Gesture recognition with a time-of-flight camera

Eva Kollorz, Jochen Penne and Joachim Hornegger

Institute of Pattern Recognition,
Friedrich-Alexander-University Erlangen-Nuremberg, Germany
E-mail: Eva.Kollorz,Jochen.Penne@informatik.uni-erlangen.de

Alexander Barke

Department of Applied Research,
Audi Electronics Venture GmbH, Gaimersheim, Germany
E-mail: alexander.barke@audi.de

Abstract: This paper presents a new approach for gesture classification using x- and y-projections of the image and optional depth features. The system uses a 3-D time-of-flight (TOF) sensor which has the big advantage of simplifying hand segmentation. For the presented system, a Photonic-Mixer-Device (PMD) camera with a resolution of 160 × 120 pixels and a frame rate of 15 frames per second is used. The goal of our system is to recognise 12 different static hand gestures. The x- and y-projections and the depth features of the captured image are good enough to use a simple nearest neighbour classifier, resulting in a fast classification. To evaluate the system, a set of 408 images is recorded, 12 gestures from 34 persons. With a 'Leave-One-Out' evaluation, the recognition rate of the system is 94.61 %, and classification time is about 30 ms on a standard PC.

Keywords: time-of-flight camera; projection features; gesture recognition

Reference to this paper should be made as follows: E. Kollorz, J. Penne, J. Hornegger and A. Barke, 'Gesture recognition with a time-of-flight camera'

1 Introduction

In recent years, the functional range of technical devices has become more and more complex. To ensure the usability of complex devices, buttons are replaced by other human machine interfaces (HMIs). One very common system is speech recognition. This HMI uses the speech of the user to execute the desired function. The system either analyses the entire speech of the user and extracts the relevant keywords, or the user may use special keywords to control the system. A recent type of HMI is gesture recognition. Such systems use hand gestures to control functions.

This paper describes a camera based gesture recognition system for automotive applications. Due to the special needs of automotive systems, the processing time must be under 70 ms (real-time). Some recent works about gesture recognition are presented in section 2. Section 3 describes the algorithm used in this paper, specifically the employed x- and y-projections. The experimental results are shown in section 4. Finally, section 5 gives some ideas about future work.

2 State of the art

One of the first gesture recognition systems was presented by Maggioni and Röttger (1999) at Siemens. This development uses a video projector which displays a user interface onto any surface and a video camera to capture the hand of a user. By moving the hand, the user can move objects on the projected desktop. This interface uses dynamic hand movements to control the system. Another possibility are lexical gesture based systems. Such systems use static hand gestures to control the system. A static gesture based system was presented by BMW and is described in Akyol *et al.* (1999). The system is used to control the infotainment inside a vehicle. For image capturing, a standard webcam is employed which makes segmentation very complex compared to a TOF camera.

The University of Bielefeld developed a camera based system for static gestures based on neural networks. It is called Gesture REcognition based on FInger Tips (GREFIT) as described in Noelker and Ritter (2002). The system works in two steps. First, the position of the finger tips is calculated in global coordinates by using neural networks. In the second step the gesture is reconstructed by a model of the hand. Another work is presented by Athitsos and Sclaroff (2003). They use a model based approach to classify 26 static gestures. For each gesture a set of 4128 different views is stored. Hence, the complete database consists of 107328 pictures. The captured image is compared to the stored images from the database.

Chang *et al.* (2002) presents a feature extraction based approach based on Curvature Scale Space (CSS) for translation, scale, and rotation invariant recognition of hand gestures. The CSS image is used to represent the shapes of boundary contours of hand gestures. Nearest neighbour techniques are used to perform CSS matching between the multiple sets of input CSS features and the stored CSS features for hand gesture recognition. For six gestures and 300 sets of data, the recognition rate is 98.3 %. All systems presented so far use standard cameras for image recording. These cameras do not provide depth information about the scene.

A recent work is presented by Breuer *et al.* (2007). They describe a system based on a Swissranger TOF camera which provides depth information. To reconstruct the hand, a principle component analysis and a special hand model are used. In the current version, they reconstruct the first seven degrees of freedom of the hand with a frame rate of about 3 Hz, which is not enough for our desired application which requires a response time under 70 ms. The recognition rate cannot be compared directly, because the system does not classify different gesture classes but reconstructs the shape of the hand.

