Gesture Interfaces with Depth Sensors

Foti Coleca[1,2], Thomas Martinetz[1], and Erhardt Barth[1]

[1] Institute for Neuro- and Bioinformatics, University of Lübeck,
160 Ratzeburger Allee, 23562, Lübeck, Germany
{coleca,martinetz,barth}@inb.uni-luebeck.de
[2] gestigon GmbH, Maria-Goeppert Straße 1, 23562 Lübeck, Germany

Abstract. Computers and other electronic devices shrink and the need for a human interface remains. This generates a tremendous interest in alternative interfaces such as touch-less gesture interfaces, which can create a large, generic interface with a small piece of hardware. However, the acceptance of novel interfaces is hard to predict and may challenge the required computer-vision algorithms in terms of robustness, latency, precision, and the complexity of the problems involved.

In this article, we provide an overview of current gesture interfaces that are based on depth sensors. The focus is on the algorithms and systems that operate in the near range and can recognize hand gestures of increasing complexity, from simple wipes to the tracking of a full hand-skeleton.

1 Introduction

In this chapter we focus on gestural interfaces, specifically close-range applications using a depth camera.

Gesture interfaces are different from the input devices currently in use, and for them to be successful, they must be designed from the ground up, with natural human interaction in mind. For this purpose, we first present a gesture taxonomy.

In Section 2 we show how depth cameras affected the field of gesture interaction and algorithmic approaches to hand pose estimation. We then guide the reader through the state of the art. Thereby, related hardware issues are presented only briefly, the focus being the algorithmic approaches. While the main discussion is about solutions which use depth sensors, we also give a brief overview of methods that are using 2D cameras.

The next section is dedicated to identifying remaining challenges, from hardware shortcomings to environment and ergonomic limitations, also proposing solutions to some of these limitations.

Section 4 follows recent developments in hardware, commercial solutions, as well as our own work in the field. We first give an overview on pose estimation using self-organizing maps and then present a few recent extensions.

Finally, Section 5 provides some example applications of gestural interfaces, showing the wide variety of fields that can benefit from this technology.

M. Grzegorzek et al. (Eds.): Time-of-Flight and Depth Imaging, LNCS 8200, pp. 207–227, 2013.

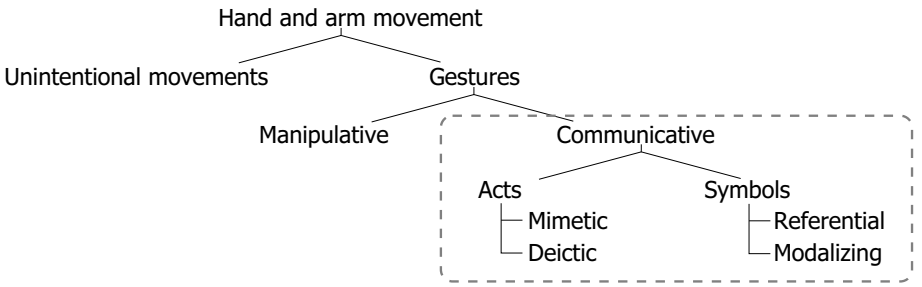


Fig. 1. Hand and arm movement types, as shown in [1]. Communicative gestures are the main focus of touch-less gestural interfaces.

1.1 Gesture Taxonomies

Humans use gestures in everyday life to communicate and interact with the environment. It is not obvious what gestures are and how they can be used to build a better interface. As a brief introduction to the topic, we summarize, in Fig. 1, the hand and arm movement taxonomy:

Unintentional movements are movements unrelated to, and not serving a meaningful communication purpose. These are dependent entirely on the context of the situation, as the same gesture that can be used to communicate something in a certain situation might be completely unintentional in another.

Gestures are hand and arm movements done with the specific intention of communication. Gestures specifically made during verbal communication between humans are known as *gesticulations*. They are first separated by their physicality, as *manipulative* and *communicative* gestures.

Manipulative gestures are used to physically act upon objects in an environment, and depend on the type of action being done on the objects themselves. In the context of human-computer interaction, these are found in interfaces where a direct physical contact is required to use them (e.g. touch-based interfaces).

Communicative gestures have a communicational purpose and are used together with, or instead of, natural speech. Communicative gestures are the focus of touch-less gestural interfaces. Depending on the situation, any part of the body can be used to generate them. They can bring a richer means of interaction, at the cost of being harder to detect and classify.

Acts relate directly to the intended interpretation, are transparent, and can be understood without prior learning. They can either be *mimetic* imitating actions or objects or *deictic*, pointing gestures, which are further split into *specific*, *generic*, and *metonymic* (when pointing at an object to signify some entity related to it). Deictic gestures are useful in simple interfaces, as pointing is a natural way of communicating intention. Example applications are controlling a slideshow [2] or even robots [3].

Symbols are motion short-hand that cannot be used without prior learning. They can vary greatly between cultures and are deeply rooted in the human

interpersonal communication. Symbols are useful for gesture interfaces as humans are adept at learning novel ones, which can be created specifically to control said interface. The two categories of symbols are *referential* and *modalizing*. The former refer to iconic gestures linked directly to meanings (e.g. the thumbs-up gesture, rubbing fingers and thumb together to symbolize money), while the latter are used to change the meaning (mode) of communication, (eg. shrugging shoulders to indicate uncertainty, which would not be apparent if one would only read a transcript of the conversation).

2 State of the Art

2.1 Time-of-Flight Sensors and Alternative Hardware

Time-of-flight (ToF) sensors have led to the first compact 3D cameras that could deliver depth maps at video rate [4]. Early work with 3D cameras was based on either the Swissranger cameras [5,6,7,8,9,10], the PMD sensors [11,12] or the Canesta cameras [13], which were all using the same principle of light modulation and phase measurement. Alternatively, some authors were using the 3DV Zcam [14], which used pulses, and was one of the early compact 3D devices, but was not widely available. With the introduction of the low-cost Microsoft Kinect, the field has expanded quickly [15,16,17,18,19,20,21,22]. Limitations of ToF cameras and open issues are discussed in Section 3.1 as well as in Chapters 1 and 2 of this book.

With stereo-based approaches it is difficult to obtain a dense range map. This issue is hard to overcome because stereo disparities can only be estimated at those locations which have a distinct image structure, and it is known that such image patches are rare in natural images [23]. A further limitation is size, because miniaturization is limited by the need to have a sufficient baseline. We have performed extensive tests with different stereo cameras and different illumination settings, and have always obtained range maps that cannot properly resolve the fingers of a hand.

3D cameras that use structured light also require a baseline and two optical systems, for the camera and the projector. Moreover, insensitivity to ambient light is more difficult to achieve. Limitations are discussed in Section 3.1.

