Real-Time Hand Tracking for Natural and Direct Interaction

Daniel Mohr
Clausthal University
dmoh@tu-clausthal.de

Gabriel Zachmann
Clausthal University
zach@tu-clausthal.de

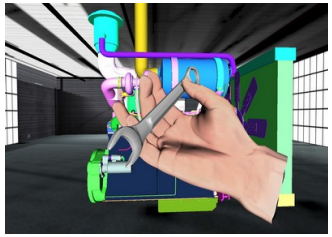

Figure 1: A hand tracking application: In virtual assembly simulation an engineer can manipulate virtual objects without any intrusive device.

CHI 2010, April 10–15, 2010. Atlanta, Georia, USA
ACM 978-1-60558-930-5/10/04.

Abstract

Hand tracking is a useful technique for interaction in many applications, for example for navigation in virtual environments, virtual assembly simulation, gesture recognition, and motion capture. Therefore, our goal is to track the global position and all finger joint angles of a human hand in real-time.
Due to measurement noise, occlusion, cluttered background, inappropriate illumination, high dimensionality, and real-time constraints, hand-tracking is a very important and interesting scientific challenge.
We capture images of the hand from different directions with multiple cameras. We use a model based approach to track the hand. As matching features, we use skin segmentation combined with a silhouette area comparison and an edge-gradient based similarity measure. We utilize dimension reduction techniques to cope with the high complexity of the tracking problem (the hand has about 20 local DOFs and 6 global DOFs).

Keywords

pose estimation, template matching, silhouette area similarity, edge feature

ACM Classification Keywords

H.5.2 Information Interfaces and Presentation: User interfaces – Evaluation/ methodology - Input devices and strategies

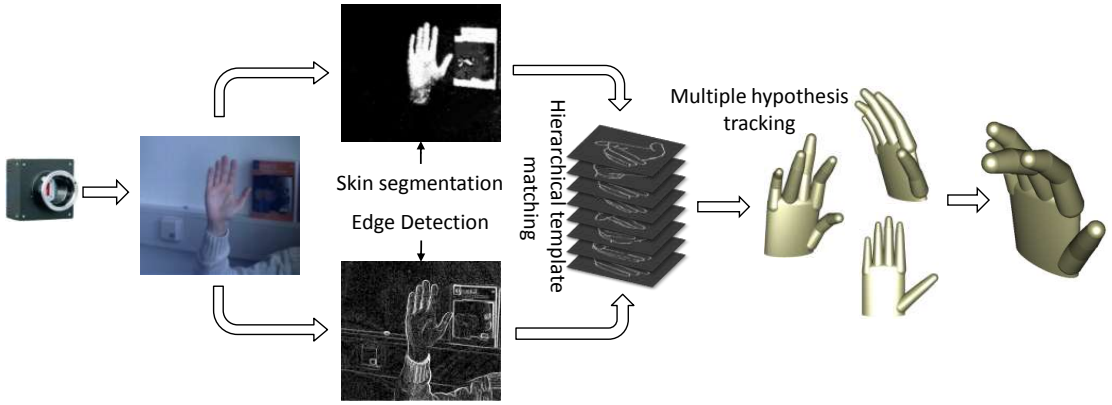

Figure 2: Overview of our hand-tracking approach. The hand is capured by a industrial camera, features are extracted, the most likely hand parameters computed, and finally one parameter set chosen as result.

I.4.8 Image Processing and Computer Vision: Tracking - Object Recognition

Introduction

The most important "tool" of our body for interaction are our hands. They are needed in nearly all human actions from simple gestures up to more complex tasks such as controlling a computer, manipulating real and virtual objects, and navigating in virtual environment.

Consequently, hand-tracking could become a great enrichment in many fields e.g. HCI, VR, AR. In some applications it has already been used successfully. In some HCI applications one can achieve impressive results even by tracking a small subset of the hand's degree of freedom (DOF). For example an algorithm that is able to classify a few gestures can be used to understand sign languages or be used as input device for a computer to replace a mouse [2, 6]. Here, the 2D hand position in the image plane replaces the mouse movement and the hand gestures the mouse buttons. One can also use hand-tracking to enrich a presentation on a wall by directly moving or adding virtual items in the presentation [9].

Even for simple 2D interactions, hand tracking can replace device-based interactions. For instance, tracking two fingertips can replace touchpads [3, 4]. A keyboard could be replaced by a virtual keyboard by tracking the fingertips of both hands.

Of course, exploiting the full DOF tracking, one could use the hand for dextrous manipulation and interaction. The hand could become a much more powerful input device then common input devices (e.g. mouse or touch screens).

