# 3-D Gesture-Based Scene Navigation in Medical Imaging Applications Using Time-Of-Flight Cameras

[1]Stefan Soutschek, [1]Jochen Penne, [1]Joachim Hornegger, [2]Johannes Kornhuber
[1]Chair of Pattern Recognition, Department of Computer Science
[2]Department of Psychiatry and Psychotherapy
University of Erlangen-Nuremberg, Erlangen, Germany
{Stefan.Soutschek, Jochen.Penne, Joachim.Hornegger}@informatik.uni-erlangen.de
Johannes.Kornhuber@uk-erlangen.de

## Abstract

*For a lot of applications, and particularly for medical intra-operative applications, the exploration of and navigation through 3-D image data provided by sensors like ToF (Time-of-Flight) cameras, MUSTOF (Multisensor-Time-of-Flight) endoscopes or CT (Computed Tomography) [8], requires a user-interface which avoids physical interaction with an input device. Thus, we process a touchless user-interface based on gestures classified by the data provided by a ToF camera. Reasonable and necessary user interactions are described. For those interactions a suitable set of gestures is introduced. A user-interface is then proposed, which interprets the current gesture and performs the assigned functionality. For evaluating the quality of the developed user-interface we considered the aspects of classification rate, real-time applicability, usability, intuitiveness and training time. The results of our evaluation show that our system, which provides a classification rate of* $94.3\%$ *at a framerate of* $11$ *frames per second, satisfactorily addresses all these quality requirements.*

## 1. Introduction

In our daily life speech and gestures are fundamental in communicating with our environment. People that are not able to talk, either through disease or while doing sports like diving, have their own set of gestures which makes it possible to communicate with everyone who understands the meanings of this sign language.

For Human-Machine interaction, most user-interfaces are centered around tactile oriented devices like mouse, keyboard or touch-screens. However, speech- and gesture recognition systems are becoming more competitive in some special application areas, *e.g*. the automotive field [1] and the medical sector [11].

While speech recognition systems already found there way in the operating room, gesture based Human-Machine-Interfaces (HMI) are still in their infancy. Especially here, these interfaces would be very useful since the surgeon normally is not allowed to touch devices as he has to remain sterility. A big advantage of our gesture based HMI over a speech recognition system is, that it does not need any additional equipment like a microphone that impedes the surgeon while operating.

## 2. State of the Art

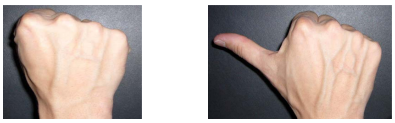
Up to now most gesture based approaches needed additional devices like data gloves, colored gloves or other objects which help segment and identify the gesture in the acquired camera image. These devices are often impractical or even inapplicable and therefore lowered the willingness to work with such an interface. It is not feasible for a surgeon to wear a data glove during the surgery. Other gesture recognition systems based on 2-D video data [3, 5, 12, 4] that do not need additional objects are limited in their functionalities. Thus HMIs based on these systems [9, 11] either provide only a limited functionality or need a complex set of gestures to compensate for the missing depth information.

But with the help of the latest developments in the field of 3-D cameras which directly provide 3-D depth information in addition to 2-D gray value images, several new opportunities have opened up for the investigation of gesture based interfaces. Gesture recognition system using Time-of-Flight (ToF) cameras to classify hand gestures [7] or to interpret the movement of the hand [2, 10] are already proposed.

The ToF camera system utilized for this work is the MESA SR-3100, which is based on a pixelwise measuring of the time of flight of an actively emitted optical reference signal. The used ToF camera provides a resolution of $176 \times 144$ pixels at a framerate of $\geq 15$ fps, i.e. ap-

978-1-4244-2340-8/08/$25.00 ©2008 IEEE

(a) Translation (b) Cursor (c) Click



(d) Rotation (e) Reset

Figure 1. Set of gestures

(a) ToF camera SwissRanger 3100 (b) 3-D data texturized with amplitude data (c) 3-D data of Segmented hand

Figure 2. ToF camera and segmentation of the hand from the acquired 3-D point and amplitude data.

prox. 20000 3-D points are available in real-time. 55 light-emitting diodes (LEDs) with a central wavelength of 850 nm compose the illumination unit. The output of the camera is a complete distance map in addition to the local brightness in the scene. The local brightness values are referred to as amplitude values and are commonly used as a texture for the 3-D data. Note that each 3-D point projects uniquely to one pixel and the center of the coordinate system coincides with the optical center of the ToF camera.