2.2 Algorithmic Approaches

Gesture interfaces can range from simple motion detection to complex, pose-driven gesture recognition. In this section, we will focus on hand-pose estimation for gesture recognition. Although there exist a variety of methods to capture the pose of the hand, most can be categorized using combinations of the following dichotomies (Fig. 2):

Partial methods estimate the locations of specific features of the hand. These include approaches from simple geometry and motion parameter extraction of the hand image such as blob tracking and averaging (hand center) to fingertip detection and tracking.

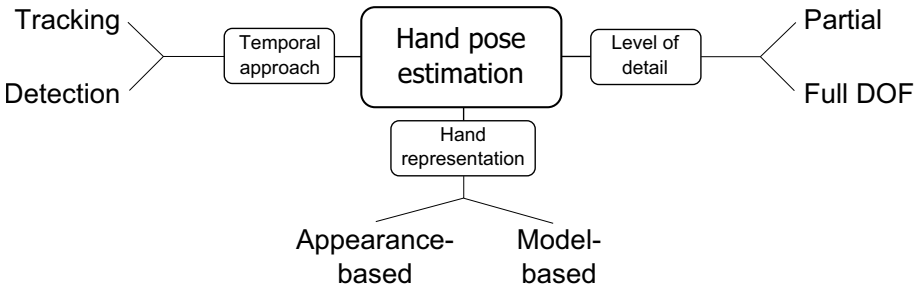


Fig. 2. Pose estimation dichotomies

Full DOF (degree-of-freedom) methods attempt to extract all the kinematic parameters of the hand pose such as fingertips, joint positions, hand orientation, finger angles etc. This is usually done with a full hand skeleton, via a model-based approach.

Appearance-based approaches try to infer gestures directly from the appearance of the hand. These methods are used frequently with 2D cameras, as they are based on a series of 2D views of the 3D object.

Model-based methods estimate the hand position and the specific angles of the joints using a model or skeleton. The model can vary greatly in complexity, from using simple geometric primitives to model a hand skeleton, to accurate computer renderings of hand meshes. Usually these methods attempt to recover the full degree of freedom of the hand.

Tracking approaches use the previous discovered parameters of the hand pose to predict the new ones. This approach is used extensively in methods that need to search over a large state space for the parameters which best match the current hand configuration (i.e. model-based approaches). Using prior information, the search can be restricted only to the most probable hand configurations.

Detection methods disregard temporal information and attempt a single-shot pose estimation. This is sometimes preferred, as the hand and fingers are capable of rapid motion, making time coherence assumptions useless [24].

Methods are often combined to balance their strengths and weaknesses. For example if only tracking is used the tracker may drift away, and when only detection is used, the hand pose can be unstable.

Hand pose estimation is a particularly difficult problem, which poses a number of challenges:

Size : compared to the human body (also used in gesture interfaces), the hand is significantly smaller, with, complex articulated fingers, which are easily affected by segmentation errors [20].

High dimensionality : human hand models used for pose estimation usually have around 26 degrees of freedom [24], the state space being very large.

Skin : The hand is chromatically uniform, which poses a problem for finger detection using color, especially in complex poses. The skin color is also heavily dependent on scene illumination, if skin segmentation is used for hand detection.

Severe self-occlusions : Due to the complexity of the hand, fingers often occlude each other while gesturing. Trying to bypass this problem by forcing the user into non-self-occluding poses (such as an keeping the hand parallel to the interface) makes for an unnatural interface experience and should therefore be avoided.

Performance : Real-world interfaces need short response times in order to be usable. With the ever-increasing computing power available and the introduction of 3D cameras, which simplify tasks like scene segmentation, real-time performance is no longer an unattainable goal.

2.2.1 Using 2D Cameras

The progress made in the early day of gesture interfaces and the limitations of the early approaches are comprehensively reviewed in [1]. The authors conclude that "Although the current progress is encouraging, further theoretical as well as computational advances are needed before gestures can be widely used for HCI" (Human-Computer Interaction). The review emphasizes the popular distinction between model-based and appearance-based approaches and separates between volumetric and skeletal hand models. Regarding applications, a distinction is made between manipulative and communicative gestures. We may conclude that many of the conceptual issues had been clarified early but still, we had to wait for many years until a more mature sensing technology and a few new algorithmic ideas have brought the field much closer to real applications.

Due to limitations of the computing hardware and the lack of depth sensors, early approaches often relied on detecting the hands using a color skin model [25]. Only a few approaches have been developed into systems that would work reliably under a variety of conditions, as for instance [25]. Here, 2D-color-blobs associated with the hands and the head are tracked based on a Maximum A Posteriori Probability approach. In [26] hidden Markov models (HMM) were used for the recognition of 18 different Tai Chi gestures. Different features extracted from a stereo-camera system that could track the head and the hands were evaluated; typical recognition rates were around 90 percent correct. Using more than two RGB cameras can enhance the performance of 2D-camera based hand-pose estimation. This approach is used in [27] with no less than 8 cameras, which allows for the pose capture of two strongly interacting hands and an additional object. These methods usually aggravate the problem of computational overload, which can be then dealt with by using GPUs instead of CPUs [28].

2.2.2 Using 3D Cameras

The approach for body-skeleton tracking developed by Shotton et al. [29] was extended to the hand pose in [15]. The authors claim real-time performance but do not show hand poses for real-life data. Another popular approach is to detect

fingertips and use the positions directly as input to the gesture interface [11,30]. In one of the first approaches that used depth data for hand pose estimation [31], the authors employed an active, structured-light stereo system to detect the fingertips with a combination of skin segmentation and 3D principal curvature analysis. The detected fingertips and hand position and orientation were subsequently used for a coarse model of the hand, achieving real-time detection of static or dynamic gestures. A model-based approach is used also in [32], where the hand direction is first coarsely estimated using principal component analysis (PCA), after which a model fitting is able to estimate 7 degrees of freedom of the hand.

When moving from RGB cameras to 3D cameras, the issue of choosing appropriate representations or features had to be readdressed [4,5,14,33]. Even when using standard methods such as the PCA on 3D data, the interpretation of the main axes may differ in 3D [14].

Segmenting objects by their distance from the camera is often a better way of recognizing the hand compared to color segmentation, for example, by assuming it to be the *closest object* to the camera [34], especially in cases where multiple people are in the frame or there is a partial hand-face occlusion [35]. Still, some approaches [35,36] use skin color for hand detection, mainly for enhancing depth-based segmentation. While the authors of [35] do not report a significant increase in performance, there are certain situations where a combination of skin and depth for hand segmentation may be useful, for instance the former example would be enhanced by assuming the hand to be the *closest skin colored object*, which would exclude other objects close to the camera, such as a keyboard. It would also provide a better hand segmentation for users that wear long-sleeved shirts and salvage cases where depth segmentation is prone to errors, such as the hand being too close to another object.

