database.

A completely different approach is proposed by Zhou and Huang [23]. Although they extract the silhouette from the input image, they use only local features extracted from the silhouette boundary. Their features are inspired by the SIFT descriptor [24]. Each silhouette is described by a set of feature points. The chamfer distance between the feature points is used as similarity measure.

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 our knowledge, there are much fewer approaches working directly on the skin color likelihood map of a segmentation. The skin likelihood map contains for each pixel the probability that it belongs to a region consisting of skin. In [25] 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 [26, 27, 28], the skin color likelihood map is directly compared to the hand silhouette. Given a hypothesis, the silhouette foreground area (Fig. 8) 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 probability of all pixels that correspond to the foreground is computed and the inverse

[2] Theoretically, it could make sense if one would combine multiple models e.g. varying hand geometry into one hypothesis.

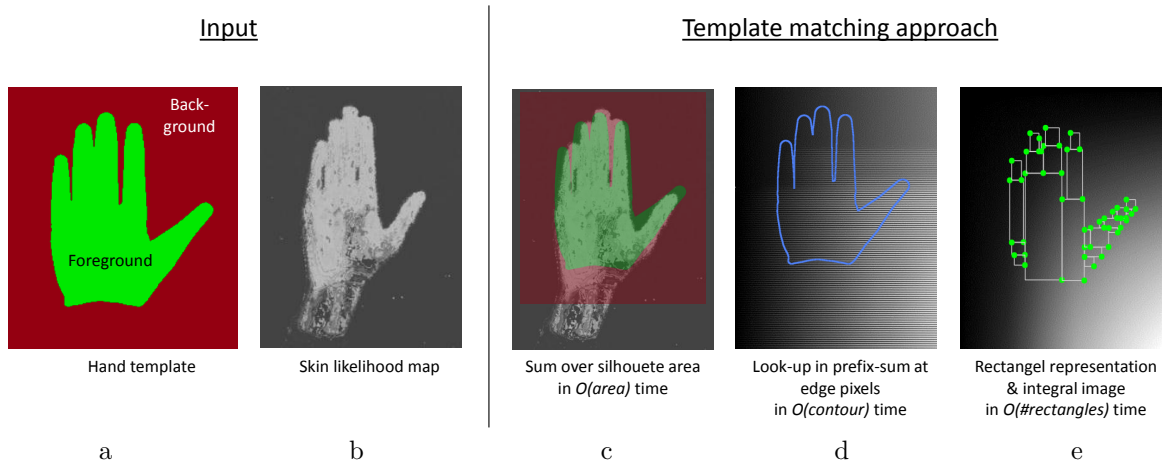

Figure 8: Illustration of silhouette area based matching approaches. The matching consists of computing the similarity measure between a hand hypothesis represented by the silhouette foreground and background and the input image represented by the skin likelihood map. A naive approach needs O(#pixels) time. Stenger et al. [26] proposed an approach to reduce the computation time to O(contour-length) utilizing the prefix-sum. Mohr et al. [29] further reduced the computation time to O(near-const).

probabilities corresponding to the background respectively. The fore- and background joint probabilities are combined to the final similarity measure. The drawback of this measure is that its complexity linearly depends on the number of pixels, and thus, on the image resolution and hand size in the input image.

Stenger et al. [26, 27] proposed a method to reduce the computation complexity to be linear to the contour length. To this end, he utilized the prefix sum as acceleration structure. First, the product in the joint probability is converted into sums by simply computing the joint probability in log-space. The row-wise prefix sum in the log-likelihood image is computed. The original product along all pixels in a row reduces to three look-ups in the prefix sum. With this, he reduced the computational complexity of the similarity to be linear to the contour length which is the square root of the naive computation complexity. Figure 8d illustrates the basic idea.

[29] further reduced the computational complexity to compute the joint probability as similarity measure proposed by [26, 27] from linear to near-constant time (Fig. 8e). 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. There are mainly two ways to segment the hand: background subtraction [30] and skin segmentation [31]. The former makes simplifying assumptions about the background color distribution, which is especially problematic if the background is non-static, and the latter one about the skin color. 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 (this problem is further discussed at the end of this subsection).

[32] 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 [29] 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.