In our approach a user-interface which interprets gestures and enables a user to explore and navigate through 3-D data sets based on a real-time dynamic gesture recognition system using the SR-3100 (see Figure 2(a)) has been developed. This system, compared to those mentioned above, uses the 3-D data provided by the ToF camera not only for the gesture recognition and classification, but also for enlarging the functionality of the HMI by utilizing the additional depth information.

## 3. Gesture Recognition

In gesture based user-interfaces, one must first decide on an appropriate set of gestures for communicating necessary information. Once the gesture vocabulary is decided, one can then concentrate on the image analysis tasks. In our case these are composed of segmenting the hand, computing appropriate features and recognizing the shown gesture based on classification of the computed features.

### 3.1. User interactions and set of gestures

The exploration and navigation of 3-D data sets requires at least the possibility to rotate and translate the reconstructed scene. Furthermore, a movable cursor has to be available in the visualization of the data set to point to objects of interests (show some conspicuity to colleagues, etc.). By utilizing a gesture that represents a ”mouse”-click, points can be consecutively selected, which is important for measuring the size of anatomical structures. A further common clinical requirement is the possibility to specify a volume of interest (VOI) for further analysis. Finally, a ”reset” gesture provides the functionality to undo previously performed actions.

Based on these functional requirements we selected the 5 gestures shown in Figure 1. The main focus when specifying the gestures was to get a small and intuitive gesture set. Furthermore, the complexity of physically performing a certain gesture was chosen to be as small as possible.

### 3.2. Segmentation, feature extraction and classification of gestures

A coarse segmentation of the hand is accomplished by applying lower and upper thresholds to the depth data acquired by the ToF camera. These thresholds define the minimum and maximum working distance.

For a finer estimation, the mean distance of all segmented pixels that have an amplitude value higher than a threshold, which marks the border between object and background, is calculated. The resulting distance can be interpreted as the distance of a plane, parallel to the ToF chip that lies between the body and the hand. As long as the hand is not directly next to or behind the body, which would not be very comfortable for the user and therefore can be neglected, this assumption is fulfilled. This yields a 3-D connected point cloud, which represents the hand attached to a part of the forearm as displayed in Figures 2(b) and 2(c).

The next crucial step is the removal of the forearm by using the 2-D pixel coordinates of each segmented 3-D point. The method implemented in our work extends the idea of [3], which crops the forearm in two steps. First, the palm is described as that circle in the image with the largest radius which only contains foreground, i.e. segmented, pixels. Deriving the position of the forearm is done by iteratively increasing the radius of the circle, followed by an analysis of the intersections of the new circle with the segmented pixels following the rule based search algorithm, introduced in [3]. If the radius is enlarged, the circle will intersect for example fingers if they are spread apart, but it will in any case also intersect the forearm. If the segment that repre-

(a) Segmented Hand (b) Features (c) Segmented Hand (d) Features

Figure 3. Sampling the cropping-circle for the hand in steps of 1°

sents the forearm has been determined, the cropping line can be set up as the tangent of the palm circle that intersects the straight line defined by the midpoint of the forearm segment and the midpoint of the determined cropping-circle. If no forearm segment has been found, the current frame is discarded.

To derive suitable features from the segmented hand which are used for the classification process, the cropping-circle is sampled in adequate angle steps and the information, whether a circle point lies on the hand or not, is stored for each angle.

The images in Figure 3 show a segmented hand ((3(a)), (3(c))) and the extracted features ((3(b)), (3(d))). It is obvious that the two smaller peaks of the features represent the two fingers and the larger one the beginning of the forearm. When the hand is rotated the features remain scale-invariant and only a translation in the feature space is the result.