When using ToF sensors, quite a few authors have stressed the importance of fusing the 2D and 3D data (the intensity and the range maps) [5,7,8,9,10,37,38]. In [38], for example, the recognition rates for a set of simple arm gestures were between 78% and 88% correct when using only the 3D data, while with the fused 3D and 2D data the rates improved the rates to between 90% and 95%. The authors of [38], also argue for representations of gestures as a sequence of discrete primitives as opposed to recognizing gestures through a trajectory based approach. Their approach is further developed in [10] by including optical flow for better motion estimation. Another approach is [21], which uses two Kinects and two RGB cameras to capture a wider 3D scene, which improves the robustness of hand tracking, while the high definition web cams determine the hand pose. As well as fusing data from depth and RGB images, the authors of [36] use angular data from an inertial measurement unit to normalize and orient the hand upwards for pose estimation.

Hand pose estimation for sign language recognition is also a very active field. The authors of [39] use a combination of three letter classifiers to detect words from sequences of hand gestures. As a novel feature, the letter classifiers are improved by updating the training samples when a word is detected with high

confidence. For an extended overview of sign language recognition systems we direct the reader to Chapter 4.2 of this book.

The bag of visual-and-depth-words approach is used in [40] in conjunction with a concatenated Viewpoint Feature Histogram (VFH) and Camera Roll Histogram (CRH) feature vector. Spatio-temporal pyramids are used to fuse geometrical and temporal information. With the addition of late fusion of the RGB (Histogram of Oriented Gradients, Histogram of Optical Flow) and depth (VFH-CRH) descriptors, the mean Levenshtein distance between the recognized sequence of gestures and the ground truth is improved from 0.30 to 0.26.

The authors of [17] achieve a 87% hand gesture recognition accuracy with a multi-step approach, finding hand-sized blobs, performing scale and rotation normalization, then extracting four feature descriptors and classifying gestures using an action graph as an alternative to HMMs. In [18], a 26 DOF hand model is matched to the hand pose using particle swarm optimization. The GPU is then used to accelerate the implementation to near real-time frame rates (15 Hz). Model-based pose detection is also used in [19]: a one-shot pose estimation is done using a hand pose database consisting of 20 prototype models (poses) rendered from 86 different viewpoints. The images from the database are compared to the actual segmented hand pose by means of a weighted depth matching and chamfer distance similarity measure. In tests, the authors discovered that anthropometric features varied greatly between users' hands and that the real-world 3D data could not be aligned perfectly to the generated poses. They obtain a recognition rate of 76% for a 1–64 pixels error between the winner pose and the real-world 3D pose. The authors of [20] achieve a recognition rate of 90% and a runtime of 0.5 s per pose, with a method based on Earth-Mover's distance. This method is also robust to finger-melding poses, when two fingers are close enough, or partially occluding each other, to be considered to be part of the same blob. Alternative approaches use the full-body tracking of the OpenNI framework to help in hand detection [21] or provide a basis for full-body gesture detection [22].

Only few approaches deal with the simultaneous tracking of body and hands [41,42]. While in [41] the authors have shown how gesture recognition can be improved by tracking both the body and the hands, the only reference to simultaneous and real-time extraction of hand and body skeletons we are aware of is [42].

3 Main Remaining Challenges

3.1 Shortcomings of Current 3D Cameras

ToF Sensors

Low resolution is common in ToF cameras compared to regular RGB ones. While the resolution is sufficient for tracking two hands at a particular distance, the flexible tracking of hands at various distance ranges would require higher resolution. Alternatively, in such cases, the interface might be reduced only to simple gesture recognition via blob tracking, as the fingers might not be clear enough.

(a) (b)

Fig. 3. (a) Frame from a near-range ToF camera (PMD CamBoard Micro): note the large amount of noise in the background and near the edges of objects. (b) Motion artifacts: the moving hand (right) has thinner fingers than the static hand (left).

Working range is limited by the range ambiguities inherent to the ToF principle and also by the illumination they use. This is not necessarily an issue for near-range gestural interfaces, although they tend to have more noise due to the lower level of active illumination (Fig. 3a).

Motion artifacts may lead to erroneous values at the borders of the measured objects. This issue is more prominent for hand gestures, as panning the hand can lead to loss of data around the fingers, effectively making them thinner and therefore harder to track (Fig. 3b).

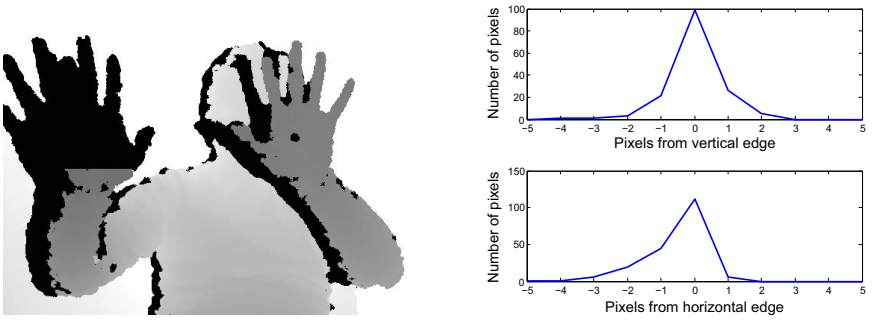
Systematic distance errors, multiple reflections and **flying pixels** may also affect ToF-based gesture interfaces. We refer to [4] and Chapters 1, 2 where various solutions to these problems are discussed.

Structured Light Sensors (PrimeSense Technology)

Low resolution may not be apparent as the device has an output resolution of 640x480. From measurements on the Microsoft Kinect (Fig. 4b), the spatial localization of an edge is approximately 2 pixels off in either direction perpendicular to the edge, which indicates that the effective lateral resolution of the sensor is about 4 times lower in x and y directions.

Working range is limited by design, as the sensor was built to work best at medium range and indoors. At close range (less than 40cm) the sensor fails to produce any data (Fig. 4a). Although there are devices that improve close range output ("near mode" for the Kinect for Windows and the PrimeSense Carmine short range sensor), they are still limited to around 40cm.

Missing data occurs maily due to the baseline between the sensor and the illuminating laser, resulting in shadows around the outside of object borders. Data can also be missing in regions where the diffraction pattern has hot spots (more obvious at close range). Also, where the sensor does not have enough information from the pattern to make a depth measurement, small patches of missing data can occur (Fig. 4a).



(a) (b)

Fig. 4. (a) Kinect sensor artifacts: missing data (black) due to shadowing and near range (right hand) and hot spots (left hand). (b) Vertical (top) and horizontal (bottom) edge fidelity denote reduced lateral resolution. The graphs represent the position deviation of the edges of a rectangular object placed at 100 cm from the Kinect sensor. The deviation is calculated by counting erroneous pixels that correspond to the background inside the object boundary and vice-versa over a strip of the edge. Similar results have been found by [43].