[33] uses a completely different segmentation-based approach by requiring the user to wear a colored

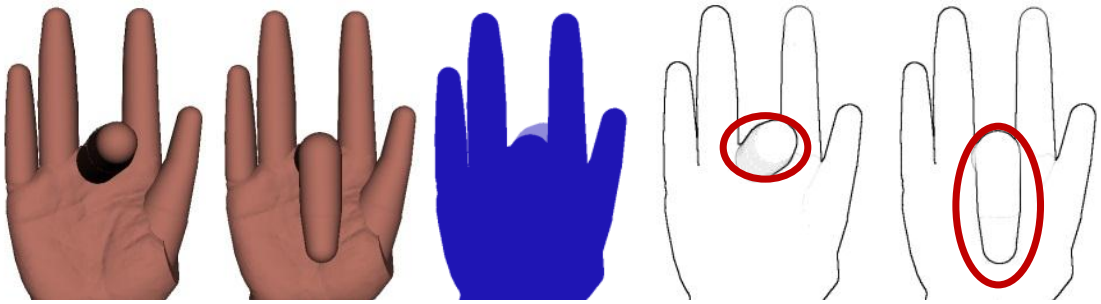

Figure 9: Silhouette area-based similarity measures cannot resolve all ambiguities. Two example hand poses (left) illustrate the problem. Using the silhouette-area as feature does not allow to distinguish between both poses because the difference area (middle image, light blue area) is by far to small. 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.

glove. The descriptor is based on the gloves color coding. Each of the colors correspond to a specific part of the hand. In a preprocessing step, the authors generate a database, containing descriptor instances for a large number of hand poses. During tracking, they compare the database 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.

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 9.

3.2. 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. 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 similarity or, equivalently, a distance measure between the edges is defined to compute the similarity between a hypothesis and the input image. In order to outline the most popular distance measures, we first have to introduce some notations. Let I_A and I_B be two edge images and A and B the set of coordinates of the edge pixels.

Then, one can use the Hausdorff distance from A to B as distance measure between I_A and I_B. The directed Hausdorff distance with respect to metric d [34] is defined as the maximum of all distances from each point in A to its nearest neighbor in B:

$$\mathcal{H}(A,B) = \max_{a_i \in A}\{\min_{b_j \in B}\{d(a_i,b_j\}\}. \tag{1}$$

The generalized form uses the kth largest distance instead of the maximum,

$$\mathcal{H}(A,B) = \underset{a_i \in A}{k\text{th}}\{\min_{b_j \in B}\{d(a_i,b_j\}\} \tag{2}$$

where kth returns the k-largest value. The value k can be used to control the number of outliers that are tolerated. In the area of hand tracking, I_A would be used as template and I_B as input image.

An approximation of the Hausdorff distance is the chamfer distance, which replaces the max operator by the sum. The directed chamfer distance [35], [36] $\mathcal{C}$ from set A to B is defined as

$$\mathcal{C}(A,B) = \frac{1}{|A|} \sum_{a_i \in A} \min_{b_j \in B} d(a_i, b_j) \tag{3}$$

The chamfer distance can be formulated as a convolution of image I_A with the distance transform of image I_B, and then, computed faster in Fourier space. Chamfer matching for tracking of articulated objects is, for example, used by [26], [37], [38], [39], [40], [28], [18] and [41]. 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 [42], [26]. To achieve some robustness against outliers, [26] additionally limited the nearest neighbor distance from a point of set A to set B by 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.

[43] 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. [28] presented a similar approach to incorporate edge position and orientation into chamfer distances.

Edge orientation information is also used by [44] 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 [45] 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.

[46] 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.

[47] combine image edges with optical flow and shading information. They use a generalized version of the gradient-based optical flow constraint, that includes shading flow. Using this model, they track the articulated motion in the presence of shading changes. A forward recursive dynamic model is used to track the motion in response to data derived 3D forces applied to the model.

[48] search for the fingers. Basically, the finger candidates are identified by the finger boundary edges. They first compute the edge response using the Sobel operator. Next, the image is convolved by a specific kernel containing the finger contour distance constraints. The candidates are scored by evaluating the image content around the candidate location. Hysteresis thresholding is applied and finally the center position of the fingers detected using Camshift. Applying a second Camshift to the skin segmentation gives the hand center. The hand principal axis is computed as the difference between the center of the fingers and the center of the palm.

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.

3.3. Other Similarity Measures

Edges and skin color have proven to work fairly robust in many cases, but one could consider other features such as shading, texture, and optical flow to improve the matching quality. Only a few approaches have been proposed that use such features, which will be surveyed in the following.

[49] uses hand texture and shading information. Pose parameters and shading are integrated into an objective function. For shading information, they use the pixel-wise difference of color values between the artificial hand model and the input image. Partial derivatives with respect to the pose space parameters are taken. The resulting objective function is estimated using partial derivatives with respect to the pose space parameters and the quasi-Newton optimization method. They have to handle the silhouette contour separately because the derivative is not continuously differentiable. Texture information is estimated per frame after the pose estimation to update the artificial hand model. The approach needs two strong conditions: first, the artificial hand model has to be very close in shape and shading to the human hand to be tracked, which is very difficult to realize; and second, the approach uses a local optimization, which needs a manual initialization step.

