Applying Hand Gesture Recognition with Time-of-Flight Camera for 3D Medical Data Analysis

Filip Malawski
AGH University of Science and Technology, Faculty of Computer Science, Electronics and Telecommunication
Department of Computer Science
al. Mickiewicza 30, 30-059 Krakow, Poland
e-mail: fmal@agh.edu.pl

This paper describes a human-computer interface based on hand gesture recognition, intended for analysis of 3D medical data. The gestures are designed to minimize the required muscle tension when using the system. Gesture recognition is based on a 3D sensor. Depth maps are acquired by a time-of-flight camera, designed specifically for hand gestures recognition. The depth images are denoised and segmented to right and left hand. The contours of the hands are found and a modified Shape Context descriptor is utilized for each hand, providing a set of features, which are employed to train and test various classifiers. Naive Bayes, Random Forest and Support Vector Machine (SVM) classifiers are utilized, with search of optimal parameters using cross-validation. The best accuracy (95%) is achieved with the Support Vector Machine classifier. The gestures are mapped to various controls of a 3D medical visualization module. Two visualization methods are employed - isosurface and cut-planes. The left hand is assigned to switching between different control modes and the right hand gestures are corresponding to controlling various properties in each mode. The system is convenient to use and runs in real-time on a typical PC machine.

Key words: human-computer interaction, hand gesture recognition, time-of-flight camera

Introduction

Hand gesture recognition is an area of extensive research. Touchless human-computer interfaces have been drawing attention of both scientific and business communities for decades. Hand gestures can be employed as an alternative to the standard mouse interface. In particular, 3D applications, such as games, design tools or data analysis applications, may benefit the most, by utilizing the fact that our hands are able to express a wide range of gestures and can operate in 3D space, contrary to the mouse. Moreover, using hands is the most natural way of interaction with different things.

Multiple solutions for hand gesture recognition have been presented over the years. They can be divided into two main groups: methods based on 2D images and methods based on 3D images. Methods from the first group often focus on efficient extraction of regions of interest. Searching for the hand in the image is mostly based on skin color and texture features, which both are illumination-dependent. Several advanced methods have been proposed in this area. In [1] a hierarchical elastic graph matching is employed, in [2] authors utilize Lucas-Kanade algorithm. Application of neural networks for shape fitting is investigated in [3].

Recently however, depth sensors, such as Microsoft Kinect and time-of-flight (ToF) cameras, have gained considerable popularity in the area of hand gestures recognition. Depth cameras are robust to illumination changes and greatly facilitate extraction of relevant data in the image, since foreground/background segmentation can be performed easily with the available depth data. For the 3D data acquisition the most commonly used device is Microsoft Kinect. Approaches found in literature employ a range of different descriptors – histograms of oriented gradients (HOG) [4], Finger-Earth Mover's Distance (FEMD) [5], contour model [6]. In [7] the authors combine multiple depth-based descriptors and utilize Support Vector Machine (SVM) classifier. In [8] the authors focus on the fingertip localization, using the Oriented Radial Distribution and providing a custom database with color-labeled fingertips.

Due to being inexpensive, Kinect is a common choice for hand gesture recognition. However, it is not the best-suited solution for this application, since the sensor was designed to recognize whole body rather than just hands. Time-of-flight cameras can often provide more precise data, therefore several methods based on these devices have been proposed as well. In [9] a chamfer distance matching is employed for shape recognition and Finite State Machine (FSM) is utilized for hand trajectory recognition. In [10] the authors propose a method employing the Principal Component Analysis (PCA) with subsequent model based fine-matching.

Despite potential advantages and numerous recognition methods, human-computer interfaces based on gestures have not actually gained much popularity. Performing the gestures for a prolonged time has proved to be too tiring and inconvenient. In this paper a hand gesture based interface is presented, which emphasizes the ease of use, rather than wide range of gestures. The interface recognizes

4 gestures, separately for each hand and is applied for analysis of 3D medical data. Recognition is based on a modified Shape Context descriptor and various classifiers. Depth maps are acquired by CamBoard Nano time-of-flight sensor, produced by PMD.

Methods

The methods employed in this work are discussed in 4 subsections describing a) design of gestures b) data acquisition and preprocessing c) gestures recognition d) 3D medical data visualization.

Gestures

As mentioned before, the main focus of the proposed solution is to allow for convenient use. Base position of the hands is presented in Fig. 1 - the user keeps both hands in front of him, with exterior surfaces facing the camera, and fingers directed towards the opposite hand. This position requires much less muscle tension than facing the camera with the interior surfaces of the hands and pointing upwards, as is often the case in other gesture-based interfaces. Moreover, hands can be rested on the table when needed, with minimal movement.