Environment Limitations

One must also consider the limitations of cameras that require active lighting before using them in environments with substantial infrared light. Typical scenarios are outdoors in full sun, in cars near the dashboard, or in rooms that use high powered incandescent lighting. A different set of limitations is created by scenarios which require the cameras to be behind a transparent cover, such as digital signage or window-shopping entertainment: while the sensors themselves are not affected by a transparent surface, reflexions from the active lighting can cause sensor saturation with loss of information in those regions.

3.2 Latency and Real-Time Performance

Although approaches like the ones to be presented in Section 4 can run in real time on rather modest hardware, future requirements will aim at further reducing cost and size. The complexity of hardware and algorithms must be therefore further reduced.

For most applications, a tight coupling between the hands and the application is essential. Current solutions all provide a more or less squashy and wobbly interaction due to both the latencies of the sensor readout and of the middleware. In addition to reduced latencies, more realistic and predictive models are needed.

Moreover, since the hand can move very fast, with speeds of up to $5\,\mathrm{ms}^{-1}$ for translation and $300°\mathrm{s}^{-1}$ for wrist rotation [24], higher framerates may be required.

3.3 User Interaction

Unintentional movements are a big challenge for gesture recognition when interacting with touch-less interfaces. In contrast to touch-enabled interfaces, where physical contact with the controlling surface indicates the user's intent to begin a gestural command, touch-less interfaces must decide when and where the user actually wants to interact with them. Possible solutions to this problem are:

Active area defines a region which limits the interface to a bounding rectangle (2D) or box (3D) in which the user can use gesture control to interact with the interface. Some type of feedback is needed so that the user can know when he is inside the active area.

Modal interfaces become active when a "clutch" action is performed by the user. This can be anything from giving a vocal command or using a very obvious and unique gesture such as waving, opening the hand to show all five fingers or making a gesture with his other hand. After this action is performed, the interface is active, and the user can interact with it by performing other gestures. Deactivating the interface can be done automatically, e.g. after the user finishes the current gesture, after a preset idle time, or by moving the hand outside the active area of the interface (in conjunction with the previous strategy).

Dwell time is usually implemented for emulating virtual buttons: in order to interact with the interface, the user must perform a gesture for a certain amount of time. For example, to press a button, the user must point to it, and then keep his hand inside the button perimeter for a preset time. This is usually done in conjunction with visual or auditory feedback (a timer, change of color or short sound) to announce to the user that if he keeps doing the gesture the corresponding action will be performed.

Preset idle pose is a variation of the modal interface: instead of switching to the active state after the specific gesture has been performed, here the interface is in a neutral state while a certain gesture is performed, becoming active when the gesture is changed. This is implemented usually to force users in a particular pose to better suit the application purpose. For example, if accuracy is needed for a particular interface, the user could be forced to keep his thumb and index fingers in an "L" shape, with the rest of the fingers being curled. The application can then track the index finger as a cursor and use the thumb moving towards the hand as an indication of interface activation, with the benefit of stability while gesturing (the index finger does not move much when adducting the thumb).

Multi-modal interaction presents the user with other forms of input that can be used to gain attention of the gesture recognition system: a vocal command, head and gaze tracking, or even pushing a button (where extreme robustness is required, for instance in medical applications) can be used to activate the interface.

Ergonomic limitations can become an issue with touch-less interfaces that force the user in unnatural poses. Humans prefer having their hands supported by the work surface, while the work is done with wrist and elbow movements. In the case

of interacting with interfaces by the means of pointing gestures, fatigue and soreness quickly set in if the hand is held in an unsupported position for too long.

Possible solutions to this problem include not being limited to hand-only or pointing gestures for interface control, or requiring the use of hand gestures only for a short time. Cubic-foot applications should not have this issue, as users can support their elbows on the desktop or their body. The main takeaway from this limitation is that interfaces should be designed from the ground up, with human biomechanics in mind.

3.4 Novel Gesture Interfaces and Standards

There is a growing agreement that touch-less gesture interfaces should not be designed as computer-mouse replacements. Instead, the whole interface needs to be re-designed in order to enable the potential of such novel interfaces. In order to be accepted, these novel interfaces must be standardized in the sense that similar actions should be triggered by similar gestures across different applications.

3.5 Multi-modal Interfaces

Examples of alternatives to gesture interfaces are speech recognition and gaze tracking. All these modalities have their strengths and weaknesses and one future challenge will be to fuse them. For example, gaze is much faster for pointing but can hardly produce any semantics, which could be done by speech and/or gestures. The recognition of emotions and the integration of wearable sensors could be further extensions.

4 Selected Recent Developments

4.1 3D Cameras

Currently ToF cameras are becoming much smaller and also cheaper. The Creative Interactive Gesture Camera Developer Kit, for example, costs US$150, while solutions based on PrimeSense technology are more expensive (US$200 and US$250 for the PrimeSense Carmine sensor and Microsoft Kinect for Windows sensor respectively). Similarly, ToF modules for automotive and consumer applications are expected to be targeted at prices well below US$100. The pace at which ToF cameras have been shrinking is impressive and it certainly facilitates the development of near-range gesture interfaces.

Light-field cameras are another interesting option since they neither require a baseline nor active illumination. We have tested Raytrix[1] light field cameras with the algorithms presented in the next section and found that they provide more robust hand-skeleton tracking than standard stereo systems.

[1] `www.raytrix.de`

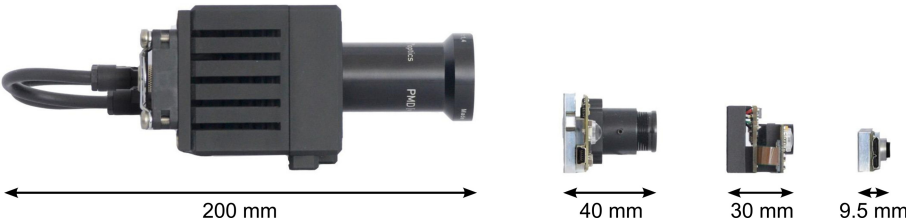


Fig. 5. Various ToF cameras, from left to right: PMD CamCube, PMD CamBoard Micro, PMD CamBoard Nano, and the most recent PMD CamBoard Pico. Other manufacturers have completed a similar miniaturization process (e.g. SoftKinetic).

4.2 Commercial Solutions

There are a number of commercial solutions for desktop PCs which provide a framework for body tracking and gesture recognition, such as the Omek Beckon[2], the SoftKinetic iisu SDK[3] and the Microsoft Kinect for Windows[4].

Probably the most widely known body tracking solution is the Microsoft Kinect for Xbox360. It employs machine learning algorithms to estimate the user's body posture [29], which is then used as input for interactive games on the console. However, extensions to hand gestures have only been recently introduced for the Kinect for Windows, but hand skeleton tracking is still not available.