[50] combine edge features, optical flow, and additional salient features. As salient features they propose to use finger nails. A Hough forest is first trained and then used to detect the finger nails. The nail correspondences problem is solved using integer programming. Differences of edges, optical flow, and nail positions between hypothesis and input image are combined in an optimization function and minimized by the Levenberg-Marquardt algorithm. The approach is computation-intensive and can, thus, be hardly used in real-time tracking systems.

4. 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. 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.

The hand pose space forms a manifold in a high dimensional space. Depending on the hand model the pose space has up to 27 dimensions.[3] Consequently, in a typical tracking application, a huge number of templates have to be matched. Additionally, at initialization, there is no previous knowledge about position and state of the target object. Thus, one has to scan the complete input image to find the hand pose and position. Of course, it is prohibitive to compare all templates to the input image due to the real-time tracking condition. Thus, smart acceleration data structures are crucial so as to reduce the number of templates and also the number of positions in the input image the templates have to be matched to.

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. Making the assumption that the object does not move very fast allows the approach to apply a local image space and pose space search. Perfect image segmentation trivially allows to determine position and size of the object. This also heavily reduces the search space because the location of the object does not have to be found, which significantly simplifies the pose estimation task.

[3] The 27th dimension is the torsion angle of the thumb. The reason this additional DOF is often included in the hand model is that the flexion and abduction angles cannot describe all valid thumb positions due to its complex joint configuration and the surrounding muscles and tissue.

In [22], 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 [28]. 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 [51, 49]. 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 [49] and skin color in [51].

[16] 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.

[52] 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.

Exemplar-based matching is basically the same task as image retrieval (image database query), as for example used by several Internet search engines. Given a query image and an image database, the task is to find the images in the database most similar to the query image. Converted to the problem of pose estimation, the database contains the object in all poses and the query image corresponds to the input image (e.g. obtained using a camera).

Many image database query approaches extract salient features e.g. color histograms, texture information and coarse object silhouettes from the image. A descriptor is built based on the features and often further compressed to obtain a very compact descriptor that can be compared extremely fast to the database. Due to the very large databases, comparing all descriptors still is too slow, and thus, acceleration data structures are utilized for nearest neighbor search (in many applications, the approximate nearest neighbor is sufficient).

Hashing, for example, is a popular data structure in the image retrieval research field. Several variants of hashing are used, e.g. semi-supervised hashing [53], locality sensitive hashing (LSH) [54, 55] and modulo-based hashing with a complex binary image descriptor as hash key [56]. Image database query approaches try to find visually similar looking images, for example, identify images that contain a specific landscape or the same object such as a car.

Object detection approaches, in contrast, need to find images from the database that best match with respect to specific features e.g. the silhouette or edges. Basically, one only needs to use different features to be able to apply image retrieval techniques to object detection. But the very compact image retrieval descriptors are not able to characterize the object silhouette or edges appropriately. Thus, the descriptor has to be redesigned and the acceleration data structure adapted.

Such an approach is for example proposed by [44]. They extended the LSH to the needs of human pose estimation. They learned a set of binary hash functions offline from training examples. Each hash function

is trained from a training example pair, and hash values of +1 or -1 are assigned, depending on whether the distance between the elements of the pair is below or above a threshold. A subset of hash functions are selected that minimize the classification error. Given a query image, the corresponding hash value is computed and LSH utilized for a fast nearest neighbor search in a database consisting of example poses.

Hashing is also used by [57] 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 main challenges of hashing for pose estimation are to choose the optimal size of the hash table(s) and good locality preserving hashing functions i.e. hashing functions, such that similar poses are always hashed to neighboring table entries and different hand poses to different entries. There is always a trade-off between missing true positive matches (e.g. too large a hash table) and too many false positive matches (e.g. too small a hash table or unfit hash function).

The idea to convert the evaluation of similarity measures to vector distances is used in [39, 58]. 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.

The main challenges using embedding techniques for pose estimation are the choice of the mapping functions and the number of dimensions. A bad mapping function can make the matching fail in many cases. Too low a dimension also leads to bad matching results, while using too many dimensions mighty be computationally prohibitive.

Several other approaches are proposed to reduce the computation time and the number of the similarity measures for exemplar based object detection. A relevance vector machine (RVM) is used in [59] to reduce the number of human pose examples the input has to be matched to. They extract the shape silhouette and encode it using histogram-of-shape context descriptors to get some robustness to silhouette errors. The vector quantization of the descriptor is used as an input for the RVM. They "train the regressors on images resynthesized from real human motion capture data". Pose estimation is formulated as a one-to-one mapping from feature space to pose space.

Thayananthan et al. [42] also used a 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.