For classification of the gestures the "k-d Tree Based k-Means Clustering" and the "Bayesian Plug-In Classifier", which are included in the medical image processing library "Insight Segmentation and Registration Toolkit"[1] (ITK), as well as an approach based on nearest neighbor have been tested against each other (Results see section 5.1). To further reduce errors caused by a wrong classification, a majority vote from several classified gestures is performed.

## 4. Navigation

Once a set of gestures is defined and an image-based methology for identifying them is established, the HMI can be set up. A sketch of the main algorithm, which shows how the different interactions are integrated into the interface is given in Figure 4. Please note that the maximum range for performing all gestures is reduced to a size that is smaller than the field of view of the camera. Otherwise the hand that performs the gesture would be only partially visible and lead to wrong classifications.

### 4.1. Cursor Movement

The basic functionality that is available for a user is the movement of a cursor in the 3-D data. Having the 3-D position of the hand $H$ (determined by the center of mass of 3-D points of the segmented hand) at the distance $d = \|H\|$ and also the required intrinsic camera parameters, it is possible to calculate the projection of the center of mass of the hand to the image plane $h$, by the following equation:

$$h = \frac{f * H}{d * p} \tag{1}$$

The required intrinsic camera are the focal length $f$ and the physical dimensions $p$ ($p = p_x = p_y$ as the camera system used has square pixel) of a pixel on the camera chip.

[1] http://www.itk.org/

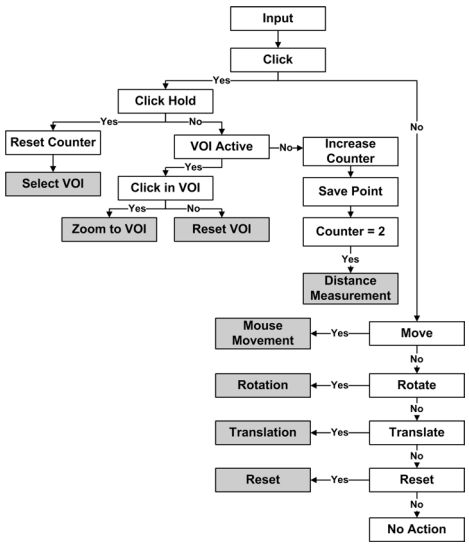



Figure 4. Overview of the navigation system

### 4.2. "Click" and Selection of a VOI

If the "click" gesture is observed (see Figure 1(c), 2(b), 2(c)), the current position of the projected point (computed from the center of mass of the hand as described above) is saved and continuously displayed until the user triggers a third "click" event while two points have already been selected. If two points are selected, the interface automatically initiates a calculation of the Euclidean distance between these points which provides the functionality of measuring distances in 3-D. With the third "click" all selected points are erased in the 3-D data and the user can start another selection.

Besides the selection of points, a VOI can also be specified with the "click"-gesture by holding the first "click" gesture for at least one second. After this second, the first corner of the volume is fixed and it can be expanded by holding the "click" gesture while moving the hand. When the final volume is selected, the definition is completed by performing any other gesture. Based on this selection, the user has two choices to continue his work with the interface. A click

anywhere outside the VOI will result in a reset which clears the reconstruction from the selected VOI and the user can start the next selection or any other task. Moving the cursor inside the VOI and triggering a click event there, will result in a zoom to the selected region of interest which is realized by recalculating the position and orientation of the virtual camera.

To calculate the new position of the virtual camera and the new viewing direction, several steps have to be performed. Our first step involves determining the position and parameters of the virtual camera which could generate the desired "zoomed-in" view. Let $d$ be the distance of the virtual camera to the center of the 3-D data set. $v_x$ and $v_y$ are the width and height of the VOI, $\alpha$ and $\beta$ are the half of the aperture angle for the width respectively for the height angle angle of the virtual camera. In order to assure that the complete VOI is visible after the zoom, we are always using the larger values according to width and height and the corresponding aperture angle. For example, if the height $v_y$ is larger than the width the equation for the calculation of the new distance $d$ reads as

