Sensyble Workshop 2010 manuscript No.
(will be inserted by the editor)

Recognition of Dynamic Hand Gestures with Time-of-Flight Cameras

Nadia Haubner[1], Ulrich Schwanecke[2], Ralf Dörner[1], Simon Lehmann[1], Johannes Luderschmidt[1]

[1] RheinMain University of Applied Sciences, Wiesbaden Rüsselsheim Geisenheim, Department of Design, Computer Science and Media, Kurt-Schumacher-Ring 18, 65197 Wiesbaden, Germany
[2] RheinMain University of Applied Sciences, Wiesbaden Rüsselsheim Geisenheim, Department of Design, Computer Science and Media, Unter den Eichen 5, 65195 Wiesbaden, Germany

Received: date / Revised version: date

Abstract In this paper an algorithm to recognize single stroke gestures using a Time-of-Flight camera is presented. To achieve this goal, freehand gesture input is provided by articulating a single stroke in three-dimensional space using a stretched finger. The position of the fingertip is tracked and gesture input can be deactivated by forming a fist. In this way, the start and end of a gesture is recognized automatically. The gesture is then recognized with the $1 gesture recognizer which is adapted to depth data. The algorithm is applied to recognizing ten gestures composed of the numbers zero to nine and evaluation results reveal a fingertip tracking accuracy of over 88% and an average gesture recognition rate of 75%.

1 Introduction

Since gestures provide a natural way of communication between humans, gesture recognition is a major field of research in human-computer interaction. High attention was paid on computer vision methods which offer an unobtrusive way of gesture input as there is no need for the user to attach any sensors to his body. Vision based systems have been introduced that use a single intensity or color image to recognize gestures. As gestures occur in three-dimensional (3D) space, recognition based on two-dimensional (2D) information only can lead to ambiguities. To overcome this drawback, other approaches first extract 3D information out of one or more images of the observed scene to achieve a more robust gesture recognition. Here, sometimes complex image processing algorithms that either determine distinct features such as edges or silhouettes have to be applied or correspondences in multiple images have to be found. On the one hand these algorithms can be error prone and on the other hand may imply a high computational cost. An overview on those algorithms can be found in [31].

This paper presents a computer vision based approach for recognizing dynamic hand gestures in interactive framerates using a Time-of-Flight (TOF) camera. This kind of sensor captures an image that provides a depth value for each pixel. Images are provided with approximately 25 frames per second. The provided depth information obtained with the TOF camera is used directly in each frame for segmenting the foreground of the observed scene and for the extraction of the fingertip, which is used to perform the gestures. It is assumed that the finger is pointing in the direction of the camera and thus the fingertip has the shortest distance to the camera. In combination with smoothing on the raw depth image, straight forward and robust tracking of the fingertip is achieved.

The proposed gesture recognition algorithm is a 3D adaptation of the $1 gesture recognizer that was introduced as an easy and slim algorithm to recognize 2D single stroke gestures on touch displays, e.g. Pocket PCs [41]. In our approach, gestures are performed in 3D space by drawing a path in space with the fingertip. The tracking of the fingertip can be activated and deactivated by the user by switching between a stretched finger and a fist. The positions of the tracked fingertip results in a 3D polygon which describes the performed gesture. The $1 gesture recognition works by matching a performed gesture with a set of templates. To permit a direct comparison of a gesture polygon and a template polygon both are preprocessed in the same manner. First, polygons are resampled to achieve a fixed number of equidistant points. Then they are transformed to obtain invariant rotation, scale and translation. In this way, the matching can be accomplished by determining the Euclidean distance between gesture and template polygons.

The remainder of this paper is organized as follows. First related work is presented and based on this then our approach for the recognition of dynamic hand gestures using depth data is described. Afterwards, the algorithm is evaluated with respect to its accuracy using an exemplary set of gestures composed of the numbers

zero to nine. Here, the fingertip tracking and the gesture recognition are evaluated separately. Finally, an outlook on future improvements is given.