RVM's have a low generalization error, but they were originally designed for scalar outputs. For hand pose estimation they have to be extended to multivariate outputs. The main problem of RVM's for hand pose estimation is the mapping, which basically is a weighted average of functions comparing learned hand poses with the input image. Due to the highly nonlinear hand pose space, the RVM's can only locally perform well. Thus, a huge number of RVM's have to be combined to cover the whole hand pose space. This could lead to high computation times and potentially to limited accuracy at the intersections between the individual RVM's.

Another widely used acceleration data structure are hierarchies. Hierarchies can be built in pose or feature space. A model-based tracking approach for multiple humans is proposed in [60]. They build a combined hierarchy over the set of humans and the pose space of each human. In order to search for the pose of the ith human in the scene, they synthesize humans with the best pose parameters found earlier. Then, they search for the best torso/head configuration of the ith human while keeping the limbs at their predicted values. Chamfer matching is used for hypothesis testing.

In contrast, a feature space hierarchy is utilized in [61]. They use an agglomerative clustering approach based on the chamfer distance between object edges to build the hierarchy. Cluster prototypes represent the nodes in the corresponding tree. Given a query image, the matching starts at the root node and successively

visits the child nodes. For each node, the subtree is only further traversed if the chamfer distance to the query image is below a threshold. They have measured a speedup of three orders of magnitude.

[62] 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.

[26] 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 [26], [29] 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 general, template hierarchies have to be built carefully: templates have to be partitioned into child nodes such that similar templates end up in the same node. Otherwise, during the matching, the probability that the wrong child node is chosen (and, thus, the best matching hand pose is discarded) becomes too high. This problem can be alleviated to some degree by multi-hypothesis tracking, i.e., following multiple paths during matching.

A "degenerated" version of hierarchical matching is cascading-based matching. The idea of a classical hierarchy is to keep the computation time low by minimizing the number of matches in the upper tree levels. In contrast, the idea behind cascading is to still match all candidates, but heavily reduce the computation time for each match. In this way, the root node consists of a very fast but inaccurate measure, and the leaf/leaves of more expensive and accurate measures.

The idea of cascading was first proposed in [63]. They use a large set of features and learn the most discriminating one utilizing AdaBoost. Then, they combine "increasingly more complex classifiers in a cascade, which allows background regions of the image to be discarded early while spending more computation time on more promising object-like regions". For evaluation, they applied their approach to face detection.

The idea of cascading is also used in [64] for face detection. First, they perform feature reduction by choosing relevant image features using statistical learning approaches. Second, they build a hierarchy of classifiers. "On the bottom level, a simple and fast classifier analyzes the whole image and rejects large parts of the background. On the top level, a slower but more accurate classifier performs the final detection".

In [38], cascading 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.

The risk with cascading lies in the choice of the optimal balance between simple (but fast) and complex (but slow) classifiers to reject true negatives. If the features are too simple, true positives could be rejected as well. If the features are more complex, the approach may become computationally inefficient but still does not have the guarantee that no true positives are rejected.

5. Other Approaches

Some approaches cannot clearly be categorized into one of the above categories. In this section, we will provide an overview of such approaches.

Depth information has the potential to replace edge features because all edges belonging to hand and the contour of the fingers that are relevant for matching are also visible in the depth images. The reason is that at the hand contour, there is always a depth discontinuity which results in different depth values.

Additionally, depth images also have area information and, thus, can be used for area-based similarity measures.

5.1. Approaches Based on Depth Information

Several approaches using depth information have been presented. [65] used two cameras positioned at the same distance as the human eyes. A foreground segmentation was applied, filtered and edges based on the segmentation extracted. Both segmentation and edges are used to compare the hypothesis to the input images. A particle filter is used for hand pose prediction. The input images from the two cameras are treated separately and combined as a last step in the tracking pipeline.

A disparity map is generated in [66] to estimate the depth at all image positions. A hand model consisting of cones and spheres is matched to the depth image using an ICP (Iterative Closest Point) technique. The tracker is initialized manually and the pose in the next frame predicted using Kalman filtering. Only moderate frame-to-frame movement is allowed.

The main problem of stereo images are the correspondence problem (i.e. find corresponding point in the images from the two cameras) and, consequently the limited reconstruction accuracy of a disparity map. Additionally, disparity maps are generated using images from only two viewpoints, which by far is not enough to resolve all ambiguities, and it still suffers from the problem of conventional color images (edge noise, segmentation errors, etc.).

A better way to recover the 3d information are range cameras. In the recent years, cameras providing depth information became more and more available. For several years, Time of Flight (ToF) cameras are available.

[67] used projective geometry to match the hand template to the depth image. First, a part of the 27-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 [68], 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.

ToF-cameras are very expensive. In 2010, Microsoft released a low-price alternative to the ToF cameras: the Kinect. The technique behind the Kinect is completely different (the distortion of a set of projected points is used to estimate the depth), but the output is similar.