$$d = \frac{v_y}{2 * \tan \beta}. \tag{2}$$

Furthermore, one needs to know in which direction the virtual camera has to be translated. To compute this direction, two vectors $a$ and $b$ are defined. Both vectors start at the midpoint $m$ of the VOI. $a$ ends in the upper left and $b$ in the upper right corner of the VOI. Calculating the crossproduct

$$u = b \times a \tag{3}$$

of these vectors will result in a vector $u$, which points in the direction the virtual camera needs to be translated. With the midpoint $m$ of the volume of interest as starting point, the new position $c$ of the virtual camera can be determined by

$$c = m + d * \frac{u}{|u|}. \tag{4}$$

### 4.3. Rotation and Translation

Unlike the methods described above, the projection of the center of mass onto the image plane is important to get information in which direction the 3-D data set has to be translated respectively rotated.

When either the rotation or the translation gesture is recognized (see Figure 1(d), 1(a)), a virtual cube, (see Figure 5) with a side length that corresponds to the dimension of the field of view of the ToF camera at the current distance of the gesture, is set up and subdivided into 27 smaller equal-sized cubes. After the cube has been set up, it remains fixed until any other gesture is performed. Each sub cube, except the center one, represents a direction or a rotation axis, that is either one or a combination of the axes of the camera

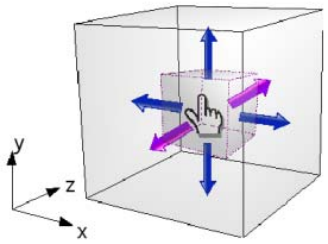



Figure 5. Virtual cube (light gray) for rotation and translation. (For a better overview, not all 27 sub-cubes but only the center cube (dark gray) is displayed.

coordinate system. According to the classified gesture the virtual camera is translated or rotated using the current direction or rotation axis. The subdivision into smaller cubes is done so as to avoid, moving the 3-D data set at the moment the gesture of either the rotation or the translation is recognized. Rather, the user needs to explicitly activate the action by moving his hand to the center cube. After activating the rotation or translation, the user is able to modify the view to the scene by navigating the according gesture to one of the other cubes.

The available modifications for the 3-D data set are the rotation around the three axes triggering the rotation and a translation along the three axes triggering the translation. This is quite an advantage for the user compared to using a standard mouse device, which can only handle two dimensional transformations. The gesture-based navigation system intuitively enables the user to involve the third dimension, by just moving the hand forward or backward. A rotation around the z-axis for the rotation and a zoom for the translation are realized with that additional dimension. To end the rotation respectively translation, the user just needs to perform any other gesture or return to the center cube.

Both actions, the rotation and the translation, will have no effect on previously selected points, to assure, that the user is able to continue his prior work, but from a better viewing position as before.

### 4.4. Reset

The final functionality implemented for the gesture-based HMI introduced in this work, offers the possibility to jump back to the starting point of the exploration of the 3-D reconstruction. This is important as one often desires to restart the navigation from a default position.

By performing the reset gesture (see Figure 1(e)) for at least one second, all changes concerning the position and the orientation of the virtual camera made so far are set to default values. This reset does not affect any selected points or volumes in the 3-D reconstruction, so that no work per-

formed up to the reset is lost.

## 5. Experimental Evaluation and Discussion

The evaluation is divided into two parts. The first part deals with the software evaluation including a short analysis of the classification rates of the implemented classifiers and the evaluation concerning the real-time applicability of the whole system.

Besides the technical evaluation of the implemented software, the usability, the intuitiveness and the training time of the HMI also play an important role for this work. For that reason, tests were performed to validate the user-friendliness of the proposed HMI. This experiments form the second part of the evaluation.