In virtual assembly simulation, an engineer interacts with his CAD application in a virtual environment [5] (e.g. in a cave). Using hand tracking, he could navigate very intuitive by freely moving his hands, control and handle its CAD application without additional input devices, and manipulate the objects to be desiged in a natural way. A grasping movement,

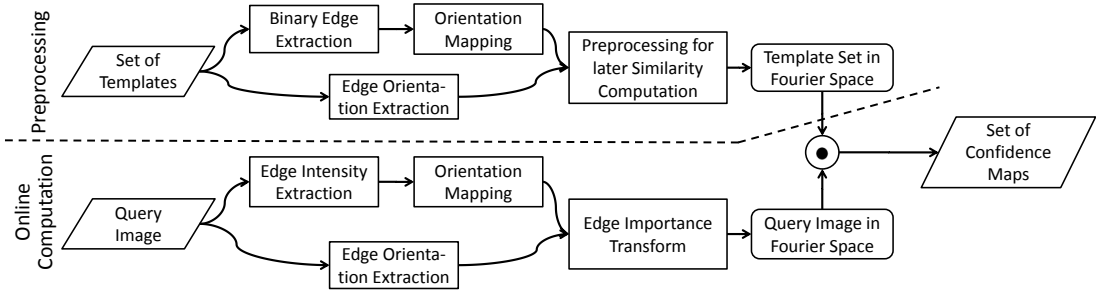

Figure 3: Overview of our edge-gradient based similarity measure.

for example, to open a door of a car, is the same action as in the real world, in contrast to traditional interaction through a mouse, where it has to be simulated by a sequence of mouse clicks.

A completely different application of hand tracking are video and computer games. Imagine a shooter, where the user uses his hand to focus his target. This is much more intuitive then using a mouse, because he could easily control the 6 DOF (translation and rotation) needed in 3D. In an adventure game, the user could pick up and drop objects in an intuitive way in contrast to using keyboard shortcuts. In a flight simulator, the translational DOF can be used to control the acceleration, the rotational DOF to modify the pitch, roll, and heading angles of the airplane.

These are only a few of the numerous applications of hand tracking. Most of them need, obviously, real-time, precise, tracking of the hand with 26 DOFs. So, algorithms to achieve this are an enabling technology for this kind of interaction paradigm. The focus of the remainder of this paper is, therefore, to describe our approach to provide such a technology.

Our approach

An important initial step in object tracking is to localize the object in the 2D image delivered by the camera. This is a challenging task especially with articulated objects, due to the huge state space and, possibly, time constraints. Most approaches formulate tracking of articulated objects as detecting multiple objects: given a database of many objects, find the object from the database that best matches the object shown in the input image. This also involves finding the location in the input image where that best match occurs. Each of the objects in the database represents the articulated object in a different state and viewpoint. Typically, the database consists of images, called *templates*, which are possibly preprocessed. This can result in a database size of thousands of templates. Thus, tracking of articulated objects can be reformulated as template matching.

We generate our templates by an artificial 3D hand model. This model can be rendered in any desired state, and it can be easily projected onto 2D and extract nearly perfect edge-gradients or binarized to get the hand silhouette.

For template matching we mainly use to features: edge-gradients and hand silhouette are similarity. The hand sil-

houette in the input image is extracted by skin segmentation.

Edge based template matching

The shapes of most articulated objects is very characteristic and does not appear in the background. Therefore, a powerful method to match templates is to compare the edge images of template and input image. However, edges are no silver bullet because the quality of the edge images is negatively influenced by various factors such as scene illumination, camera parameters, object and background color, shadows.
To overcome this problems, we proposed a novel method for template matching based on edge features, which is robust against varying edge response. To this end, we proposed a novel similarity measure between a template and the query image that utilizes the continuous edge gradient (orientation *and* intensity). The input to our algorithm is a query image and a set of templates. The output is a *confidence map*. It stores for each position in the query image the index and similarity of the best matching template.
The *main contributions* are:

1. Our method does *not perform any binarization* or discretization during the online matching process. By contrast, all current methods based on edge distance/similarity need binary edge images. This incurs thresholds that are difficult to adjust automatically, which reduces the robustness of these approaches.

2. We utilize the *orientation and intensity* of edges of both the templates and the query images directly in our similarity measure. By contrast, most current methods discretize edge orientations into a few intervals, which renders the similarity measure discontinuous with respect to rotation of the object.

3. Our method is well suited for a complete implementation in the *stream processing model* (e.g., on modern GPUs), which allows for extremely fast template matching.

In subsequent steps, the *confidence map* is combined with other confidence maps using different features, e.g. silhouette are matching, presented in the following.

Silhouette area comparison

We developed a novel method for very fast approximate area silhouette comparison between model templates and the segmented input image. For one template comparison, Stenger et al. [7] achieved a computation time proportional to the contour length of the template silhouette. We propose a new method, which reduces the computation time to be *constant* in the contour length and image resolution. To achieve this, we first approximate all template silhouettes by axis-aligned rectangles, which is done in a preprocessing step. In the online phase, we compute the integral image [1, 8] of the segmented image. With this, the joint probability of a rectangle to match to an image region can be computed by four lookups in the integral image.
Moreover, we developed an algorithm to build a template hierarchy that can compare a large set of templates in sublinear time.
The *main contributions* are:

1. An algorithm that approximates arbitrary shapes by a minimal set of axis-aligned rectangles. This results in a resolution-independent, very memory efficient silhouette area representation.