2 Related Work

As already stated in the introduction, gesture recognition in general has been a popular field of research. Especially computer vision based approaches have been of high interest as these offer an unobtrusive way of gesture input. Overviews of vision based gesture recognition approaches are given in [30,33,31,13].

The recognition of hand gestures in particular has been of high interest as those gestures can provide a natural and direct way of human-computer interaction. Currently, the most effective technology used for sensing hand based input is the data glove [40]. One drawback of this device is that it has to be worn on the user's hand to capture motion which limits the naturalness of the interaction. Additionally, user dependent calibration is also a problem. To overcome these limits, vision based approaches to capture hand motion have been investigated. Surveys on this topic can be found in [10,22].

Several vision based systems to capture body and hand motion only use single images [1,38]. Here, occlusions and ambiguities are likely to occur. Multi-view tracking approaches use multiple cameras to overcome these drawbacks [5,9,36]. Thoug, in both single and multiple image based approaches, features have to be extracted from the images before conclusions about the three dimensional scene can be made. Stereo vision methods have been introduced to determine depth information of the scene [21]. Some approaches use additional information like color and texture [35,27]. However, tracking algorithms based on the mentioned systems can be error prone and computational expensive.

Among others, TOF cameras [24] have been introduced to obtain depth information directly from the scene. This kind of sensor captures a range image in real time and has been applied to the field of human-computer interaction by a few approaches. These range from full body tracking over head tracking and face detection to the recognition of hand gestures and some of them are discussed in the following paragraphs.

In [18] Haker et al. present an algorithm using self-organizing maps to fit an upper body model to the 3D data for pose estimation. The model captures the human pose robustly and can be used to track the major body parts. Guðmundsson et al. describe a system for people tracking in smart rooms where a TOF camera is used to enhance people segmentation in a multiple camera setup in [15]. A comparison of a color image based and a range image based people tracking algorithm is conducted in [37]. Approaches about tracking multiple people with a TOF camera are presented in [4] and [20].

Efforts to use TOF cameras for the purpose of head tracking and face recognition have been presented in [29]. This is achieved by a nose tracking and the determination of a spherical intersection profile of the face. Scale-invariant features in range maps [17] and the Viola-Jones face detector [7] are also used for face detection with TOF cameras.

Hand gesture based interaction using TOF cameras has also been in the scope of many investigations. In [28] Liu et al. analyse the hand shape and motion to recognize dynamic hand gestures and point out the usefulness of depth data for the segmentation of hands. Breuer et al. use a principal component analysis to estimate the hand pose and fit it into a hand model as a pilot study in the field of TOF data based hand gesture recognition in [8]. In [3] Benko et al. investigate how freehand gestures can extend multi-touch based interaction on a vertical display by placing a depth camera behind a transparent display screen. Ahn et al. realize a spatial touch display by defining two depth zones in which hand gestures are recognized [2]. Haker et al. present a robust deictic gesture recognizer based on a TOF camera in [19]. Kollorz et al. recognize static hand gestures using x- and y-projections and a nearest neighbour classification in [25]. Static hand gestures are recognized with a TOF camera and applied to the navigation in 3D medical images in [39].

In our approach, dynamic hand gesture recognition mainly is based on the $1 gesture recognizer that first was presented in [41] and adapted to recognizing gestures in 3D acceleration data in [26]. It was introduced to recognize 2D single stroke gestures on touch displays, e.g. Pocket PCs and turned out to be an easy and slim algorithm which is especially useful in real-time applications due to its low computational time. In addition to that it achieved a recognition accuracy higher than 97% which encourages further investigation of that approach.

3 Technology

In this approach a TOF camera is used to obtain depth data, namely the PMD[vision]®CamCube 2.0[1] which is shown in figure 1. This sensor works by illuminating the environment with modulated infrared light and measuring the time the reflected light needs to arrive at the sensor chip using the phase shift of the light wave. From this time a distance can be determined for each pixel. As a result an depth image with 204x204 pixels is obtained. In addition, intensity values for every pixel are also retrieved. The raw data is quite noisy and high attention was paid to analysing and reducing this noise in previous work. This includes systematic error analysis [34,11], noise reduction by solely investigating the depth data [6] and sensor fusion approaches [14,16,32,23].