[69] integrated the Kinect into their hand tracking algorithm. The hand is localized conventionally through skin segmentation. Hand pose estimation is formulated 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 [70] they extended the approach to track two interacting hands.

A "tracking by classification" approach is proposed by [71]. They adapt the method of human body tracking [72] to hand tracking. In [72], 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 [73]. After classification of each position in the depth image, they compute the body part positions by the mean shift algorithm. [71] 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.

Currently, the resolution and accuracy of depth cameras is too low the for a full-DOF hand pose estimation, but the potential of depth information as input modality is high. Additionally, one has to be aware that depth cameras still provide only the depth values from one viewpoint i.e. they do not provide the visual hull of the hand.

5.2. Approaches Based on Visual Hull Reconstruction

In this section we present object detection and tracking approaches based on visual hull reconstruction. Usually, they use a set of conventional cameras.

The basic idea is to first reconstruct the visual hull of the hand using some kind of segmentation technique, and then estimate the hand pose based on the reconstruction.

Visual hull reconstruction is a challenging task. [74] shows that a high quality reconstruction requires a high quality foreground segmentation and a lot of cameras surrounding the hand. Furthermore, the visual hull computation is very time consuming, and cannot be computed in real-time in scenes with a complex background.

But, using a more simple visual hull reconstruction, one can coarsely estimate the hand pose or recognize a few hand gestures.

[75] first reconstruct a rough voxel map of the observed hand. The hand in each image was first segmented and then the different 2D silhouettes combined to generate the 3D shape. The hand hypothesis is fitted in 3D to the reconstructed voxel model to find the underlying hand pose.

A similar approach is used in [76, 77] to detect the global position and orientation of the hand. Simultaneously, they can recognize four different gestures. The hand is captured using 3 near-infrared (NIR) cameras. The hands are segmented using background subtraction. The foreground segmentation is reprojected onto 3D for all cameras. The 3D hand volume is reconstructed by intersecting the re-projection volumes. Finger tips are detected and, based on the number and relative position, the gesture classification is performed. They are able to detect two hands simultaneously in real-time and tested its functionality in multiple applications, e.g. games.

However, such visual hull based approaches are very limited in their application to hand tracking, because the cameras have to be positioned around the hand, the background has to be very simple and the reconstructed visual hull is too coarse or computationally too expensive.

Recently, similar to the visual hull reconstruction, a 3D object reconstruction utilizing depth images (using the Kinect) are proposed [4, 5]. But the camera has to take capture the target object from many different viewpoints to be able to reconstruct the object which is not yet practicable in real-time.

6. Conclusions

6.1. Summary

In this survey, 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 presented that tried to solve the problem. Many approaches make an important contribution to the robust real-time hand tracking. Several algorithms can, of course, be combined to increase the robustness, e.g. edge-based, silhouette-area based features, and depth information. Combining acceleration data structures is more challenging because, often, they exclude each other, e.g. a template hierarchy and hashing with a hash table built from templates.

The most recent research focuses on the currently very popular Kinect or similar depth sensors. Some promising approaches have been presented but, it turns out that current hardware provides too a low resolution for a full-DOF hand pose estimation. Only small subsets of the pose space e.g. open hand with abducted fingers and all finger tips clearly visible can be estimated successfully.

6.2. The future of hand tracking

In the near future it might be a good idea to combine depth with conventional high resolution color 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.

When the sensor resolution of depth cameras will heavily be improved in the future, hand tracking can significantly profit from depth information and the color image could become unnecessary in many situations. However one could think about cases with many objects at similar distance to the camera as the hand, which

makes 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.

References

[1] X. Zabulis, H. Baltzakis, A. Argyros, Vision-based hand gesture recognition for human-computer interaction, in: C. Stephanidis (Ed.), The Universal Access Handbook, Human Factors and Ergonomics, Lawrence Erlbaum Associates, Inc., 2009, pp. 34.1–34.30.

[2] A. Erol, G. Bebis, M. Nicolescu, R. D. Boyle, X. Twombly, Vision-based hand pose estimation: A review, Computer Vision and Image Understanding 108 (2007) 52–73.

[3] T. B. Moeslund, A. Hilton, V. Krüger, A survey of advances in vision-based human motion capture and analysis, Computer Vision and Image Understanding 104 (2006) 90–126.

[4] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, et al., Kinectfusion : Real-time 3d reconstruction and interaction using a moving depth camera, International Symposium onMixed and Augmented Reality (2011) 1–10.

[5] S. Izadi, R. A. Newcombe, D. Kim, O. Hilliges, S. H. David Molyneaux, P. Kohli, J. Shotton, A. J. Davison, A. Fitzgibbon, Kinectfusion: Real-time dynamic 3d surface reconstruction and interaction, in: SIGGRAPH.