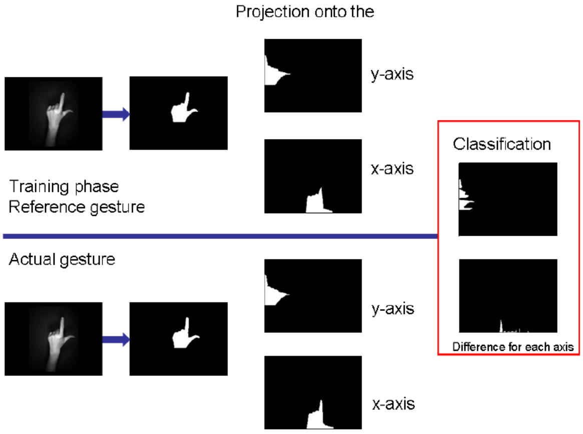

Figure 1 Overview of the introduced algorithm: extraction of the hand, calculation of the depth features on the segmented hand without contour, then projection onto the image axes and classification.

3 Algorithm

In this section, we introduce our approach to the recognition of different gestures with a TOF camera. The basic idea is to use simple and computationally cheap features for an efficient classification. The gestures used in the system (see Fig. 3) show a good separation potential along the two image axes. Hence, the projections of the hand onto the x- and y-axis are used as features for the classification. The region of the arm is discarded since it contains no useful information for the classification and due to strong variation between human beings. Additionally, depth features are included to distinguish certain gestures: gestures which have same projections, but different alignments. For example, gesture L is performed with the thumb towards the camera and gesture G is a fist, but both have similar projections. In these cases, the depth features provide useful information. The computation of features in 2-D with additional z-values is cheaper than reconstructing the hand in 3-D and performing there the feature extraction there. Fig. 1 gives an overview of the introduced algorithm.

The algorithm can be divided into five steps:

1. **Segmentation of the hand and arm via distance values:**
   The hand and arm are segmented by an iterative seed fill algorithm. The distance range for the segmentation of the hand and arm can be adjusted during design. If more is segmented than the hand and arm, e.g. background or other subjects, the classification is aborted. Otherwise the hand needs to be extracted, because features will only be calculated for the region of the hand. The region of the arm will negatively influence the projections as well as the depth features.
2. **Determination of the bounding box:**
   The segmented region is projected onto the x- and y-axis to determine the bounding box of the object (Hornegger and Niemann (2000)). In the following, projections onto the x-axis will be referenced as p_x, projections onto the y-axis as p_y (see Fig. 2). The resolution of the camera is denoted $M \times N$.

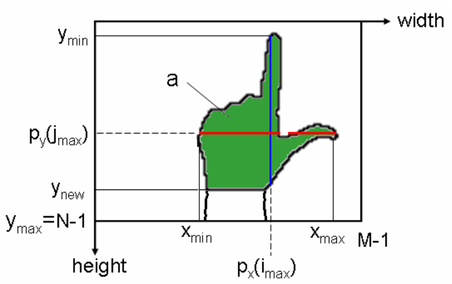

Figure 2 Drawing to clarify the used variables; shaded area a: segmented hand without the boundary pixels (contour); p_y: projection onto the y-axis (height); p_x: projection onto the x-axis (width); i_{max}: index of the maximal projection value of p_x; j_{max}: index of the maximal projection value of p_y; y_{min}, y_{max}: minimal and maximal border of p_y; x_{min}, x_{max}: minimal and maximal border of p_x.

The borders are determined by the leftmost and rightmost projection value greater than a threshold θ. The resulting borders are denoted as x_{min} and x_{max} for the minimal and maximal index of p_x and y_{min}, y_{max} for p_y. By means of the indices, we can recognise at which side of the bounding box the arm enters the field of view. We assume for this paper that the arm enters the bounding box from the lower side ($y_{max} = N$-1).

3. **Extraction of the hand:**
It is assumed that the average hand has a length l [mm]. Hence, the index y_{new}, which specifies the cut-off position, is determined by projecting a "virtual" hand of length l at distance cog_z mm into the image plane (cog: centre of gravity of the segmented object; z: direction to the camera). The y-value is used as y_{new}. The computation with 12.0 mm focal length is as following:

$$y_{new} = y_{min} + \frac{12.0 \cdot l}{(cog_z \cdot 0.04)} \tag{1}$$

where 0.04 mm is the pixel dimension.
For the experiments, we made a few assumptions:

- The arm is not covered at the wrist by a pullover or a watch.
- The person can wear a ring.