In this work, very simple measures to reduce the noise in the raw data are taken. First, outlier removal is achieved by checking the direct neighbourhood of the depth image for similar depth values. If there are no such

[1] www.pmdtec.com

Figure 1: PMD[vision]®CamCube 2.0

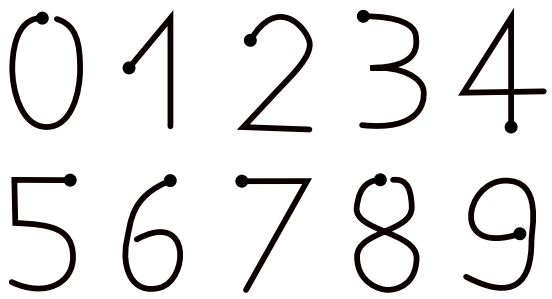


Figure 2: Single stroke gestures

neighbours, a value is declared as outlier and is removed. Afterwards the data is smoothed by applying a Gaussian filter.

4 Recognition of Dynamic Hand Gestures

The proposed gesture recognition algorithm is composed of two main parts. First, a fingertip tracking provides gesture input which can be activated and deactivated by the user switching between a stretched finger and a fist. Second, gestures are recognized based on the $1 gesture recognizer that has been adapted to depth data. The algorithm is applied to single stroke gestures composed of the numbers zero to nine which are shown in figure 2. Next, the fingertip tracking is presented. Afterwards, the original $1 gesture recognizer is depicted briefly before the adapted algorithm is explained in more detail.

4.1 Fingertip Tracking

The fingertip is defined as the closest point to the camera and thus determined as the measurement with the shortest distance. In this way, a straight forward tracking can be implemented by exploiting the obtained depth data directly. However, to provide the user the possibility to control gesture input there has to be found a way to activate and deactivate the fingertip tracking. This is achieved by distinguishing between a stretched finger and a non-stretched finger. A convenient solution for

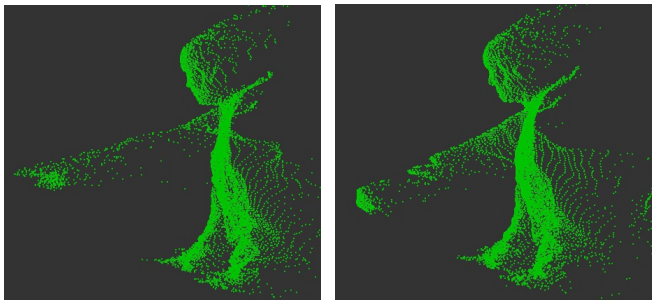


Figure 3: The depth data can be used intuitively to distinguish between a stretched finger and a fist.

user input is to switch between a stretched finger and a fist as can be seen in figure 3. In this example, the tracking is inactive as long as the fist is hold frontally to the camera. The activation of the fingertip tracking can be accomplished by stretching a finger in direction of the camera and the fingertip will be tracked immediately. Again, the depth information captured with each frame is used intuitively for the distinction.

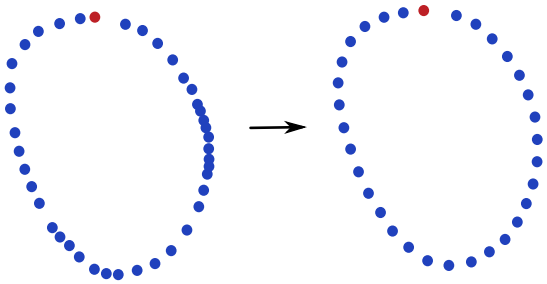
4.2 $1 Recognizer for Depth Data

The $1 gesture recognizer [41], on which our approach mainly is based, passes through four steps which are depicted in figure 4.