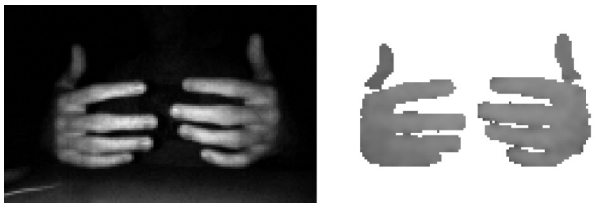

Figure 1. Basic position of hands – 2D image (left) and depth map (right)

Each hand is processed separately. The user may 'close' or 'open' fingers and hide or show the thumb. Therefore, there are 4 possible gestures (per hand), as depicted in Fig. 2 (gestures for the other hand are analogous). These 4 gestures, together with the position of the hand, will be sufficient for control of the 3D medical data visualization. The user is required to keep the hands in such a position, that they do not overlap. When the hands are seen by the depth sensor as one continuous object this in fact constitutes additional gesture, which indicates that the interaction should be paused. Therefore when the user has a need to rest the hands, he needs only to join them and lay on the table.

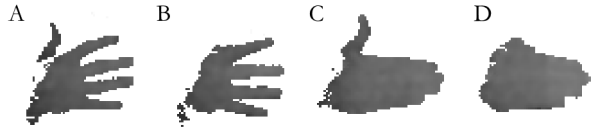

Figure 2. Gestures for one hand (depth map)

Data acquisition and preprocessing

For the acquisition of the depth data CamBoard Nano time-of-flight sensor has been employed. It provides a 160x120 depth map with frame rate up to 90 fps and field of view 90^0 x 68^0. The sensor is specifically tailored to recognize hands – it works in close range and handles best surfaces such as human skin. Along with the depth map, it provides a set of flags for each pixel. Valid pixels have no flags set, while invalid ones have flags describing for what reason is the pixel invalid – e.g. too much noise.

The setup of the acquisition assumes that the user keeps his hands in distance about 20-30 cm from the sensor, where the image of the hands is large enough to provide precise recognition and at the same time the user has enough operating space for performing the gestures. For the purpose of gesture classification a database of the 4 gestures has been acquired. Each gesture has been recorder 50 times, resulting in total of 200 images. The gestures have been acquired for one hand only, since gestures of the other hand can be recognized by applying symmetry.

Before being passed to the gesture recognition module, the images are preprocessed in order to discard all irrelevant pixels. Preprocessing of a single frame is a multi-stage operation. In the first stage, the closest pixel is found and all pixels, which are more than 15 cm further, are marked as invalid. This provides a simple, yet effective background/foreground segmentation.

Next step is the noise reduction. Although the sensor detects most noise and marks the corresponding pixels as invalid, additional noise reduction is required. Single pixels or groups of pixels may occur, which are close to the hands regions, but are not actually connected with them. These can have negative impact on the further processing, therefore must be removed. For each pixel, the number of invalid pixels in its 3x3 neighborhood is computed and if there are at least 6 invalid pixels in its neighborhood, the pixel is marked as invalid as well. This method also provides smoothing of the hand regions by removing 'protruding' pixels on the edges.

Subsequently, segmentation to left and right hand is performed. The algorithm searches for a vertical line of invalid pixels, starting in the middle of the image and moving towards the edges, further in each iteration, in both directions simultaneously. Moreover number of valid pixels on both sides of the line must be greater than zero, in order to handle situation, where hands are joined and moved to one side. If no such line is found between 1/4 and 3/4 of image width, the image is classified as the resting position. Otherwise, the image is divided to left and right hand images. Each hand is now processed separately.

Finally, position of each hand is computed as a mean position of all valid pixels in the segment relevant for this hand. However, due to the noise and the fact that the hand is usually not completely steady, this can introduce undesired 'shaking' of the controlled object or property. In order

to compensate for this problem, mean hand position from last 5 frames is utilized.

Gesture recognition

Once the noise reduction and segmentation steps are completed the descriptors may be computed. Valid pixels, which have at least one invalid adjoining pixel, are marked as contours and then the Shape Context (SC) [11] descriptor is employed.

The basic idea of the Shape Context descriptor is to select a number of evenly distributed points on the contour and compute their relative positions. For each pixel, length and direction of the vectors connecting the pixel with all other pixels are computed. Fig. 3 (left) depicts points selected on a hand contour. For one of the points of the thumb a few of the connecting vectors are presented.

The set of all connecting vectors for each point is a too specific descriptor, therefore, in order to make it more discriminative a histogram is created for each point. The bins of the histogram are in a log-polar space, which results in closer points having more impact on the final descriptor. Figure 3 (right) shows the diagram of the log-polar bins used to compute the descriptor for one of the points of the thumb.

