

Sparse Features for Finger Detection

Simion Georgiana

Faculty of Electronics and Telecommunications

“POLITEHNICA” University from Timișoara

Timișoara, Romania

email: georgiana.simion@etc.upt.ro

Keywords: edges, line, Hough transform, finger detection, hand posture.

Abstract. The use of hand as direct input device evolved with the development of Natural User Interfaces (NUI). Touch screens are well integrated in our daily life and the new challenge is to implement interfaces which involve no direct contact. This paper presents such a solution implemented within the framework of sparse techniques. The feature vectors representing key distances and angles are extracted and used to detect fingers. The experimental results have demonstrated that this technique is able to obtain an error rate about 5% in finger detection.

Introduction

The use of hand as an interfacing method has become more and more popular in the last years. Nowadays we use hand as a direct input device, we have touch screens in our pockets embedded in smart phones with no keys, touristic information points based on touch screens, but the ultimate frontier is to use hand gestures without direct contact to interact with the surrounding devices. The Computer Vision based techniques can offer such non-contact solutions. These solutions have applications in diverse fields from virtual reality to healthcare, from vehicles interfaces to desktop and tablet PC applications. Some of the cutting edge technologies employed by these solutions are briefly presented in the following. Some of the most promising technologies are represented by the 3D acquisition solutions based on either structured light approach [1, 2] or Time-of-Flight (TOF) principle [3, 4, 5]. The big advantage of these approaches is simplifying hand segmentation. The electrical-field-based 3D gesture controller recently patented by Microchip Technology Inc. [6] offers low-power, precise, fast and robust hand position tracking with free-space gesture recognition.

Although promising, these technologies still have some drawbacks, e.g. for the 3D cameras, the cost is still prohibitive and for the e-filed solutions the operation range is relatively small.

The proposed framework uses a sparse representation for the hand fingers. Our goal is to detect the fingers in order to recognize some gesture. Patches, atoms, salient points, interest points, edges [7, 8, 9, 10, 11] are representative features for sparse representation. It has been proven that edges have a very important role in humans' visual processing system. Humans are able to recognize people from sketches because this contains enough useful information for our brain view-processor.

The remainder of this paper is organized as follows. Section II describes the theoretical backgrounds. The proposed algorithm for finger detection is detailed in Section III. Section IV shows the experimental results and Section V concludes the paper.

Theoretical Backgrounds

Canny edge detector

This work is based on hand edge detection. The framework diagram can be seen in figure 1. In order to detect the hand contours, in this work Canny's approach to optimal edge detection was followed.

In his work [12], [13], Canny specifies three performance criteria that an edge detector should fulfill. The criteria Canny uses to detect sharp edges can be summarized as follows:

Good detection: Edge detection should be robust to noise; it means that should be a high probability to mark real edge points and a low probability to falsely mark non edge points. This criterion can be formalized [12], [13] by requiring the edge detector to maximize the output signal to- noise ratio (SNR) for a given input signal-to-noise ratio.

Good localization: The points considered located on an edge should be as close as possible to the center of the true edge. This criterion is formalized in [12] by requiring that the location of the maximal detector response to an edge have a low variance.

Uniqueness of response: The edge detector should produce only one response to a single edge. This criterion can be formalized by requiring the filter which performs edge detection to have a small spatial width [13].

Hough Transform

The Hough Transform was developed as it is used nowadays by Duda and Hart [14] and it represents a technique which can be used to isolate features of a particular shape within an image. Because it requires that the desired features be specified in some parametric form, the classical Hough transform is commonly used for the detection of regular curves such as lines, circles and ellipses. Frequently an edge detector is used in a pre-processing stage to obtain image points or image pixels that are on the desired curve in the image space. Due to imperfections in either the image data or the edge detector, however, there may be missing points or pixels on the desired curves as well as spatial deviations between the ideal line/circle/ellipse and the noisy edge points as they are obtained from the edge detector. For these reasons, it is often non-trivial to group the extracted edge features to an appropriate set of lines, circles or ellipses. The purpose of the Hough transform is to address this problem by making it possible to perform groupings of edge points into object candidates by performing an explicit voting procedure over a set of parameterized image objects.