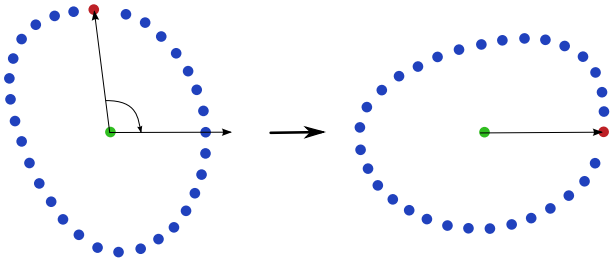
First, the captured point path is resampled to achieve a fixed number of equidistant points. In this way gesture paths can be compared directly. In the next step the gesture is rotated once by its indicative angle. This angle is defined as the angle formed by the vector of the gestures centroid and the gesture's first point and the vector of the gestures centroid and a coordinate on the zero axis. Afterwards, the point path is scaled so that it fits into a unit square to avoid differences in the aspect ratio and translated to origin. Now the gesture path is transformation invariant and can be matched with a gesture template that has also been processed to achieve transformation invariance. The rotation of the point path by its indicative angle might not be the optimal rotation to match it with the template, as there exists no closed-form solution to determine the rotation in such a way that the path-distance to the template is minimized. Because of that an iterative optimization of the indicative rotation is conducted in the template matching step.

For the adapted $1 gesture recognition algorithm a gesture is defined by the tracked fingertips, i.e. a 3D polygon. As the coordinates of this polygon tend to be very noisy, it is smoothed using a corner cutting algorithm [12]. The resampling of the polygon is carried out the same way as in the original $1 recognition algorithm except for extending this to 3D.

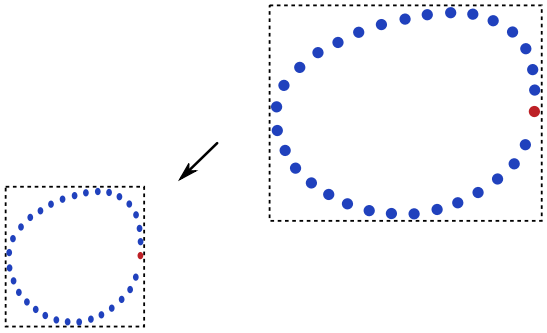
With real 3D gestures it would make sense to also extend the remaining steps of the recognition process in



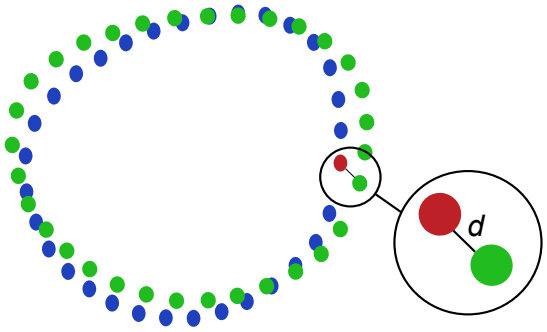
(a) Resample the raw point path.



(b) Rotate by the indicative angle.

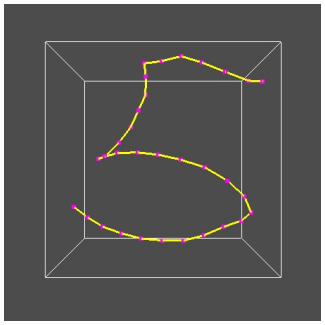


(c) Transform and scale.

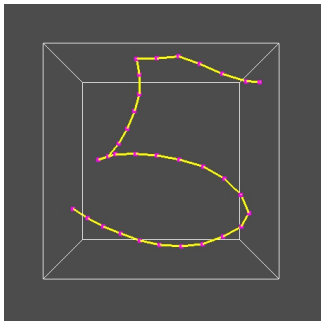


(d) Match with gesture template.

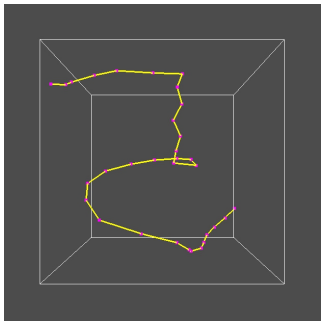
Figure 4: Steps of the $1 gesture recognizer.