The first point is important for cutting off the hand. The initial y_{ini} is nearly correct for gestures with a stretched hand, but if the hand forms e.g. a fist, y_{ini} has to be adjusted. If the height projection p_y fulfills $p_h(i) - p_h(i+\alpha) < \beta$ (i ranges from y_{ini} to y_{min}), y_{ini} is decreased by one until the criterion is not satisfied anymore. y_{new}, and y_{max} respectively, is the resulting y_{ini}. This criteria expresses the small bottleneck at the wrist.

4. **Projection of the hand region onto the x- and y-axis:**
As outlined above, the truncated hand is projected onto the x- and y-axis. These projections are smoothed with a median filter of kernel size three to eliminate outliers. The pre-processed projections are still denoted as p_x and p_y. For the truncated hand, additional depth features a_{min}, a_{max} and a_{avg} are computed: For each inner pixel of the segmented hand area a, a reliable z-value is measured. The minimum of this amount of z-values is described by a_{min}, synonymously for a_{max} and a_{avg}.

5. **Classification with majority decision over m frames and k nearest neighbour:**
For each gesture in the training set, the smoothed height and width projections are saved as well as the parameters a_{min}, a_{max} and a_{avg}. The current gesture to be classified is normalised to the reference (R) gestures' domain:

$$c = \frac{i - x_{\mathrm{min}}}{x_{\mathrm{max}} - x_{\mathrm{min}}} \cdot (x_{\mathrm{R,max}} - x_{\mathrm{R,min}}) + x_{\mathrm{R,min}}, \tag{2}$$

where i is a bin of the current width projection p_x, c the corresponding bin of the reference width projection $p_{x,\mathrm{R}}$ and $x_{\mathrm{R,min}}/x_{\mathrm{R,max}}$ the minimum/maximum of the $p_{x,\mathrm{R}}$. If c is a floating point number, an interpolation between these two bins is applied. The difference for the width is:

$$x_{\mathrm{diff}} = \sum_i \left| \frac{p_x(i)}{p_x(i_{max})} - \frac{p_{x,R}(c)}{p_{x,R}(i_{max,R})} \right|, \tag{3}$$

where $p_x(i_{\mathrm{max}})$ is the maximal observed value of the width projection p_x and $p_{x,R}(i_{max,R})$ that of $p_{x,R}$. The same is done for the height projection p_y to compute y_{diff}. The distance d for the current gesture to one of the reference gestures is computed as follows:

$$d = \left| \frac{a_{avg} - a_{min}}{a_{max}} - \frac{a_{R,avg} - a_{R,min}}{a_{R,max}} \right| + \frac{x_{\mathrm{diff}}}{x_{max} - x_{min}} + \frac{y_{\mathrm{diff}}}{y_{max} - y_{min}}. \tag{4}$$

The last two terms normalise the projections in the codomain.
The majority decision is first done for k nearest neighbour for each frame. The result is the class that occurs most often in the set of neighbours. For a tie, the minimal classification distance decides. Following this, the majority decision over m frames is done in the same manner.

The pseudo code shown in Algorithm 1 illustrates the approach.

4 Experimental results

In our scenario the segmented distance range for the seed fill algorithm is between 70 and 110 cm. For the computation of the bounding box, the threshold θ is set to 4 pixels, assuming that projection values which are smaller than 4 are unreliable changing border pixels, e.g. at the tip of the thumb, which can be neglected. The length l of an average hand was set to 210 mm. α is set to 3 and β is 2 for the criteria of the small bottleneck at the wrist.

The described algorithm was tested with a set of 12 static hand gestures, shown in Fig. 3. A processing time under 70 ms is required. As shown in Fig. 3, some of the 12 gestures are very similar. For example, gesture A and gesture J only differ in the position of the thumb. Gesture H and gesture K are also similar: if gesture H is rotated slightly clockwise it looks like gesture K. Other gestures, especially gesture L, differ from the others in their depth. For these gestures, the advantages of a TOF camera can be used. For the evaluation of the algorithm, a PMD-Sensor with a resolution of 160 × 120 pixels and a viewing angle of 40° was used. The distance between the hand and the sensor was between 70 cm and 110 cm.