The Hough transform uses the line equation written in the Polar system. The line equation can be written according to Eq. 1:

$$y = \left(-\frac{\cos \theta}{\sin \theta} \right) x + \left(\frac{r}{\sin \theta} \right) \quad (1)$$

From Eq.1, r can be written as:

$$r = x \cos \theta + y \sin \theta \quad (2)$$

For each point (x_0, y_0) a family of lines that goes through that point can be defined as in Eq.2:

$$r_\theta = x_0 \cos \theta + y_0 \sin \theta \quad (3)$$

Where each pair (r_θ, θ) represents each line that passes by (x_0, y_0) .

If for a given point (x_0, y_0) the family of lines that goes through it are plotted, the result is a sinusoid. Then only points for which $r > 0$ and $0 < \theta < 2\pi$ are considered. This operation can be performed for all the points in an image. This *point-to-curve* transformation is the Hough transformation for straight lines. When viewed in Hough parameter space, points which are collinear in the Cartesian image space become readily apparent as they yield curves which intersect at a common (r, θ) point. It means that in general, a line can be detected by finding the number of intersections between curves. The more curves intersecting means that the line represented by that intersection have more points. In general, we can define a threshold of the minimum number of

intersections needed to detect a line. The transform is implemented by quantizing the Hough parameter space into finite intervals or accumulator cells. As the algorithm runs, each (x_i, y_i) is transformed into a (r, θ) discretized curve and the accumulator cells which lie along this curve are incremented. Resulting peaks in the accumulator array represent strong evidence that a corresponding straight line exists in the image.

The proposed algorithm

Our work is based on hand edge detection and line detection. The framework diagram can be seen in Fig. 1.

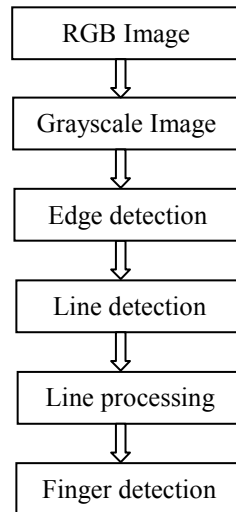


Fig. 1 The framework diagram

First the RGB image is converted in a grayscale image in order to extract the hand contour using Canny edge detector. Once the hand contour is extracted, the Hough transform is used to detect the strait lines on the hand contour. Initially a set of segments are detected. Not all segments detected by the Hough transform are connected, even if there are belonging to the same line. The algorithm checks if two segments are on the same line and if there are, connects the two segments.

In order to connect two segments that belong to the same line, the midpoint of each line segment is computed. The midpoints are connected and a new line segment is created. The angle of each line segment is also computed. In the next step, the difference between the line segments angles and the length of the new line segment are processed. If these values are smaller than the predefined thresholds σ, τ , the two segments are connected.

For all segments, the length is computed. The ratio between two segments is also evaluated. If this ratio is smaller than α the considered segments are good candidates to define a finger. In order to evaluate the candidate segments, several operations are performed.

The midperpendicular of the smaller segment is drawn and the intersection point with the other segment is marked, as it is illustrated in Fig. 2 .

The Eq. 4 corresponds to the line defined by points A and B.

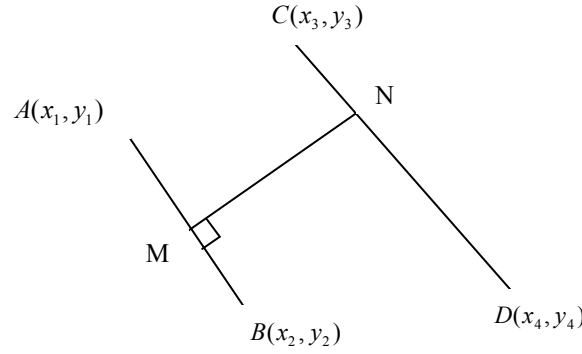


Fig. 2 A pair of two lines, the midperpendicular of the smaller line segment intersects the other line segment.

$$\frac{x - x_1}{x_2 - x_1} = \frac{y - y_1}{y_2 - y_1} \quad (4)$$

$$(y_1 - y_2) \cdot x + (x_2 - x_1) \cdot y + (x_1 y_2 - x_2 y_1) = 0 \quad (5)$$