Regarding hand gesture recognition, a recent collaboration between Intel, Creative and SoftKinetic released the Creative Interactive Gesture Camera Developer Kit[5]. It is a near-range time-of-flight camera that allows tracking of the user's hand up to one meter. While the accompanying software solution does not fully model the hand in 3 dimensions, it does provide the extended fingertips' position and various anatomical landmarks (palm, elbow).

The Leap Motion[6] device promises to allow full 3D tracking of the user's fingers, provided they keep their hands over the device's field of view. The device itself is a small box that needs to be placed on the user's desktop and facing upward. At the time of this writing, the device has not been released yet. Finally, there have been some attempts to use mobile devices to track the user's hand or face and respond to simple gestures.

Another solution for hand and finger detection is provided by Metrilus[7]. Their algorithms include finger tracking, pointing, swipes, and direction evaluation.

For full hand skeleton tracking, 3Gear Systems[8] proposes a desktop solution which involves a PrimeSense[9] Carmine short range sensor mounted above the

[2] www.omekinteractive.com
[3] www.softkinetic.com
[4] www.microsoft.com/en-us/kinectforwindows
[5] www.click.intel.com/intelsdk/
[6] www.leapmotion.com
[7] http://www.metrilus.de/
[8] www.threegear.com
[9] www.primesense.com

user's desk. The system provides hand pose estimation and gesture recognition with the added step of first calibrating the model with the user's hands.

Omek Interactive's Grasp solution promises full hand skeleton tracking, although it is not currently available for review.

An alternative solution for hand skeleton tracking, which is of low complexity and requires no calibration, has been developed by gestigon[10], and it is based on the approach presented in the next section.

4.3 Hand and Body Tracking Using Self Organizing Maps

In this section we will present some of our own recent developments, showing how self-organizing maps (SOM) can be used for hand and full body tracking. We use a range camera for data acquisition and apply a SOM-learning process for each frame in order to capture the pose. While the standard SOM algorithm [6] and some extensions [44] have been proposed before, we will introduce further constraints and a performance analysis on an embedded system. Details on the embedded system implementation are given in [45].

4.3.1 The SOM Tracking Algorithm

SOMs are a well-established method for topology-preserving data transformations and have been used for gesture recognition based on 2D appearance models, for which the SOM can help to find the low-dimensional space of hand-pose transformations [46]. Similarly, in [47] SOMs are used as an intermediate stage to cluster hand trajectories before feeding them into an HMM for gesture recognition. These uses of SOMs, however, are completely different from our approach, which we will describe next.

The node-based SOM tracking algorithm proposed by [6,42] (which we will refer from now on as the Standard SOM Algorithm) takes a different approach, by modeling the hand as a SOM topology. The process starts with the initialization of the network weights in the shape of the hand topology (Fig. 6c) in the center of the hand point cloud, followed by the iteration of two steps: the competition and the update of the weights. At every iteration, a sample point from the dataset is randomly chosen. First, during the competition phase, a winner node (i.e. the weight with the minimum Euclidean distance to the sample point) is computed.

Next, the update phase aims at decreasing the distance between the two points by moving the winner-node weight towards the sample point by a fraction ϵ of the distance between them. The standard SOM algorithm then also applies a neighborhood update, in the sense that not only the winner-node weight is updated, but also the weights of the neighbor-nodes, with a smaller learning rate.

These steps are repeated for hundreds or thousands of iterations. This makes the skeleton graph fit to the point cloud and stay within its confines.

[10] www.gestigon.com

4.3.2 Topology Expansion

We expand the 44-node upper body topology presented in [6,42] (Fig. 6a) to two topologies, one representing the whole body (Fig. 6b), and the other representing the human hand (Fig. 6c). The models were chosen so they mimic the anatomical landmarks of their real-world counterparts — limbs and joints for the body and phalanges and interphalangeal joints for the hand. The rigid bodies (torso and palm) are modeled as a mesh. Both produce good qualitative results in our implementation.

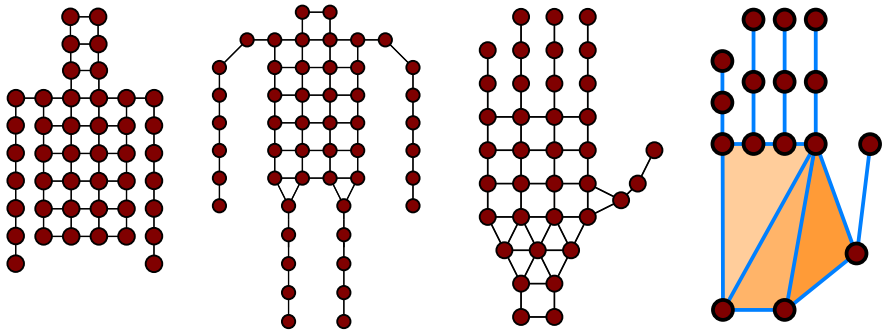


Fig. 6. SOM topologies: (a) The upper body topology proposed in [6]; (b), (c) Expanded topologies for the whole body and the hand; (d) The extended SOM hand topology for segments and planes

4.3.3 The Extended SOM

Our proposed algorithm extends the competition and the update step to 1D and 2D network segments. The 1D-segments are the lines between pairs of connected nodes, and the 2D-segments are the triangles determined by triples of connected nodes. 1D-segments allow to represent the fingers more accurately, and the 2D-segments model the palm of the hand. We now have not only elements of dimension zero (nodes) like in the standard case described in the last section, but also elements of dimension one and two for representing the data distribution. The new topology can be seen in Fig. 6d.

This approach is motivated by the fact that a hand-like topology involves a difficult separation between the nodes corresponding to different fingers. A node that belongs to one finger can easily be attracted by another finger, given the topological closeness. This may lead to an erroneous tracking of the hand and destroy the topological relations. With these 1D and 2D segments we can represent fingers and parts of the palm more accurately and expect the self-organizing maps to be less prone to this type of errors.

4.3.4 Performance Analysis

Because the algorithms have low computational complexity, they can be implemented on low power devices, such as embedded systems. We implemented both

the standard SOM algorithm and the extended version on a PandaBoard ES, which is the next iteration of the popular Pandaboard platform. It is powered by a Texas Instruments OMAP4460 system-on-chip (SoC), which is used in a number of mobile devices available on the market such as the Samsung Galaxy Nexus. The board features a 1.2 GHz dual-core Cortex-A9 ARM CPU, a PowerVR SGX 540 graphics processing unit, 1 GB DDR2 SDRAM, two USB 2.0 ports, Ethernet connection and various other peripherals.

After implementing the standard SOM for the hand and whole-body skeleton, we have obtained the results shown in Figure 7. We have been able to successfully reach our target of real-time performance, at 30 frames per second (FPS).