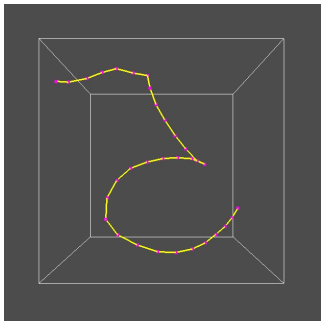
(a) Front view of the original $1 invariant transformed gesture polygon.



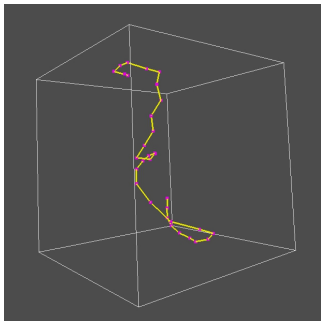
(b) Front view of the first projected and then transformed gesture polygon.



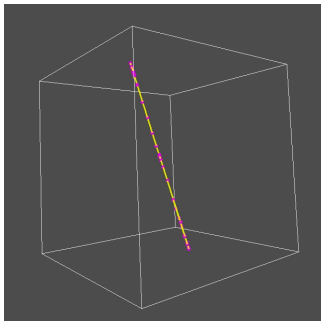
(c) Side view of the original $1 invariant transformed gesture polygon.



(d) Side view of the first projected and then transformed gesture polygon.



(e) Flat view of the original $1 invariant transformed gesture polygon.



(f) Flat view of the first projected and then transformed gesture polygon.

Figure 5: The gesture polygon is projected to the plane perpendicular to the polygon's third component to avoid distortion.

a straight forward manner. But the number set cannot be considered as a real 3D gesture set as the gestures are intended to be articulated in a plane. As a result from a straight forward extension of scaling the gesture to a cube to achieve invariance in aspect ratio, the gesture will get distorted. This phenomenon is shown in figure 5 (a), (c) and (e). To avoid this, the gesture first is projected onto the plane it most likely was intended to be performed. This is achieved by determining the principal components of the gesture polygon and then projecting it onto the plane that is perpendicular to the third component. This operation is also illustrated in figure 5 (b), (d) and (f). Now the gesture can be treated as two dimensional and the transformation steps of the original $1 algorithm can be applied. Finally, the gesture is matched with the resampled and transformation invariant gesture templates in the same manner as with the $1 recognizer.

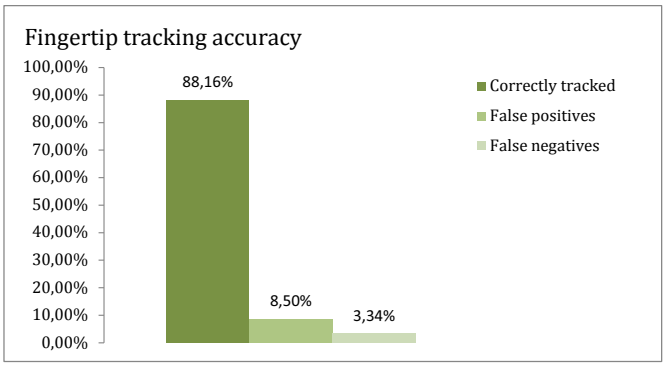


Figure 6: Fingertip tracking accuracy.

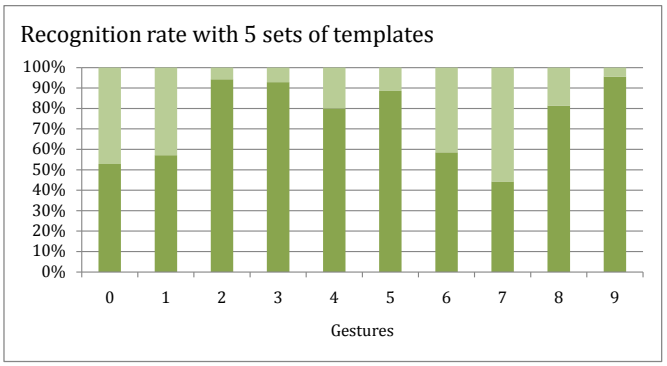


Figure 7: Recognition rate of single stroke gestures using five sets of gesture templates.