[6] Y. Cui, J. J. Weng, Hand segmentation using learning-based prediction and verification for hand sign recognition, in: In Proc. IEEE Conf. Comp. Vision Pattern Recognition, pp. 88–93.

[7] Y. Cui, J. Weng, Appearance-based hand sign recognition from intensity image sequences, Computer Vision and Image Understanding 78 (2000) 157–176.

[8] R. Rosales, V. Athitsos, L. Sigal, S. Sclaroff, 3d hand pose reconstruction using specialized mappings, in: International Conference on Computer Vision, pp. 378–385.

[9] K. A. Barhate, K. S. Patwardhan, S. D. Roy, S. Chaudhuri, S.Chaudhury, robust shape based two hand tracker, in: IEEE International Conference on Image Processing, pp. 1017–1020.

[10] A. Argyros, M. Lourakis, Tracking multiple colored blobs with a moving camera, in: IEEE Conference on Computer Vision and Pattern Recognition, p. 1178.

[11] X. Wang, X. Zhang, G. Dai, Tracking of deformable human hand in real time as continuous input for gesture-based interaction, in: International Conference on Intelligent User Interfaces, pp. 235–242.

[12] R. Rosales, S. Sclaroff, Inferring body pose without tracking body parts, in: International Conference on Computer Vision and Pattern Recognition, volume II, pp. 721–727.

[13] P. F. Felzenszwalb, D. P. Huttenlocher, Pictorial structures for object recognition, Int. J. Comput. Vision 61 (2005) 55–79.

[14] P. F. Felzenszwalb, D. P. Huttenlocher, Efficient matching of pictorial structures, in: Proc. IEEE Computer Vision and Pattern Recognition Conf., pp. 66–73.

[15] P. Dhawale, M. Masoodian, B. Rogers, Bare-hand 3D gesture input to interactive systems, in: 7th international conference on Computer-human interaction: design centered HCI, pp. 25–32.

[16] J. Y. Lin, Y. Wu, T. S. Huang, 3D model-based hand tracking using stochastic direct search method, in: International Conference on Automatic Face and Gesture Recognition, p. 693.

[17] Y. Wu, J. Y. Lin, T. S. Huang, Capturing natural hand articulation, in: International Conference on Computer Vision, volume 2, pp. 426–432.

[18] M. Kato, Y.-W. Chen, G. Xu, Articulated hand tracking by pca-ica approach, in: International Conference on Automatic Face and Gesture Recognition, pp. 329–334.

[19] H. Ouhaddi, P. Horain, 3D hand gesture tracking by model registration, in: Workshop on Synthetic-Natural Hybrid Coding and Three Dimensional Imaging, pp. 70–73.

[20] K. Nirei, H. Saito, M. Mochimaru, S. Ozawa, Human hand tracking from binocular image sequences, in: 22th International Conference on Industrial Electronics, Control, and Instrumentation, pp. 297–302.

[21] A. Amai, N. Shimada, Y. Shirai, 3-d hand posture recognition by training contour variation, in: IEEE Conference on Automatic Face and Gesture Recognition, pp. 895–900.

[22] N. Shimada, K. Kimura, Y. Shirai, 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.

[23] H. Zhou, T. Huang, Okapi-chamfer matching for articulated object recognition, in: IEEE International Conference on Computer Vision, volume 2, pp. 1026–1033.

[24] D. G. Lowe, Object recognition from local scale-invariant features, in: IEEE International Conference on Computer Vision, volume 2, pp. 1150–1157.

[25] H. Zhou, T. Huang, Tracking articulated hand motion with eigen dynamics analysis, in: IEEE International Conference on Computer Vision, volume 2, pp. 1102–1109.

[26] B. Stenger, A. Thayananthan, P. H. S. Torr, R. Cipolla, Model-based hand tracking using a hierarchical bayesian filter, in: IEEE Transactions on Pattern Analysis and Machine Intelligence, volume 28, pp. 1372–1384.

[27] B. D. R. Stenger, Model-based hand tracking using a hierarchical bayesian filter, in: Dissertation submitted to the University of Cambridge.

[28] E. B. Sudderth, M. I. Mandel, W. T. Freeman, A. S. Willsky, Visual hand tracking using nonparametric belief propagation, in: IEEE CVPR Workshop on Generative Model Based Vision, volume 12, p. 189.