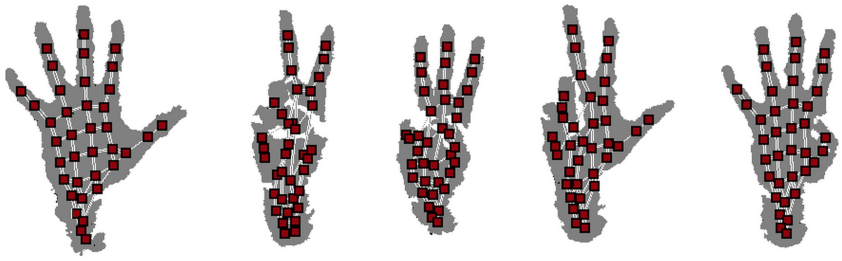


Fig. 7. The Standard SOM Algorithm results for various hand poses [45]

It can be seen that the hand tracker is able to cope with missing data (Fig. 7b,c as white areas on the palm), the skeleton's topology remaining stable, the fingers being retracted in the palm. This is considered to be correct behavior, as the fingers will be reported as "bent" to a subsequent gesture recognition algorithm.

The results for the extended SOM for hand tracking can be seen below. Although the competition and update phases are more complex than those of the standard SOM, the algorithm still runs in real time (30 FPS) with the same number of iterations, as the new topology has less than half the nodes of the old one (16 vs. 37). In figure 8 we show five hand poses taken directly from the real-time video on the embedded platform. The qualitative performance is similar to the one of the node-only SOM implementation. The topology converges correctly on the straightened as well as the bent fingers.

In figures 8d and e, hand poses in which the fingers are held together are shown. The fingers remain in the correct places and do not retract into the palm — the new segment-plane updates solve the problem of the previous SOM implementation that appeared when data points were too close to each other and the nodes from one finger wandered into the space of other fingers. This allows for a more robust representation of the hand gestures, as melded fingers (that could come from out-of-plane hand rotations or hands which are too far away for the camera to distinguish between fingers) are no longer a problem.

The computational efficiency of the method makes it ideal for implementations on a low-powered systems such as embedded devices. The algorithm could be

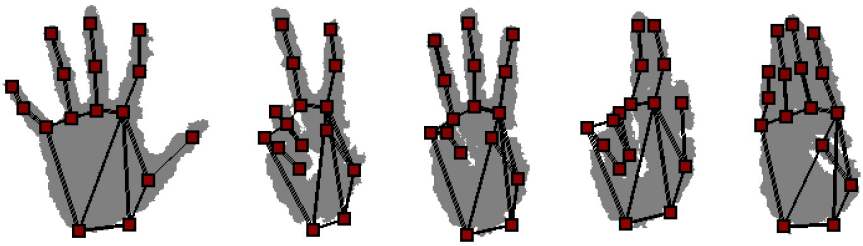


Fig. 8. The extended SOM results for various hand poses

used in such devices, which need low-power, low-complexity solutions to enable gesture technologies — granting extended interaction capabilities to current and future mobile interfaces. Natural user interfaces can be used to enhance the usability of devices ranging from the current mobile devices to the next generation head-mounted displays.

From the testing done with the Microsoft Kinect and PMD CamBoard we concluded that our method is robust and can adapt to any 3D data that is being supplied, as long as it is accurate enough, meaning that the proposed algorithm is able to work with a wide range of cameras. Another benefit is that the self-organizing map approach can easily be applied to any deformable object that needs to be tracked by simply changing the network topology (e.g. torso, full body, pets etc.). This presents a definite advantage over methods that use machine learning to recognize objects, as the self-organizing map algorithm need not be trained in advance.

However, the SOM approach has its limitations: it requires hand segmentation and adequate initialization. Also, since only a topology is being defined, it is not obvious how geometrical constraints can be applied.

5 Example Applications

5.1 Consumer

Cubic Foot applications have been pioneered by Intel in an attempt to make ultrabooks more interesting. The idea is to use the 3D volume spanned by the opened notebook for gesture-based interaction (i.e. the "cubic foot"). Currently, the hand gestures are all based on fingertips, not a full hand skeleton. However, this initiative has contributed to the ongoing miniaturization of ToF cameras and it seems that ToF sensors may win the race for the smallest sensor for gesture interfaces, although some promising alternatives such as the Leap Motion device exist.

Gaming applications have always been big adopters of alternative input interfaces. One of the first commercial touch-less controllers designed for games was the Sony EyeToy, a QVGA resolution webcam that could be used in

low-light environments. Released in 2003, it leveraged the PlayStation 2 powerful video processor to let users interact with games using their whole body as the input device. Other gesture interfaces followed: the Nintendo Wii console along with the Wii Remote as a motion sensing device in 2006, EyeToy's successor, the PlayStation Eye and its accompanying motion sensing controller, the PlayStation Move in 2010, and soon after, the Microsoft Kinect for XBOX360, a definitive boost to gesture control interfaces.

Mobile and Embedded Control is another range of applications that could benefit from gesture control. Almost every commercial gesture control framework on the market today is aimed at desktop PCs, due to the computing power required by the algorithms they use. As mobile platforms shrink in size, gestural interfaces will start to become a viable alternative to touch-based interaction [45]. Gesture control could be a potent interface with which the user could control device parameters of mobile applications or other personal devices such as cameras. Being able to control appliances such as TVs from a distance without the need for a remote control is starting to become a feature in the new range of consumer devices, although only for high-end models, due to the limitations described above.

5.2 Automotive

Automotive suppliers aim at (i) replacing the growing number of buttons and joysticks in the car by a generic virtual interface and (ii) creating new forms of interaction. This will, however, be a gradual development starting with simple gestures that control harmless functions. Due to the extreme variations in ambient light and high demands on reliability and robustness, this application field has its own challenges.

5.3 Medical

The prototypical medical application is that of using gestures during surgery to access medical records or to control equipment in a sterile environment. The obvious benefit is the lack of physical contact between the operator and the device. Moreover, surgeons prefer not having to put down their tools in order to be able to press buttons or touchscreens. We refer to Chapter 4.1 for a review of such applications.

5.4 Digital Signage

As gesture control is a highly user-interactive experience, gesture driven signage will certainly see emergence in the future, with some companies specializing in gesture marketing (e.g. GestureTek [11] and ZiiCON [12]). Possible applications in this area include virtual tours, information kiosks, gaming, art installations, even

[11] `www.gesturetek.com`
[12] `www.ziicon.com`

interactive window-shopping which could, for instance, take the user's clothing dimensions automatically or target marketing based on one's actions. Because the system is tracking the user at all times for gesture input, features such as customizing an ad for the tracked person could be used to grab attention and increase the impact of the signage.