5 Evaluation

The evaluation is twofold. First, the activation of the fingertip tracking is evaluated with respect to the recognition reliability of the stretched finger. Afterwards, the gesture recognition with the $1 recognizer is evaluated separately.

5.1 Test Setup

A set of ten single stroke gestures composed of the numbers zero to nine was defined for evaluating the recognition. The gesture set is depicted in figure 2. Test data was gathered by asking 14 participants to perform the gesture set five times to obtain 70 data sets in total. For this purpose they were sitting in front of the ToF-Camera and facing it. Then they were asked to perform the ten gestures in a row by stretching the index finger while drawing a number in space and separating gestures by forming a fist with their hand.

5.2 Fingertip Tracking

The ground truth for the evaluation of the fingertip tracking is established by manually marking the frames of the gathered test data when the finger is stretched. The proposed fingertip tracking algorithm is then applied to the test data and a frame by frame comparison is conducted. In total approximately 40,000 frames were obtained and of these 88.16% were tracked correctly, the rates of false positives and false negatives were 8.50% and 3.34% respectively. These results are also shown in figure 6. False results in the tracking are likely to be caused by noisy raw data which is strengthened by very fast movements of the finger.

5.3 Gesture Recognition

To evaluate the accuracy of the gesture recognition algorithm, the ground truth was also established manually. Gesture polygons were defined by the fingertip coordinates of consecutively stretched fingers based on the manually marked frames already used in the former evaluation. Each gesture then was identified manually and compared with the algorithm based gesture recognition. The results are shown in figure 7. On the one hand, it can be observed, that the recognition rates of the gestures zero, one, six and seven are quite low. It is assumed, that this is due to the high similarity in shape of the gestures zero and seven and zero and six respectively. Especially in 3D space it is not easy to perform a gesture path exactly the way it is supposed to be performed. On the other hand, the recognition rate of the other six remaining gestures are at least 80%, with three of them even more than 90%.

6 Conclusion and Future Work

In this paper an algorithm for recognizing 3D single stroke gestures with TOF cameras was presented. Gesture input is realized by a fingertip tracking which can be activated and deactivated by the user by switching between a stretched finger and a fist. The recognition is an adaptation of the $1 gesture recognizer which has been modified to be used with depth data. An important extension to the original algorithm is the projection of the raw 3D gesture polygon onto the plane which fits best into the polygon. In this way, gestures do not have to be articulated in the same camera perspective as the gesture templates which enhances the robustness of the algorithm. Results show that the algorithm is feasible for this aim but gestures need to have a distinctively different shape than the rest of the gesture set.

Improving the accuracy of the raw distance data would probably provide a more accurate fingertip tracking. In addition, recognizing the stretched finger will become more robust. As a result of this, the recognition of false gestures will likely decrease. So far, analysing the principal components of a gesture polygon is only used to determine a projection plane. An additional benefit of the principal components is that they contain information about how a polygon is orientated in 3D space. This can be exploited in using the same gesture polygons and

distinguishing them only by orientation, e.g. gestures indicating rotation. Moreover, the gesture recognition can be made more flexible by executing the template matching with a set of templates composed of several possibilities of how a gesture can be performed by the user. Finally, tracking of multiple fingertips and the recognition of multi stroke gestures are considered as a worthwhile extensions to the current algorithm.