[29] D. Mohr, G. Zachmann, Fast: Fast adaptive silhouette area based template matching, in: Proceedings of the British Machine Vision Conference, BMVA Press, 2010, pp. 39.1–39.12. Doi:10.5244/C.24.39.
[30] M. Piccardi, Background subtraction techniques: a review, in: Systems, Man and Cybernetics, 2004 IEEE International Conference on, volume 4, pp. 3099 – 3104 vol.4.
[31] P. Kakumanu, S. Makrogiannis, N. Bourbakis, A survey of skin-color modeling and detection methods, Pattern Recogn. 40 (2007) 1106–1122.
[32] D. Mohr, G. Zachmann, Segmentation-free, area-based articulated object tracking, in: G. Bebis, R. Boyle, B. Parvin, D. Koracin, S. Wang, K. Kyungnam, B. Benes, K. Moreland, C. Borst, S. DiVerdi, C. Yi-Jen, J. Ming (Eds.), 7th International Symposium on Visual Computing, volume 6938 of *Lecture Notes in Computer Science*, Springer, 2011, pp. 112–123.
[33] R. Y. Wang, J. Popović, Real-time hand-tracking with a color glove, in: ACM SIGGRAPH 2009 papers, SIGGRAPH '09, ACM, New York, NY, USA, 2009, pp. 63:1–63:8.
[34] D. Huttenlocher, G. A. Klanderman, W. J. Rucklidge, Comparing images using the hausdorff distance, in: IEEE Transactions on Pattern Analysis and Machine Intelligence.
[35] H. G. Barrow, J. M. Tenenbaum, R. C. Bolles, H. C. Wolf, Parametric correspondence and chamfer matching: Two new techniques for image matching, in: International Joint Conference on Artificial Intelligence.
[36] G. Borgefors, Hierarchical chamfer matching: A parametric edge matching algorithm, in: IEEE Transaction on Pattern Analysis and Machine Intelligence.
[37] V. Athitsos, S. Sclaroff, 3D hand pose estimation by finding appearance-based matches in a large database of training views, in: IEEE Workshop on Cues in Communication.
[38] V. Athitsos, S. Sclaroff, An appearance-based framework for 3d hand shape classification and camera viewpoint estimation, in: IEEE Conference on Automatic Face and Gesture Recognition.
[39] V. Athitsos, J. Alon, S. Sclaroff, G. Kollios, Boostmap: A method for efficient approximate similarity rankings, in: IEEE Conference on Computer Vision and Pattern Recognition.
[40] D. Gavrila, V. Philomin, Real-time object detection for "smart" vehicles, volume 1, IEEE Computer Society, Los Alamitos, CA, USA, 1999, p. 87.
[41] Z. Lin, L. S. Davis, D. Doermann, D. DeMenthon, Hierarchical part-template matching for human detection and segmentation, in: IEEE International Conference on Computer Vision.
[42] A. Thayananthan, R. Navaratnam, B. Stenger, P. Torr, R. Cipolla, Multivariate relevance vector machines for tracking, in: European Conference on Computer Vision.
[43] C. F. Olson, D. P. Huttenlocher, Automatic target recognition by matching oriented edge pixels, in: IEEE Transactions on Image Processing.
[44] G. Shaknarovich, P. Viola, T. Darrell, Fast pose estimation with parameter-sensitive hashing, in: IEEE International Conference on Computer Vision.
[45] V. Athitsos, S. Sclaroff, Estimating 3D hand pose from a cluttered image, in: IEEE Conference on Computer Vision and Pattern Recognition.
[46] D. Mohr, G. Zachmann, Continuous edge gradient-based template matching for articulated objects, in: International Joint Conference on Computer Vision and Computer Graphics Theory and Applications.
[47] S. Lu, D. Metaxas, D. Samaras, J. Oliensis, Using multiple cues for hand tracking and model refinement, in: IEEE Conference on Computer Vision and Pattern Recognition, pp. 443–450.
[48] N. Petersen, D. Stricker, Fast hand detection using posture invariant constraints, in: B. Mertsching (Ed.), KI 2009: Advances in Artificial Intelligence. German Conference on Artificial Intelligence (KI-2009), 32nd Annual Conference on Artificial Intelligence, September 15-18, Paderborn, Germany, Lecture Notes in Artificial Intelligence, Springer, Berlin, 2009.
[49] M. de La Gorce, N. Paragios, D. J. Fleet, Model-based hand tracking with texture, shading and self-occlusions, in: IEEE Conference on Computer Vision and Pattern Recognition.
[50] L. Ballan, A. Taneja, J. Gall, L. V. Gool, M. Pollefeys, Motion capture of hands in action using discriminative salient points, in: European Conference on Computer Vision (ECCV), Firenze.
[51] M. de La Gorce, N. Paragios, A variational approach to monocular hand-pose estimation, Computter Vision and Image Understanding 114 (2010) 363–372.
[52] I. Oikonomidis, N. Kyriazis, A. Argyros, Full dof tracking of a hand interacting with an object by modeling occlusions and physical constraints, in: ICCV 2011, IEEE, 2011.
[53] J. Wang, S. Kumar, S. Chang, Semi-supervised hashing for scalable image retrieval, in: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, IEEE, pp. 3424–3431.
[54] B. Kulis, T. Darrell, Learning to hash with binary reconstructive embeddings, in: In Proc. NIPS, pp. 1042–1050.
[55] B. Kulis, K. Grauman, Kernelized locality-sensitive hashing for scalable image search, in: IEEE International Conference on Computer Vision (ICCV.
[56] A. Torralba, R. Fergus, Y. Weiss, Small codes and large image databases for recognition, in: In Proceedings of the IEEE Conf on Computer Vision and Pattern Recognition.
[57] V. Athitsos, M. Potamias, P. Papapetrou, G. Kollios, Nearest neighbor retrieval using distance-based hashing, in: Proceedings of the 2008 IEEE 24th International Conference on Data Engineering, ICDE '08, IEEE Computer Society, Washington, DC, USA, 2008, pp. 327–336.
[58] V. Athitsos, J. Alon, S. Sclaroff, G. Kollios, BoostMap: An Embedding Method for Efficient Nearest Neighbor Retrieval, Pattern Analysis and Machine Intelligence, IEEE Transactions on 30 (2008) 89–104.
[59] A. Agarwal, B. Triggs, 3D human pose from silhouettes by relevance vector regression, in: International Conference on