5.5 Sign Language

This is an application that many consider to be obvious and useful [15,16]. From our own experience and extensive discussions with German associations, however, the interpretation of hand gestures is not sufficient for communication because facial expressions are essential for those who "speak" and read sign language. While the extraction of a hand skeleton can provide a good basis for sign recognition, the problem of recognizing facial expressions has to be solved in addition. For an overview of gestural language recognition please refer to Chapter 4.2 of this book.

6 Summary

Gesture interfaces promise to change the way we interact with devices. To fully exploit this potential, however, one needs to rethink the human-machine interface and adapt it to the new technological opportunities. An essential component of such gesture interfaces is hand pose estimation which, as shown in Section 2.2, remains a challenging problem although a number of promising approaches and commercial solutions exist.

One approach to alleviate some of these issues is using a depth sensor. This increases robustness to lighting conditions and gives the possibility of discriminating objects based on their depth, making segmentation a more straightforward process. Depth sensors have led to considerable progress in the field and are now becoming small and low-cost devices, which, however, still need to overcome certain limitations that we have underlined in Section 3.1.

A further limiting factor is the complexity of the algorithms needed to estimate the many degrees of freedom that a gesturing hand can have. This is particularly true for approaches that operate with a full geometrical model of the hand. As an alternative, approaches that only define the topology have lower complexity but may sometimes fail to precisely extract the correct pose and may therefore require additional constraints.

Currently, although commercial solutions exist, they are limited to specific use-cases, such as desktop or cubic foot interaction. We have also presented our own work in the field, which is based on self-organizing maps for hand pose estimation. The method has the benefit of tracking and estimating the hand skeleton in a single stage, with significant performance gains.

Finally, we have presented several commercial application domains in which gesture control can be used in order to build a better interface. As gestures are often used in human communication, is seems natural to extend them to human-machine interaction for more intuitive interfaces.

References

1. Pavlovic, V., Sharma, R., Huang, T.: Visual interpretation of hand gestures for human-computer interaction: A review. IEEE Transactions on Pattern Analysis and Machine Intelligence 19(7), 677–695 (1997)
2. Haker, M., Böhme, M., Martinetz, T., Barth, E.: Deictic gestures with a time-of-flight camera. In: Kopp, S., Wachsmuth, I. (eds.) GW 2009. LNCS, vol. 5934, pp. 110–121. Springer, Heidelberg (2010)
3. Droeschel, D., Stuckler, J., Behnke, S.: Learning to interpret pointing gestures with a time-of-flight camera. In: 2011 6th ACM/IEEE International Conference on Human-Robot Interaction (HRI), pp. 481–488 (2011)
4. Kolb, A., Barth, E., Koch, R., Larsen, R.: Time-of-flight cameras in computer graphics. Computer Graphics Forum 29(1), 141–159 (2010)
5. Böhme, M., Haker, M., Martinetz, T., Barth, E.: A facial feature tracker for human-computer interaction based on 3D Time-of-Flight cameras. International Journal of Intelligent Systems Technologies and Applications 5(3/4), 264–273 (2008)
6. Haker, M., Böhme, M., Martinetz, T., Barth, E.: Self-organizing maps for pose estimation with a time-of-flight camera. In: Kolb, A., Koch, R. (eds.) Dyn3D 2009. LNCS, vol. 5742, pp. 142–153. Springer, Heidelberg (2009)
7. Böhme, M., Haker, M., Martinetz, T., Barth, E.: Head tracking with combined face and nose detection. In: Proceedings of the IEEE International Symposium on Signals, Circuits & Systems (ISSCS), Iaşi, Romania (2009)
8. Böhme, M., Haker, M., Riemer, K., Martinetz, T., Barth, E.: Face detection using a time-of-flight camera. In: Kolb, A., Koch, R. (eds.) Dyn3D 2009. LNCS, vol. 5742, pp. 167–176. Springer, Heidelberg (2009)
9. Böhme, M., Haker, M., Martinetz, T., Barth, E.: Shading constraint improves accuracy of time-of-flight measurements. Computer Vision and Image Understanding 114, 1329–1335 (2010)
10. Holte, M., Moeslund, T., Fihl, P.: View-invariant gesture recognition using 3D optical flow and harmonic motion context. Computer Vision and Image Understanding 114(12), 1353–1361 (2010), Special issue on Time-of-Flight Camera Based Computer Vision
11. Haubner, N., Schwanecke, U., Dorner, R., Lehmann, S., Luderschmidt, J.: Recognition of dynamic hand gestures with time-of-flight cameras. In: Dörner, R., Krömker, D. (eds.) Self Integrating Systems for Better Living Environments: First Workshop, Sensyble. Number 1, pp. 7–13 (2010)
12. Kollorz, E., Penne, J., Hornegger, J., Barke, A.: Gesture recognition with a time-of-flight camera. Int. J. Intell. Syst. Technol. Appl. 5(3/4), 334–343 (2008)
13. Mo, Z., Neumann, U.: Real-time hand pose recognition using low-resolution depth images. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 1499–1505 (2006)
14. Suryanarayan, P., Subramanian, A., Mandalapu, D.: Dynamic hand pose recognition using depth data. In: 2010 20th International Conference on Pattern Recognition (ICPR), pp. 3105–3108 (August 2010)
15. Keskin, C., Kirac, F., Kara, Y., Akarun, L.: Randomized decision forests for static and dynamic hand shape classification. In: 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 31–36 (2012)
16. Keskin, C., Kirac, F., Kara, Y., Akarun, L.: Real time hand pose estimation using depth sensors. In: Fossati, A., Gall, J., Grabner, H., Ren, X., Konolige, K. (eds.) Consumer Depth Cameras for Computer Vision. Advances in Computer Vision and Pattern Recognition, pp. 119–137. Springer, London (2013)