References

1. A. Agarwal and B. Triggs. Recovering 3d human pose from monocular images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(1):44–58, 2006.
2. Y.-K. Ahn, Y.-C. Park, K.-S. Choi, W.-C. Park, H.-M. Seo, and K.-M. Jung. 3d spatial touch system based on time-of-flight camera. *WSEAS Trans. Info. Sci. and App.*, 6:1433–1442, 2009.
3. H. Benko and A. D. Wilson. DepthTouch: Using Depth-Sensing Camera to Enable Freehand Interactions On and Above the Interactive Surface. Technical Report MSR-TR-2009-23, Microsoft Research, 2009.
4. A. Bevilacqua, L. D. Stefano, and P. Azzari. People tracking using a time-of-flight depth sensor. *Advanced Video and Signal Based Surveillance, IEEE Conference on*, 0:89, 2006.
5. J. Black and T. Ellis. Multi camera image tracking. *Image and Vision Computing*, 24(11):1256–1267, 2006.
6. M. Böhme, M. Haker, T. Martinetz, and E. Barth. Shading constraint improves accuracy of time-of-flight measurements. *Computer Vision and Image Understanding*, 114(12):1329–1335, 2010.
7. M. Böhme, M. Haker, K. Riemer, T. Martinetz, and E. Barth. Face detection using a time-of-flight camera. In *Dynamic 3D Imaging*, volume 5742, pages 167–176. Springer Berlin / Heidelberg, 2009.
8. P. Breuer, C. Eckes, and S. Müller. Hand gesture recognition with a novel ir time-of-flight range camera – a pilot study. In *Computer Vision/Computer Graphics Collaboration Techniques*, volume 4418 of *Lecture Notes in Computer Science*, pages 247–260. Springer Berlin / Heidelberg, 2007.
9. K. Cheung, S. Baker, and T. Kanade. Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 1, pages 77–84, 2003.
10. A. Erol, G. Bebis, M. Nicolescu, R. D. Boyle, and X. Twombly. Vision-based hand pose estimation: A review. *Computer Vision and Image Understanding*, 108 (1-2):52–73, 2007.
11. D. Falie and V. Buzuloiu. Noise characteristics of 3d time-of-flight cameras. In *Signals, Circuits and Systems, 2007. ISSCS 2007. International Symposium on*, volume 1, pages 1–4, 2007.
12. G. E. Farin. *Curves and Surfaces for Computer-Aided Geometric Design: A Practical Code*. Academic Press, Inc., Orlando, FL, USA, 4th edition, 1996. ISBN 0122490541.
13. D. M. Gavrila. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding*, 73(1):82–98, 1999.
14. S. A. Guðmundsson, H. Aanæs, and R. Larsen. Fusion of stereo vision and time-of-flight imaging for improved 3d estimation. *Int. J. Intell. Syst. Technol. Appl.*, 5:425–433, 2008.
15. S. A. Guðmundsson, R. Larsen, H. Aanæs, R. P. Plads, K. Lyngby, M. Pardàs, and J. R. Casas. Tof imaging in smart room environments towards improved people tracking. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW '08. IEEE Computer Society Conference on*, pages 1–6, 2008.
16. U. Hahne and M. Alexa. Combining time-of-flight depth and stereo images without accurate extrinsic calibration. *Int. J. Intell. Syst. Technol. Appl.*, 5:325–333, 2008.
17. M. Haker, M. Böhme, T. Martinetz, and E. Barth. Scale-invariant range features for time-of-flight camera applications. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW '08. IEEE Computer Society Conference on*, pages 1–6, 2008.
18. M. Haker, M. Böhme, T. Martinetz, and E. Barth. Self-organizing maps for pose estimation with a time-of-flight camera. In *Dynamic 3D Imaging*, volume 5742, pages 142–153. Springer Berlin / Heidelberg, 2009.
19. M. Haker, M. Böhme, T. Martinetz, and E. Barth. Deictic gestures with a time-of-flight camera. In *Gesture in Embodied Communication and Human-Computer Interaction*, volume 5934, pages 110–121. Springer Berlin / Heidelberg, 2010.
20. D. W. Hansen, M. S. Hansen, M. Kirschmeyer, R. Larsen, D. Silvestre, and D. Silvestre. Cluster tracking with time-of-flight cameras. *Computer Vision and Pattern Recognition Workshop*, 0:1–6, 2008.
21. I. Haritaoglu, D. Beymer, and M. Flickner. Ghost3d: Detecting body posture and parts using stereo. *Motion and Video Computing, IEEE Workshop on*, 0:175, 2002.
22. R. Hassanpour, S. Wong, and A. Shahbahrami. Vision-Based Hand Gesture Recognition for Human Computer Interaction: A Review. In *IADIS International Conference Interfaces and Human Computer Interaction 2008*, pages 125–134, 2008.
23. Y. M. Kim, D. Chan, C. Theobalt, and S. Thrun. Design and calibration of a multi-view tof sensor fusion system. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW '08. IEEE Computer Society Conference on*, pages 1–7, 2008.
24. A. Kolb, E. Barth, R. Koch, and R. Larsen. Time-of-Flight Sensors in Computer Graphics. *Eurographics State of the Art Reports*, pages 119–134, 2009.
25. E. Kollorz, J. Penne, J. Hornegger, and A. Barke. Gesture recognition with a time-of-flight camera. *Int. J. Intell. Syst. Technol. Appl.*, 5:334–343, 2008.
26. S. Kratz and M. Rohs. A $3 gesture recognizer: simple gesture recognition for devices equipped with 3d acceleration sensors. In *Proceeding of the 14th international conference on Intelligent user interfaces*, pages 341–344, New York, NY, USA, 2010. ACM.
27. M. W. Lee and I. Cohen. Proposal maps driven mcmc for estimating human body pose in static images. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, 2004.