### 5.1. Evaluation of the classifier

For each classifier a ten fold cross-validation using a broad set of reference gestures has been performed. More specifically, in total 40 data sets per gesture stored from each of the 15 test person, form the basis for this evaluation. Out of these data sets, 45 data sets for each gesture are generated, consisting of 3 arbitrary chosen data sets out of the 40 data sets per gesture from each test person. The test users were not advised to take off rings or pull up their sleeves. The resulting classification rates are presented in Table 1 and show that our approach provides a robust, user-independent classification.

| | **Amount of features per gesture** | | | |
|---|---|---|---|---|
| | **histogram based** | | **sampled** | |
| **Classifier** | **160** | **320** | **90** | **120** |
| **Nearest Neighbor** | 92.4 % | 89.3 % | 80.9 % | 81.3 % |
| **k-d Tree** | 76.9 % | 78.7 % | 73.3 % | 76.0 % |
| **Bayesian** | 45.8 % | 72.4 % | 76.0 % | 75.1 % |

Table 1. Classification rates using an user-independent set of reference gestures

Applying a principal component analysis (PCA, [6] ) on the feature vectors improved the results (see Table 2). As the nearest neighbor approach performed best, only this results are presented here.

| | **Amount of features per gesture** | | | |
|---|---|---|---|---|
| | **histogram based** | | **sampled** | |
| **Classifier** | **160** | **320** | **90** | **120** |
| **Nearest Neighbor** | 94.3 % | 87.4 % | 94.3 % | 90.6 % |

Table 2. Classification rates applying principal component analysis for an user-independent set of reference gestures

### 5.2. Evaluation of real-time applicability

The best classification results and the most intuitive set of gestures are not of interest for a user if the interface has a perceptible delay. For that reason, this subsection provides the results of a performance evaluation of the implemented algorithms which is performed on a single core Pentium M 1.87GHZ.

The performance of the total system, first without and then with activated HMI, is evaluated. This is done so as to see the calculation time of the classification and navigation algorithms in relation to the overall calculation time. The results of this test are shown in Table 3. This values are mean values calculated out of fifty independent measurements.

| **Step** | **Without HMI** | | **Including HMI** | |
|---|---|---|---|---|
| | [ms] | [%] | [ms] | [%] |
| Data acquisition | 36.15 | 49.86 | 37.54 | 40.12 |
| Preprocessing | 36.35 | 50.13 | 37.19 | 39.74 |
| Classification | 0.00 | 0.00 | 16.38 | 17.50 |
| Navigation | 0.01 | 0.01 | 2.47 | 2.64 |
| **Framerate** | **13.80 fps** | | **10.69 fps** | |

Table 3. Performance measurements of the system

Even when the case the gesture-based HMI is activated, most of the calculation time is spent on the acquisition (40.12 %) and preprocessing (39.74 %) of the camera data. Activating the HMI reduces the overall performance by just 3.11 frames per second.

Clearly, most of the computational effort is spent in algorithms, which are independent of the gesture-based Human-Machine-Interface. With a maximum loss of 3.11 frames per seconds which is equivalent to a computational time of 25.77 ms, the HMI itself meets the requirements concerning real-time applicability.

### 5.3. Evaluation of usability, intuitiveness and training time

To gain information about the usability, intuitiveness and training time of the implemented HMI, seven persons, mainly computer science students so far, were asked to test the user-interface after they got a two minute introduction to the system. Each of the seven users was asked to use our HMI and perform the different kinds of gestures for about three to five minutes so as to familiarize oneself with the HMI.

Next, the user was requested to answer a list of four questions:

1. How would you evaluate the response time of the system?

2. How strong did you need to adapt to the system?

3. How good was the comfort of performing the set of gestures?

4. How intuitive are the gestures for the mouse movement and the click?

Each question had to be answered on a scale from one up to five, where one corresponds to "worse" respectively "strong adaptation" and five corresponds to "very good" respectively "no adaptation".

The answers given by the test users to the first question (see Table 4), allows to combine the pure technical evaluation to the user impressions while really working with the system. With a mean of 4.14, the result is satisfactory as there are still possibilities to improve the overall performance of the complete system. This result also reveals, that a frame rate of about 11 frames per second still validates the real-time applicability of the HMI.

| | Question | Mean |
|---|---|---|
| **1** | Response time of the system | 4.14 |
| **2** | Adaptation to the system | 3.57 |
| **3** | Comfort of gesture set | 4.00 |
| **4** | Intuitiveness of the gesture set | 4.00 |

Table 4. Evaluation results for questionnaire

Responses to question 2 (see Table 4) show that the opinions concerning the adaptation to the system differ. Except for one user, the individual results of this evaluation can be judged as good, taking into account that the test persons used the system for the first time. An important aspect for the evaluation of the usability was the comfort of the gestures (Table 4) while using the interface.