In the original version of the Shape Context descriptor a histogram is computed for each point and then the histograms are used for matching the points of two shapes. However, in order to take advantage of different classifiers we need to create a single descriptor. One approach would be to concatenate the histograms, although this would create a descriptor which is both very long and dependent on the number of selected points. Instead, we propose to combine all histograms into one, by taking the average. This constitutes a single descriptor, where each bin corresponds to one feature. The dimensions of the histogram are 12 x 5, which gives a reasonable number of features.

Furthermore, the histogram is normalized, making it scale-independent. We can choose all the points from the contour, regardless of its size, as long as the relative number of points for each finger is similar.

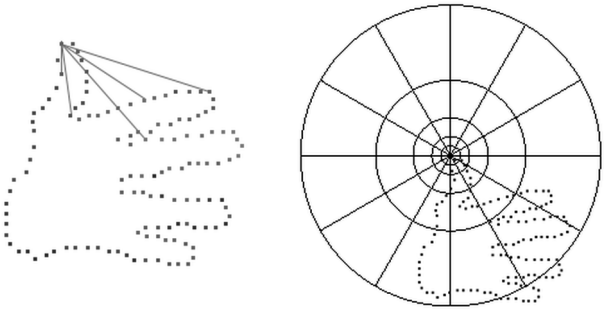

Figure 3. Shape Context descriptor: points selected from contour, with a few connecting vectors (left); diagram of log-polar bins (right)

For classification of the gestures several classifiers have been tested – Naive Bayes (NB), Random Forest (RF) [12] and Support Vector Machine (SVM) [13] with Radial Basis Function (RBF) kernel. Classifiers implementation was provided by the WEKA package.

Visualization

For the visualization module the Visualization Toolkit (VTK) framework has been employed. It provides various visualization methods commonly used for 3D medical data. In the proposed solution two methods have been utilized – isosurface and cut-planes. The first one creates a 3D surface of points with the same value. Opacity and the isovalue can be controlled by the user. In the second method 3 cut-planes are displayed, each in different axis. Their position can be controlled, as well as level and window parameters of the data in all cut-planes. Both isosurface and cut-planes are presented in fig. 4.

The four gestures (A-D, see fig. 2) have been mapped to control the properties of the visualization. For both hands the A gesture is corresponding to the idle state, meaning that no control is active for this gesture. Since there are several properties to be controlled, they have been grouped, and each group is controlled in a separate mode. There are 3 modes: a) navigation mode, b) isosurface mode, c) cut-planes mode. The D gesture of the left hand is used for switching between modes. We have found, that assigning controls to both hands simultaneously causes too much confusion for the user, therefore the properties in each mode are controlled by the right hand only. The mapping of the gestures for the right hand is presented in table 1.

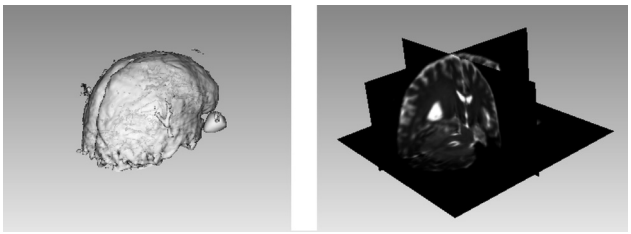

Figure 4. Brain visualization – isosurface (left) and cut-planes (right)

Table 1. Mapping of the gestures (A-D) to visualization controls in different modes (right hand)

	Navigation	Isosurface	Cut-planes
A	idle	idle	idle
B	zooming	opacity	opacity
C	translation	isovalue	position
D	rotation	isovalue	window / level

Tests and Results

The acquired database consisted of 200 images of the 4 gestures (50 images per gesture). The dataset has been divided

Table 2. Comparison of classifiers accuracy (NB – Naive Bayes, RF – Random Forest, SVM – Support Vector Machine)

NB	RF	SVM
83%	90%	95%

Table 3: Confusion matrix for the Support Vector Machine classifier

	A	B	C	D
A	**1,00**	0,00	0,00	0,00
B	0,00	**1,00**	0,00	0,00
C	0,00	0,00	**0,80**	0,20
D	0,00	0,00	0,00	**1,00**

into training set (40 images per gesture) and test set (10 images per gesture). Parameters of the classifiers, described further, have been selected based on 5 fold cross-validation of the training set.

As stated in previous section, several classifiers have been employed. Naive Bayes is a relatively simple, statistical-based classifier, which has proved to be the least accurate (83%). Random Forest classifier builds a number of decision trees, where each node is a randomly selected feature. The number of trees n and maximum tree depth m are parameters which have to be fine-tuned for specific problem. Both parameters have been tested in range from 1 to 20. The best accuracy for the Random Forest classifier (90%) has been achieved for n =10 and m = 3.