17. Kurakin, A., Zhang, Z., Liu, Z.: A real time system for dynamic hand gesture recognition with a depth sensor. In: 2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO), pp. 1975–1979 (2012)
18. Oikonomidis, I., Kyriazis, N., Argyros, A.A.: Efficient model-based 3D tracking of hand articulations using kinect. In: British Machine Vision Conference, Dundee, UK, vol. 2 (2011)
19. Doliotis, P., Athitsos, V., Kosmopoulos, D., Perantonis, S.: Hand shape and 3D pose estimation using depth data from a single cluttered frame. In: Bebis, G., Boyle, R., Parvin, B., Koracin, D., Fowlkes, C., Wang, S., Choi, M.-H., Mantler, S., Schulze, J., Acevedo, D., Mueller, K., Papka, M. (eds.) ISVC 2012, Part I. LNCS, vol. 7431, pp. 148–158. Springer, Heidelberg (2012)
20. Ren, Z., Yuan, J., Zhang, Z.: Robust hand gesture recognition based on finger-earth mover's distance with a commodity depth camera. In: Proceedings of the 19th ACM International Conference on Multimedia (MM 2011), pp. 1093–1096. ACM, New York (2011)
21. Caputo, M., Denker, K., Dums, B., Umlauf, G.: 3D hand gesture recognition based on sensor fusion of commodity hardware. In: Reiterer, H., Deussen, O. (eds.) Mensch & Computer 2012: Interaktiv Informiert Allgegenwärtig und Allumfassend!?, München (Oldenbourg Verlag), pp. 293–302 (2012)
22. Reyes, M., Dominguez, G., Escalera, S.: Feature weighting in dynamic time warping for gesture recognition in depth data. In: 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), pp. 1182–1188 (2011)
23. Zetzsche, C., Barth, E., Wegmann, B.: The importance of intrinsically two-dimensional image features in biological vision and picture coding. In: Watson, A.B. (ed.) Digital Images and Human Vision, pp. 109–138. MIT Press (October 1993)
24. Erol, A., Bebis, G., Nicolescu, M., Boyle, R.D., Twombly, X.: Vision-based hand pose estimation: A review. Computer Vision and Image Understanding 108(12), 52–73 (2007), Special Issue on Vision for Human-Computer Interaction
25. Wren, C., Azarbayejani, A., Darrell, T., Pentland, A.: Pfinder: Real-time tracking of the human body. IEEE Transactions on Pattern Analysis and Machine Intelligence 19(7), 780–785 (1997)
26. Campbell, L., Becker, D., Azarbayejani, A., Bobick, A., Pentland, A.: Invariant features for 3-D gesture recognition. In: Proceedings of the Second International Conference on Automatic Face and Gesture Recognition, pp. 157–162 (1996)
27. Ballan, L., Taneja, A., Gall, J., Van Gool, L., Pollefeys, M.: Motion capture of hands in action using discriminative salient points. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part VI. LNCS, vol. 7577, pp. 640–653. Springer, Heidelberg (2012)
28. Oikonomidis, I., Kyriazis, N., Argyros, A.A.: Markerless and efficient 26-DOF hand pose recovery. In: Kimmel, R., Klette, R., Sugimoto, A. (eds.) ACCV 2010, Part III. LNCS, vol. 6494, pp. 744–757. Springer, Heidelberg (2011)
29. Shotton, J., Sharp, T., Kipman, A., Fitzgibbon, A., Finocchio, M., Blake, A., Cook, M., Moore, R.: Real-time human pose recognition in parts from single depth images. Commun. ACM 56(1), 116–124 (2013)
30. Lahamy, H., Litchi, D.: Real-time hand gesture recognition using range cameras. In: Canadian Geomatics Conference (CGC), vol. 10 (2010)
31. Malassiotis, S., Tsalakanidou, F., Mavridis, N., Giagourta, V., Grammalidis, N., Strintzis, M.: A face and gesture recognition system based on an active stereo sensor. In: Proceedings of the 2001 International Conference on Image Processing, vol. 3, pp. 955–958 (2001)

32. Breuer, P., Eckes, C., Müller, S.: Hand gesture recognition with a novel IR time-of-flight range camera–A pilot study. In: Gagalowicz, A., Philips, W. (eds.) MIRAGE 2007. LNCS, vol. 4418, pp. 247–260. Springer, Heidelberg (2007)
33. Zhu, X., Wong, K.Y.K.: Single-frame hand gesture recognition using color and depth kernel descriptors. In: 2012 21st International Conference on Pattern Recognition (ICPR), pp. 2989–2992 (November 2012)
34. Liu, X., Fujimura, K.: Hand gesture recognition using depth data. In: Proceedings of the Sixth IEEE International Conference on Automatic Face and Gesture Recognition, pp. 529–534 (2004)
35. Van den Bergh, M., Van Gool, L.: Combining RGB and ToF cameras for real-time 3D hand gesture interaction. In: IEEE Workshop on Applications of Computer Vision (WACV), pp. 66–72 (2011)
36. Trindade, P., Lobo, J., Barreto, J.: Hand gesture recognition using color and depth images enhanced with hand angular pose data. In: 2012 IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), pp. 71–76 (2012)
37. Ghobadi, S., Loepprich, O., Hartmann, K., Loffeld, O.: Hand segmentation using 2D/3D images. In: Cree, M.J. (ed.) IVCNZ 2007 Conference, University of Waikato, pp. 64–69 (2007)
38. Holte, M., Moeslund, T., Fihl, P.: Fusion of range and intensity information for view invariant gesture recognition. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW 2008), pp. 1–7 (2008)
39. Uebersax, D., Gall, J., Van den Bergh, M., Van Gool, L.: Real-time sign language letter and word recognition from depth data. In: 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), pp. 383–390 (2011)
40. Hernandez-Vela, A., Bautista, M., Perez-Sala, X., Ponce, V., Baro, X., Pujol, O., Angulo, C., Escalera, S.: BoVDW: Bag-of-visual-and-depth-words for gesture recognition. In: 2012 21st International Conference on Pattern Recognition (ICPR), pp. 449–452 (2012)
41. Song, Y., Demirdjian, D., Davis, R.: Multi-signal gesture recognition using temporal smoothing hidden conditional random fields. In: 2011 IEEE International Conference on Automatic Face Gesture Recognition and Workshops (FG 2011), pp. 388–393 (2011)
42. Haker, M., Barth, E., Martinetz, T.: Method for the real-time-capable, computer-assisted analysis of an image sequence containing a variable pose, International patent WO/2010/130245 (filed: May 6, 2010)
43. Andersen, M.R., Jensen, T., Lisouski, P., Mortensen, A.K., Hansen, M.K., Gregersen, T., Ahrendt, P.: Kinect depth sensor evaluation for computer vision applications. Technical report ECE-TR-6, Department of Engineering Electrical and Computer Engineering, Aarhus University (2012)
44. State, A., Coleca, F., Barth, E., Martinetz, T.: Hand tracking with an extended self-organizing map. In: Estevez, P.A., Principe, J.C., Zegers, P. (eds.) Advances in Self-Organizing Maps. AISC, vol. 198, pp. 115–124. Springer, Heidelberg (2013)
45. Coleca, F., Klement, S., Martinetz, T., Barth, E.: Real-time skeleton tracking for embedded systems. In: Proceedings of Multimedia Content and Mobile Devices SPIE Conference, vol. 8667 (2013)
46. Guan, H., Feris, R., Turk, M.: The isometric self-organizing map for 3D hand pose estimation. In: 7th International Conference on Automatic Face and Gesture Recognition (FGR 2006), pp. 263–268 (2006)
47. Caridakis, G., Karpouzis, K., Drosopoulos, A., Kollias, S.: Somm: Self organizing markov map for gesture recognition. Pattern Recognition Letters 31(1), 52–59 (2010)