The slope of line defined by points A and B is computed according to Eq.6.

$$m_{AB} = \frac{y_1 - y_2}{x_1 - x_2} \quad (6)$$

The slope of [MN] is computed using Eq. 7.

$$m_{MN} = \frac{x_2 - x_1}{y_1 - y_2} \quad (7)$$

Eq. 8 describes the line segment [MN].

$$y_M = \frac{x_2 - x_1}{y_1 - y_2} \cdot x_M + n \quad (8)$$

In order to compute the coordinates of point N, which represents the intersection of midperpendicular of segment [AB] and line segment [CD], the following equation system must be solved.

$$\begin{cases} y = \frac{x_2 - x_1}{y_1 - y_2} \cdot x + n \\ (y_3 - y_4) \cdot x + (x_4 - x_3) \cdot y + (x_3 y_4 - x_4 y_3) = 0 \end{cases} \quad (9)$$

Once the values of these coordinates are available, it is possible to determine the length of the line segment [MN].

Two lines define a finger if the angle between them is smaller than β and the distance between two lines d , is in a certain range.

Experimental Results

In order to test the algorithm, 100 images were used. All this images represents hand postures in which all five fingers are shown.



Fig. 3 An example of grayscale image.

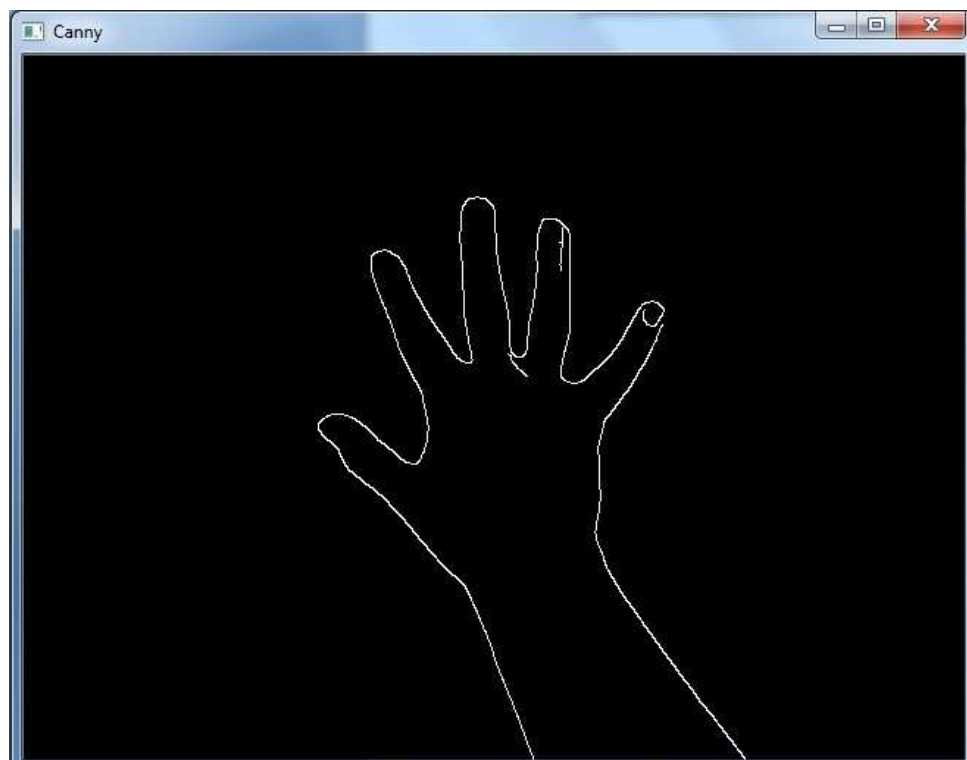


Fig. 4 Hand contour extracted using Canny edge detector.