With an average of 4.0 (see Table 4), a satisfying result concerning question 4, that was posed to the test users after they had the chance to test the system for about three to five minutes, has been achieved. This question was asked to evaluate the intuitiveness of the gesture-based interface taking the cursor movement and click event as an example. During the testing phase it could be observed that after a few minutes, each user was able to perform the gestures in a way which enabled a proper handling of the system.

## 6. Conclusion and future work

In this work we present a complete framework for a gesture-based user-interface for the exploration and navigation through 3-D data sets using a ToF camera. The proposed interface was evaluated considering the aspects of real-time applicability, usability, intuitiveness and training time. By achieving a 94.3% classification rate for 5 gestures at ≈10 fps we consider the algorithmic requirements for the gesture-based HMI using ToF cameras fulfilled. Furthermore, the user feedback concerning usability of the gesture-based HMI validates the feasibility of the proposed system. The future work will focus on an enhancement of the functionality for example integrating virtual buttons that trigger special operations on selected volumes of interest and furthermore on an larger evaluation in a clinical environment.

## References


[1] F. Althoff, R. Lindl, and L. Walchshäusl. Robust Multimodal Hand- and Head Gesture Recognition for controlling Automotive Infotainment Systems, 2005. In VDI-Tagung: Der Fahrer im 21. Jahrhundert, Braunschweig, Germany.

[2] P. Breuer. Entwicklung einer prototypischen Gestenerkennung in Echtzeit unter Verwendung einer IR-Tiefenkamera, 2005. Diplomarbeit, Universitaet Koblenz-Landau, Koblenz, Germany.

[3] B. Deimel. Entwicklung eines stabilen videobasierten Verfahrens zur Segmentierung der Hand am Unterarm, 1998. Diplomarbeit, Department for Graphical Systems, University of Dortmund, Germany.

[4] A. Elgammal, V. Shet, Y. Yacoob, and L. Davis. Learning dynamics for exemplar-based gesture recognition. *CVPR '03: Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages I–571–I–578 vol.1, 2003.

[5] S. Funk and S. Fuchs. Entwicklung eines gestisch-intuitiven Mensch-Maschine-Interfaces aud Basis der videogestuetzden Erkennung von Handzeichen, 2002. Technischer Bericht, Technische Universitaet Dresden, Dresden, Germany.

[6] I. Jolliffe. Principal Component Analysis, 1986. Springer Verlag, New York, NY.

[7] E. Kollorz and J. Hornegger. Gesture recognition with a time-of-flight camera, 2007. In DAGM e.V. 2007 - Proceedings of the workshop Dynamic 3D Imaging in conjunction with DAGM'07.

[8] J. Penne, K. Höller, S. Krüger, and H. Feußner. Notes3D: Endoscopes learn to see 3-D, 2007. In VISAPP 2007 - 2nd International Conference on Computer Vision Theory and Applications.

[9] Y. Sato, M. Saito, and H. Koik. Real-Time Input of 3D Pose and Gestures of a User's Hand and Its Applications for HCI, 2001. In VR 2001: Proceedings of the Virtual Reality 2001 Conference, p.79, IEEE Computer Society, Washington, DC, USA.

[10] T. Sünkel. Erkennung isolierter komplexer Handgesten in 2 1/2D Videosequenzen mit Hidden Markov Modellen, 2006. Diplomarbeit, Friedrich-Alexander Universitaet Erlangen-Nuernberg, Erlangen, Germany.

[11] J. Wachs, H. Stern, Y. Edan, M. Gillam, C. Feied, M. Smith, and J. Handler. A real-time hand gesture interface for a medical image guided system, 2006. In ISRACAS 2006 - Ninth Israeli Symposium on Computer-Aided Surgery, Medical Robotics, and Medical Imaging.

[12] S. B. Wang, A. Quattoni, L.-P. Morency, and D. Demirdjian. Hidden Conditional Random Fields for Gesture Recognition. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1521–1527, Washington, DC, USA, 2006.