28. X. Liu and K. Fujimura. Hand gesture recognition using depth data. In *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on*, pages 529–534, 2004.
29. S. Meers and K. Ward. Face recognition using a time-of-flight camera. In *Computer Graphics, Imaging and Visualization, 2009. CGIV '09. Sixth International Conference on*, pages 377–382, 2009.
30. S. Mitra and T. Acharya. Gesture recognition: A survey. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 37(3):311–324, 2007.
31. T. B. Moeslund, A. Hilton, and V. Krüger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104(2-3):90–126, 2006.
32. J. Mure-Dubois and H. Hügli. Fusion of time of flight camera point clouds. In *Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications*. HAL - CCSD, 2008.
33. R. Poppe. Vision-based human motion analysis: An overview. *Computer Vision and Image Understanding*, 108(1-2):4–18, 2007.
34. H. Rapp, M. Frank, F. A. Hamprecht, and B. Jähne. A theoretical and experimental investigation of the systematic errors and statistical uncertainties of time-of-flight-cameras. *Int. J. Intell. Syst. Technol. Appl.*, 5:402–413, 2008.
35. T. J. Roberts, S. J. McKenna, and I. W. Ricketts. Human tracking using 3d surface colour distributions. *Image and Vision Computing*, 24(12):1332–1342, 2006.
36. R. Rosales, M. Siddiqui, J. Alon, and S. Sclaroff. Estimating 3d body pose using uncalibrated cameras. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, 2001.
37. L. Sabeti, E. Parvizi, and Q. M. J. Wu. Visual Tracking Using Color Cameras and Time-of-Flight Range Imaging Sensors. *Journal of Multimedia*, 3(2):28–36, 2008.
38. H. Sidenbladh, M. Black, and D. Fleet. Stochastic tracking of 3d human figures using 2d image motion. In *Computer Vision - ECCV 2000*, volume 1843, pages 702–718. Springer Berlin / Heidelberg, 2000.
39. S. Soutschek, J. Penne, J. Hornegger, and J. Kornhuber. 3-d gesture-based scene navigation in medical imaging applications using time-of-flight cameras. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW '08. IEEE Computer Society Conference on*, pages 1–6, 2008.
40. D. Sturman and D. Zeltzer. A survey of glove-based input. *Computer Graphics and Applications, IEEE*, 14 (1):30–39, Jan. 1994.
41. J. O. Wobbrock, A. D. Wilson, and Y. Li. Gestures without libraries, toolkits or training: a $1 recognizer for user interface prototypes. In *Proceedings of the 20th annual ACM symposium on User interface software and technology*, pages 159–168, New York, NY, USA, 2007. ACM.