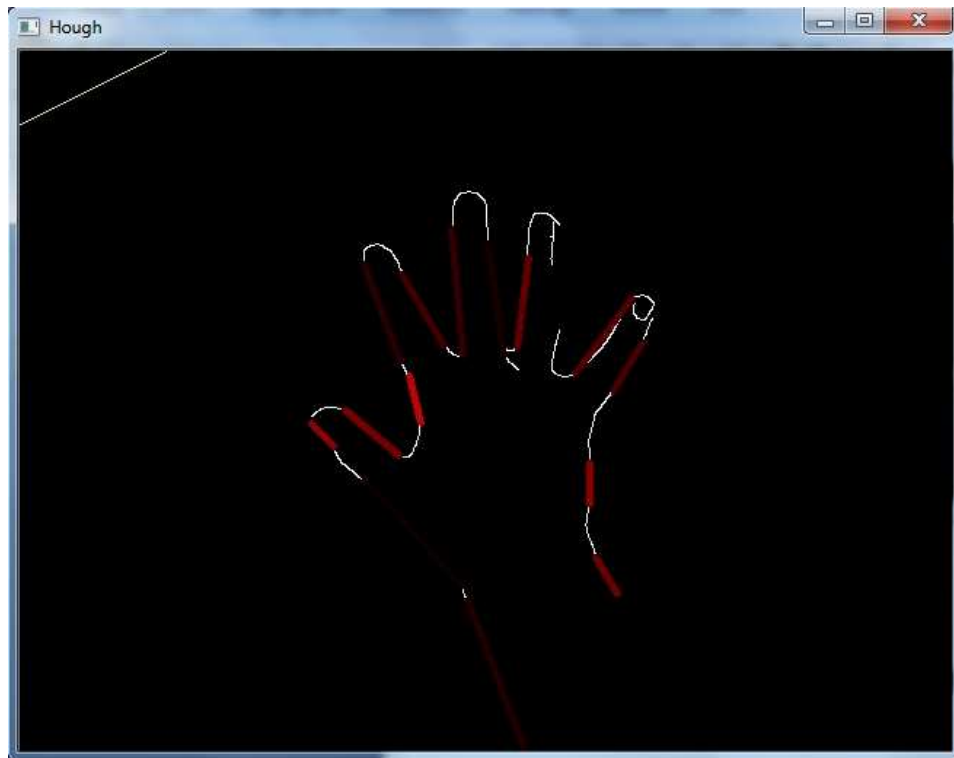


Fig. 5 Lines extracted using the Hough Transform.

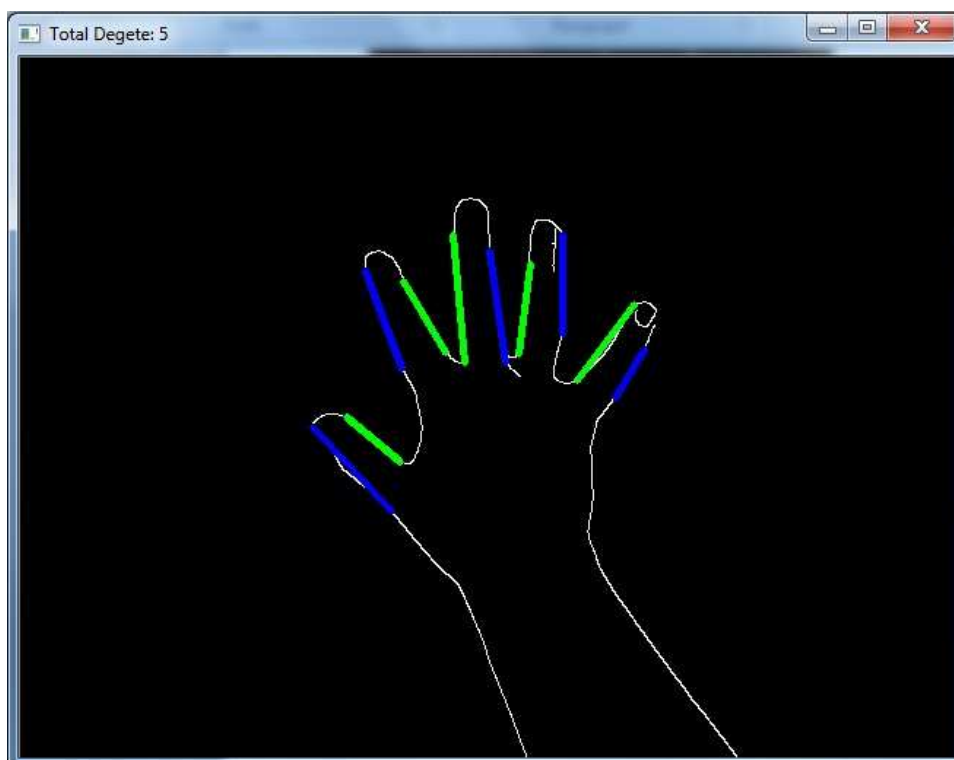


Fig. 6 Fingers detection.