Computer Vision & Pattern Recognition, CVPR 2004, June, 2004, volume 2, IEEE, Washington, DC, Etats-Unis, 2004, pp. 882–888.
[60] D. M. Gavrila, L. S. Davis, 3-d model-based tracking of humans in action: a multi-view approach, in: Conference on Computer Vision and Pattern Recognition, pp. 73–80.
[61] D. Gavrila, Pedestrian detection from a moving vehicle, in: Proceedings of the 6th European Conference on Computer Vision-Part II, ECCV '00, Springer-Verlag, London, UK, UK, 2000, pp. 37–49.
[62] C. Tomasi, S. Petrov, A. Sastry, 3d tracking = classification + interpolation, in: International Conference on Computer Vision, pp. 1441–1448.
[63] P. Viola, M. Jones, Rapid object detection using a boosted cascade of simple features, in: IEEE Conference on Computer Vision and Pattern Recognition, volume 1, pp. I–511–I–518.
[64] B. Heisele, T. Serre, S. Mukherjee, T. Poggio, Feature reduction and hierarchy of classifiers for fast object detection in video images, in: Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, volume 2, pp. II–18 – II–24.
[65] T. Gumpp, P. Azad, K. Welke, E. Oztop, R. Dillmann, G. Cheng, Unconstrained real-time markerless hand tracking for humanoid interaction, in: IEEE Humanoids, pp. 88–93.
[66] Q. Delamarre, O. Faugeras, Finding pose of hand in video images: a stereo-based approach, in: IEEE International conference on Automatic Face and Gesture Recognition, pp. 585–590.
[67] S. A. Gudmundsson, J. R. Sveinsson, M. Pardas, H. Aanaes, R. Larsen, Model-based hand gesture tracking in ToF image sequences, in: 6th International Conference of Articulated Motion and Deformable Objects, pp. 118–127.
[68] G. Hackenberg, R. McCall, W. Broll, Lightweight palm and finger tracking for real-time 3D gesture control, in: IEEE Virtual Reality Conference, pp. 19–26.
[69] I. Oikonomidis, N. Kyriazis, A. Argyros, Efficient model-based 3d tracking of hand articulations using kinect, in: BMVC 2011, BMVA, 2011.
[70] I. Oikonomidis, N. Kyriazis, A. Argyros, Tracking the articulated motion of two strongly interacting hands, in: CVPR 2012, IEEE. To appear.
[71] C. Keskin, F. K$#305;ra$#231;, Y. E. Kara, L. Akarun, 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, Springer-Verlag, Berlin, Heidelberg, 2012, pp. 852–863.
[72] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, A. Blake, Real-Time Human Pose Recognition in Parts from Single Depth Images.
[73] V. Lepetit, P. Lagger, P. Fua, Randomized trees for real-time keypoint recognition, in: Computer Vision and Pattern Recognition, pp. 775–781.
[74] C. John, 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.
[75] E. Ueda, Y. Matsumoto, M. Imai, T. Ogasawara, A hand-pose estimation for vision-based human interfaces, in: IEEE Transactions on Industrial Electronics, pp. 676–684.
[76] M. Schlattmann, R. Klein, Simultaneous 4 gestures 6 dof real-time two-hand tracking without any markers, in: ACM Symposium on Virtual Reality Software and Technology (VRST '07).
[77] M. Schlattmann, F. Kahlesz, R. Sarlette, R. Klein, Markerless 4 gestures 6 dof real-time visual tracking of the human hand with automatic initialization, Computer Graphics Forum 26 (2007) 467–476.