To validate the algorithm, a data set of 34 persons, mixed male and female, was recorded. All 12 gestures were captured from every test person, generating 408

```
input  : Image seg with segmented hand and arm;
         1: segmented, 0: otherwise
output: Classification of the gesture

Calculation of the projections $p_x$ and $p_y$;
// Calculation of the bounding box: $x_{min}, x_{max}, y_{min}, y_{max}$;
for $x \leftarrow 0$ to $M-1$ do
    if $p_x(x) < 4$ then
        $x_{min} = x$;
    end
    if $p_x(M-1-x) < 4$ then
        $x_{max} = x$;
    end
end
Analogous for $p_y$;
Calculation of initial y value for cut off, see Eq. 1;
for $i \leftarrow y_{ini}$ to $y_{min}$ do
    if $p_y(i) - p_y(i+3) < 2$ then
        $y_{ini} -= 1$;
    end
end
$y_{max} = y_{new} = y_{ini}$;
Projection of the extracted hand onto the axes ($p_x$,$p_y$);
Smoothing of the projections with a median filter;
if seg(x,y)!=border then
    Calculation of the depth features;
end
foreach Reference gesture do
    Calculation of the distance $d$ between current and reference gesture,
    see Eq. 2, 3 and 4;
end
Classification with $k$ nearest neighbour and over $m$ frames;
```

Algorithm 1: Classification of gestures

images. For evaluation, only single images were classified and the classification was done with one nearest neighbour.

The 'Leave-One-Out' method was used, in which one person is excluded from the data set. The gestures of this person are tested with the gestures of the remaining 33 persons, i.e. 396 images. Table 1 shows the results of the evaluation without depth features. The row reflects the gesture made by the user, and the column the recognised gesture. For example, gesture A was performed 34 times and recognised 31 times as gesture A and three times as J. Especially gesture L is mistaken with gesture G. Another problem is gesture K, which is sometimes classified as gesture H and vice versa. The overall classification rate is calculated with

$$\frac{\sum_{i=1}^{\#gestures} a_{ii}}{\#persons \cdot \#gestures}, \tag{5}$$

where a_{ii} is the number of correctly classified gestures in each class. The classification rate of the evaluation as shown in Table 1 is 93.14 %. Using the advantages of the TOF camera, information about distances can be used for classification. Spe-

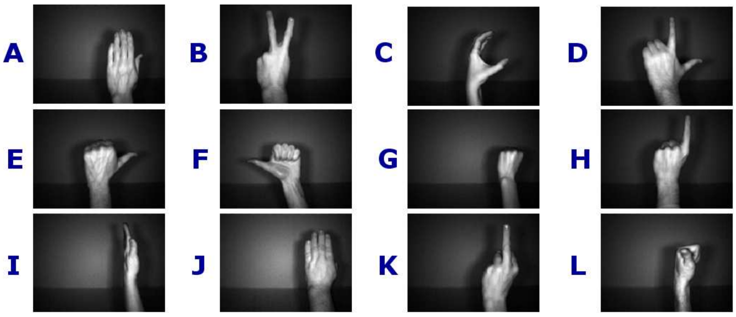

Figure 3 The 12 gestures recognised by the system.

	A	B	C	D	E	F	G	H	I	J	K	L
A	31	0	0	0	0	0	0	0	0	3	0	0
B	0	32	0	0	0	0	0	0	0	0	2	0
C	0	0	33	0	0	0	1	0	0	0	0	0
D	0	0	0	34	0	0	0	0	0	0	0	0
E	0	0	0	0	33	0	1	0	0	0	0	0
F	0	0	0	0	0	34	0	0	0	0	0	0
G	1	0	0	0	0	0	32	0	0	0	0	1
H	0	0	0	0	0	0	0	32	0	0	2	0
I	1	0	0	0	0	0	0	0	32	0	1	0
J	2	0	0	0	0	0	1	0	0	31	0	0
K	0	0	0	1	0	0	0	3	0	0	30	0
L	0	0	0	0	0	2	4	0	0	2	0	26

Table 1 Results of classification without depth features.