These pictures consist in hand postures indoor taken, while no artificial light was added. The pictures resolution is 640×480 . The finger detection rate was about 95%. The proposed algorithm was implemented using OpenCV 2.4 library (the latest available release) and compiled using Visual Studio 2010. The value for the parameter σ is set to be smaller than 15 degrees and $\tau < 60$ pixels. Parameter α was set to 2.5 and $\beta < 20$ and $d \in [5, 40]$.

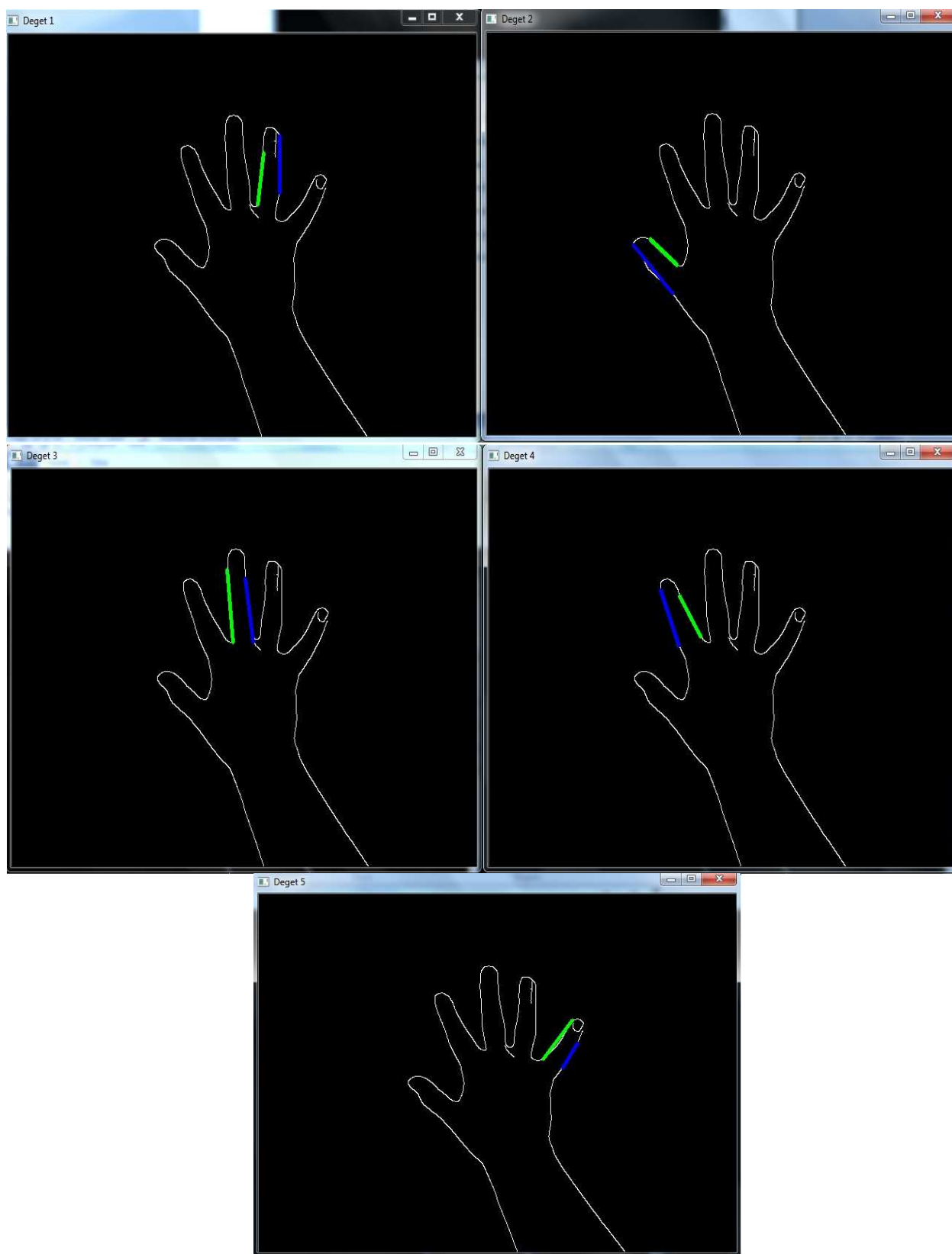


Fig. 7 Fingers illustration as they were detected.

Conclusions

In this paper, a method that detects the hand fingers was described. The hand contour is first extracted using Canny edge detector. In the next stage, the Hough transform is applied to extract the line segments from the hand contour. These line segments are analyzed and if it is concluded that they belong to the same line, a connection operation is performed. Pairs of two line segments are

processed, the distance and the angle between them is computed. Based on these values it is decided if they represent a finger. The method was tested on a set of 100 images and the detection rate was 95%.

ACKNOWLEDGMENT: This paper was supported by the project "Development and support of multidisciplinary postdoctoral programmes in major technical areas of national strategy of Research - Development - Innovation" 4D-POSTDOC, contract no. POSDRU/89/1.5/S/52603, project co-funded by the European Social Fund through Sectoral Operational Programme Human Resources Development 2007-2013.

References

- [1] S. Malassiotis, F. Tsalakanidou, N. Mavridis, V. Giagourta, N. Grammalidis, M.G Strintzis, A face and gesture recognition system based on an active stereo sensor, *Proceedings 2001 ICIP*, 3 (2001) 955–958.
- [2] M. Bray, Koller-Meier, L.V Gool, Smart particle filtering for 3D hand tracking, *Sixth IEEE International Conference on Automatic Face and Gesture Recognition*, (2004), 675-680.