2. An algorithm to compare an object silhouette in *O(1)* in contrast to [7], which is in *O(*contour length*)*.

3. We propose an algorithm to cluster templates hierarchically guided by their mutually overlapping areas. Our

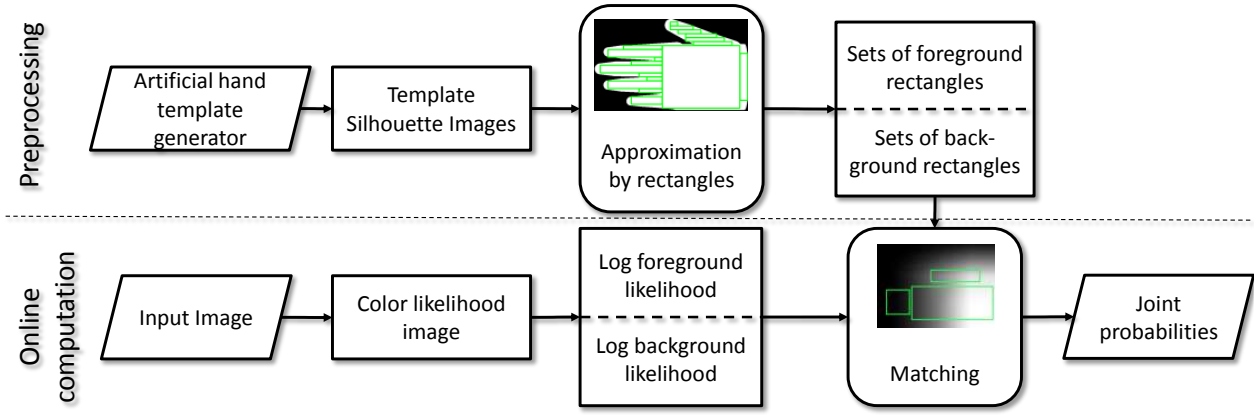

Figure 4: Overview of our silhouette area based similarity measure. We use a rectangle sets to approximate a silhouette. This speeds up the matching by a factor 5–30 compared to a state-of-the-art approach.

method builds on the recently developed batch neural gas clustering algorithm, which yields for better results than more classical algorithms. areas. This hierarchy further reduces the matching complexity for n templates from $O(n)$ to $O(\log n)$.

Our approach only requires that binary silhouettes of the model in an arbitrary pose can be generated and that the input image can be segmented. The segmentation result does not necessarily need to be binarized. The approach can handle scalar segmentations as well.

We use a skin segmentation algorithm that computes for each image pixel the probability to represent skin or background, resp. We use the joint probability as proposed by Stenger et al. [7] to compare the silhouettes with the segmentation result. A simple area overlap, of course, could be used, too. The only difference is that the sum instead of the product of probabilities whould have be computed.

Conclusions and future work

In the area of hand tracking, the most reliable features are edges and skin color. We have proposed efficient template matching approaches incorporating these two features. We are currently working on an approach to build a high quality hierarchy based on the edge features and integrate this hierarchy as well as the silhouette based hierarchy into a random forest framework to improve the robust of our tracker. We further want to improve the robustness by exploiting time coherences.

References

[1] F. C. Crow. Summed-area tables for texture mapping. In *SIGGRAPH: Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, 1984.

[2] J. O. Kim. A hand tracking for a human computer interac-

tion system b y the modified block matching algorithm. In *International Conference on Computational Science*, 2003.

[3] S. Malik and J. Laszlo. Visual touchpad: A two-handed gestural input device. In *6th International Conference on Multimodal Interfaces*, 2004.

[4] Z. Mo, J. P. Lewis, and U. Neumann. Smartcanvas: A gesture-driven intelligent drawing desk system. In *10th International Conference on Intelligent User Interfaces*, 2005.

[5] R. O'Hagan, A. Zelinsky, and S. Rougeaux. Visual gesture interfaces for virtual environments. In *Interacting with Computers 14*, 2002.

[6] J. Rehg and T. Kanade. Digiteyes: Vision-based hand tracking for human-computer interaction. In *Workshop on Motion of Non-Rigid and Articulated Bodies*, 1994.

[7] B. Stenger, A. Thayananthan, P. H. S. Torr, and R. Cipolla. Model-based hand tracking using a hierarchical bayesian filter. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2006.

[8] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2001.

[9] C. von Hardenberg and F. Berard. Bare-hand human-computer interaction. In *Workshop on Perceptive User Interfaces*, 2001.