	A	B	C	D	E	F	G	H	I	J	K	L
A	31	0	0	0	0	0	0	0	0	3	0	0
B	0	32	0	0	0	0	0	0	0	0	2	0
C	0	0	33	0	0	0	1	0	0	0	0	0
D	0	0	0	34	0	0	0	0	0	0	0	0
E	0	0	0	0	33	0	1	0	0	0	0	0
F	0	0	0	0	0	34	0	0	0	0	0	0
G	1	0	0	0	0	0	32	0	0	0	0	1
H	0	0	0	0	0	0	0	33	0	0	1	0
I	0	0	0	0	0	0	0	0	33	0	1	0
J	4	0	0	0	0	0	0	0	0	30	0	0
K	0	0	0	0	0	0	0	4	0	0	30	0
L	0	0	0	0	0	1	1	0	0	1	0	31

Table 2 Results of classification with depth features.

cial depth features can improve the recognition rate. Especially for gesture L, these features are useful because this gesture has different distances to the camera. Table 2 shows the results of the evaluation with depth features. The greatest difference is the increased classification rate of gesture L, as expected. Most of the other results are nearly the same since the other gestures are planar to the camera and have no differing distances. The overall classification rate with depth features is 94.61 %, about 1.5 % higher than without. With a standard PC (Core Duo 2.4 GHz, 1 GB RAM), 15 ms are needed for segmentation and 15 ms for classification. This is fast enough for a realtime application where a response time of 70 ms is required.

5 Conclusion and outlook

This paper presents a gesture recognition system based on a TOF camera with a resolution of 160 × 120 pixels. 12 different static hand gestures are to be classified. The advantage of the additional depth values and the disadvantage of the low resolution are especially considered. Some of the gestures are also very similar to each other; another challenge to the system. Single images are used for classification. The segmentation is done by a seed fill algorithm on depth values. After this, the x- and y-projection and the minimum and maximum distance of the hand are calculated. For classification, a nearest neighbour classifier is used.

To evaluate the system, a set of 408 gestures performed by 34 different people is used. Evaluation is done with the 'Leave-One-Out' method. The overall classification rate is 93.14 % without and 94.61 % with depth features. One way to increase the recognition rate is to use more than one image for classification. A calculation time of 30 ms is fast enough to use two images for classification. Future work will be testing other classification algorithms like neural networks or self-organizing maps. Another step is to improve the features, especially the depth features.

References

Akyol, S. and Canzler, U. and Bengler, K. and Hahn, W. (1999) 'Gestensteuerung für Fahrzeugbordsysteme', *Informatik aktuell. Mustererkennung 2000. 22. DAGM Symposium*, pp.139–146, 13-15 September, Kiel, Germany, G. Sommer, N. Krüger and Ch. Perwass (Eds.), Springer Verlag

Athitsos, V. and Sclaroff, S. (2003) 'Estimating 3D Hand Pose from a Cluttered Image', *Proceedings of the 2003 International Conference on Computer Vision and Pattern Recognition*, Vol. 2, pp.432–439, 16-22 June, Madison, USA

Breuer, P. and Eckes, C. and Müller, S. (2007) 'Hand Gesture Recognition with a novel IR Time-of-Flight Range Camera - A pilot study', *Proceedings of Mirage 2007, Computer Vision / Computer Graphics Collaboration Techniques and Applications*, pp.247–260, March 28-30, Rocquencourt, France

Chang, C.-C. and Chen, I.-Y. and Huang, Y.-S. (2002) 'Hand Pose Recognition Using Curvature Scale Space', *Proceedings of the 16th International Conference on Pattern Recognition*, Vol. 2, pp.386–389, 11-15 August, Quebec City, Canada

Hornegger, J. and Niemann, H. (2000) 'Probabilistic Modeling and Recognition of 3-D Objects', *International Journal of Computer Vision*, Vol. 39, No. 3, pp.229–251, September 2000

Maggioni, C. and Röttger, H. (1999) 'Virtual Touchscreen - a novel User Interface made of Light - Principles, metaphors and experiences', *Proceedings of HCI International on Human-Computer Interaction: Ergonomics and User Interfaces-Volume I*, Vol. 1, pp.301–305, 22-27 August, Munich, Germany

Nölker, C. and Ritter, H. (2002) 'Visual Recognition of Continuous Hand Postures', *IEEE Transactions on Neural Networks*, Vol. 13, No. 4, pp.983–994, July 2002