- [3] P. Breuer, C. Eckes, S Muller, Hand Gesture Recognition with a novel IR Time-of-Flight Range Camera - A pilot study, *Computer Vision / Computer Graphics Collaboration Techniques and Applications*, (2007) 247–260.
- [4] E. Kollorz, J. Penne, J.Hornegger, A. Barke, Gesture recognition with a time-of-flight camera. *Int. J. Intell. Syst. Technol. Appl.* 5(3/4), (2008) 334–343.
- [5] J. Molina, M. Escudero-Viñolo, A. Signoriello, M. Pardás, C. Ferrán, J. Bescós, F. Marqués, J. Martínez, Real-time user independent hand gesture recognition from time-of-flight camera video using static and dynamic models. *Machine Vision and Applications* (2011) 1-18.
- [6] Information on <http://www.microchip.com/pagehandler/en-us/press-release/microchips-new-gestic-technolo.html>
- [7] B. Ommer, J.M. Buhmann, Learning the Compositional Nature of Visual, *CVPR'07*, IEEE, (2007).
- [8] B.Ommer, J.M. Buhmann, Learning Compositional Categorization Models *Objects ECCV'06*, LNCS 3953, Springer, (2006).
- [9] Li Fei-Fei, Pietro Perona “A Bayesian Hierarchical Model for Learning Natural Scene Categories” *EEE Comp. Vis. Patt. Recog.* (2005).
- [10] A. Oikonomopoulos, I. Patras, M. Pantic, Spatiotemporal Salient Points for Visual Recognition of Human Actions, *Ieee Transactions On Systems, Man, And Cybernetics—Part B: Cybernetics*, 36, (2006).
- [11] I. Laptev, T. Lindeberg, Space-time Interest Points, *Proceedings of the Ninth IEEE International Conference on Computer Vision*, (2003).
- [12] J. Canny, Finding edges and lines in images, *Tehnickal Reoprt AITR-720*, Massachusetts Institute of Technologie, (1983).
- [13] J. Canny, A computational approach to edge detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(1986) 679–698.
- [14] Information on <http://www.ai.sri.com/pubs/files/tn036-duda71.pdf>