Support Vector Machine is a classifier that constructs a hyperplane between classes and maximizes the margin to data points of each class. This process is controlled by the regularization parameter c. Non-linear classification is achieved with Support Vector Machine by utilizing kernels. The Radial Basis Function kernel has been employed, in which influence of a single example is controlled by the parameter $gamma$. The c parameter has been tested in range 0.01 to 1000 with x3 step, and the $gamma$ parameter has been tested in range 0.01 to 10, also with x3 step. The best accuracy for the SVM classifier (95%) has been achieved for c = 1000 and $gamma$ = 0.03.

Table 2 shows comparison of the results from all classifiers. Table 3 presents confusion matrix for the Support Vector Machine classifier, which has achieved the best accuracy. The gestures are denoted A-D as in the figure 2. Three out of four gestures are always recognized correctly. Gesture C is sometimes confused with gesture D which is probably due to the fact that in this gesture it is natural to put the thumb a little behind the fingers, what sometimes results in a gap in the depth map between the thumb and the rest of the hand. When additionally the distance between the hand and the sensor is greater than usual, very few pixels of the thumb part remain after the noise reduction step, therefore they have little influence on the descriptor.

Conclusions

An interface for 3D medical data analysis has been devised, based on hand gesture recognition. The system employs a time-of-flight camera and can recognize 4 gestures per hand, which are sufficient to control a 3D visualization. Shape Context descriptor has been utilized and modified to fit various classifiers. Best achieved recognition accuracy is 95%, with the Support Vector Machine classifier. The recognition method runs in real-time on a typical PC machine. Proposed solution allows for convenient analysis of 3D medical data and addresses the problem of muscle fatigue by utilizing a relatively comfortable hands position.

Further work will include improving the accuracy and extending the analysis capabilities. The accuracy may be improved by utilizing a different descriptor or enhancing the current one. The analysis capabilities can be extended by providing additional visualization modes.

Acknowledgements

This research was partially supported by AGH-UST grant no. 11.11.230.124.

Bibliography

[1] Li Y. T., Wachs, J. P. „HEGM: A hierarchical elastic graph matching for hand gesture recognition.„ *Pattern Recognition*, 47(1), 80-88, 2014

[2] Premaratne P., Ajaz S., Premaratne M., „Hand gesture tracking and recognition system using Lucas–Kanade algorithms for control of consumer electronics.„ *Neurocomputing*, 116, 242-249, 2013

[3] Stergiopoulou E., Papamarkos N., „Hand gesture recognition using a neural network shape fitting technique.„ *Engineering Applications of Artificial Intelligence*, 22(8), 1141-1158, 2009

[4] Li H., Yang L., Wu X., Xu S., Wang Y. „Static hand gesture recognition based on hog with Kinect.„ *In Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2012 4th International Conference on* (Vol. 1, pp. 271-273). Aug. 2012

[5] Ren Z., Yuan J., Meng J., Zhang Z. „Robust part-based hand gesture recognition using Kinect sensor.„ *IEEE Transactions on Multimedia*, 15(5), 1110-1120, 2013

[6] Yao Y., Fu Y., „Contour model based hand-gesture recognition using Kinect sensor.„ *IEEE Transactions on Circuits and Systems for Video Technology*, 99, 1-10, Jan 2014

[7] Dominio F., Donadeo M., Zanuttigh P., „Combining multiple depth-based descriptors for hand gesture recognition.„ *Pattern Recognition Letters*, 2013

[8] Suau X., Alcoverro M., López-Méndez A., Ruiz-Hidalgo J., Casas J.R., „Real-time Fingertip Localization Conditioned on Hand Gesture Classification.„ *Image and Vision Computing*, 2014

[9] Li Z., Jarvis R., „Real time hand gesture recognition using a range camera.„ *In Australasian Conference on Robotics and Automation* (pp. 21-27), Dec. 2009.

[10] Breuer P., Eckes C., Müller S., „Hand gesture recognition with a novel IR time-of-flight range camera–a pilot study.„ *In Computer Vision/Computer Graphics Collaboration Techniques* (pp. 247-260). Springer Berlin Heidelberg, 2007

[11] Belongie S., Malik J., Puzicha J., „Shape context: A new descriptor for shape matching and object recognition.„ *In NIPS* (Vol. 2, p. 3), Nov. 2000.
[12] Breiman L. „Random forests„. *Machine learning*, 45(1), 5-32., 2001
[13] Cortes C., Vapnik V. „Support-vector networks.„ *Machine learning*, 20(3), 273-297, 1995

***MSc. Eng. Filip Malawski** - is a PhD candidate at AGH University of Science and Technology in the Department of Computer Science. He received his MSc degree in computer science at AGH University in 2012. His research interests include computer vision, image processing and human-computer interaction.*