

Fingertip-based Real Time Tracking and Gesture Recognition for Natural User Interfaces

**Georgiana Simion, Ciprian David, Vasile Gui,
Cătălin-Daniel Căleanu**

Politehnica University of Timișoara, Faculty of Electronics and Telecommunications, 2 Pârvan Av., 300223, Timișoara, Romania, {georgiana.simion, ciprian.david, vasile.gui, catalin.caleanu}@upt.ro

Abstract: The widespread deployment of Natural User Interface (NUI) systems in smart terminals such as phones, tablets and TV sets has heightened the need for robust multi-touch, speech or facial recognition solutions. Air gestures recognition represents one of the most appealing technologies in the field. This work proposes a fingertip-based approach for hand gesture recognition. The novelty in the proposed system is the tracking principle, where an improved version of the multi-scale mode filtering (MSMF) algorithm is used, and in the classification step, where the proposed set of geometric features provides high discriminative capabilities. Empirically, we conducted an experimental study involving hand gesture recognition – on gestures performed by multiple persons against a variety of backgrounds – in which our approach achieved a global recognition rate of 95.66%.

Keywords: fingertip detection; hand gesture recognition; image classification; image edge detection; object recognition

1 Introduction

The Human-Computer Interface (HCI) domain represents one of the most dynamic research fields. It started as CI (cards, tapes, character CRT) only to become more interactive (keyboards, joysticks, realistic rendering). Nowadays, a second revolutionary change in HCI awaits- “Natural User Interface (NUI)”. Among the key factors that contribute to the NUI revolution one must mention the availability of appropriate hardware platforms (CPU, GPU), existing software frameworks and, perhaps the most important thing, the user’s demand and acceptance. According to the NUI principles, the machines should be able to interpret and communicate close to the way people do, using multi-touch, gestures, speech and various other sensors [1].

The use of the hand as a NUI interfacing method has recently grown to be more and more popular. Currently, we use the hand as a direct input device, we have

touch screens in our pockets embedded in smart phones with no keypads, touristic information points based on touch screens. Now the ultimate frontier is to use hand gestures without a direct contact to interact with the surrounding devices. Computer vision-based techniques can offer such non-contact solutions.

Some of the cutting-edge technologies employed by these solutions are briefly presented in this research, of those, the most promising technologies are the 3D acquisition solutions based on either structured light approach [2-3] or the Time-of-Flight (TOF) principle [4-6]. One of the most important advantage of these approaches is simplifying the hand segmentation. The electrical-field-based 3D gesture controller recently patented by Microchip Technology Inc. offers a low power, precise, fast and robust hand position tracking with free-space gesture recognition. These solutions have applications in various fields from virtual reality to healthcare, and from vehicle interfaces to desktops and tablets.

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 functional range is relatively small.

In our proposed framework, the images are acquired using a 640 x 480 VGA notebook integrated webcam. The target application is a non-direct contact interface which can be used in an info-point. The hand is extracted using the background subtraction technique [7], which is based on a codebook representation. Fingertips are tracked using our previously developed algorithm based on MSMF. For the classification step a set of six gestures, easily performed by the user, is chosen. The suggested set of geometric features provides high discriminative capabilities. Empirically, we have conducted an experimental study involving hand gesture recognition – on gestures performed by multiple persons against various backgrounds – in which our approach has achieved a global recognition rate of 95.66%.

The remainder of this paper is organized as follows. Section II describes recent solutions in fingers and fingertips detection. Details regarding the fingertips tracker used within the context of our approach are presented in Section III. Section IV gives the gesture classification method. Section V shows the experimental results and Section VI concludes the paper.

2 Related Work

There are different ways to use hand gestures in order to build a NUI. In literature, one can find several articulated hand models like those employed by [3], [8-10], but the proposed model uses fingertips and the palm region. Fingers and fingertips have more simple and stable shapes and therefore can be easily detected and tracked more reliably than hands [11-16]. Thus, they are more suitable in

applications where hands are used for pointing [17], since occlusions are less frequent in such cases. A commonly used method to detect fingers and fingertips is the mathematical morphology and related solutions [18-20]. Some approaches use Hu invariant moments [21-22], others k-curvature algorithm [23], combined with template matching [11], [19]. Noise, scale, and hand direction have a significant impact on finding fingers and fingertips using skeleton [24] or dominant points [12] principles.

Color information [11-14] is also an important feature in finger tracking because it is used to detect the hand. Remarkably, skin hue has relatively low variation between different people and different races. In addition, it is invariant to moderate illumination changes. For objects that are achromatic or have low saturations, the hue information is unreliable and influenced by the illumination level. The methods based on skin color may fail under complicated or dynamic lighting although motion information, gradient or 3D based solutions [15-16], [25-27] alleviate some of the drawbacks.

Another key element in finger-based approaches is edge detection [28]. If edge detection was not so much affected by complex backgrounds or hand texture, the hand shape could be well defined. Yet, some improvement in the process of edges selection is expected because of the post-processing operation.

Using the detected edges or segmentation, the hand contour can be obtained. It is the case of the following approaches [25], [29-30]. Unfortunately, most of them were tested against uniform backgrounds and proved highly sensitive to the hand posture. For better results, the segmentation operation could be augmented by hand tracking and recognition. In [31] the hand contour is extracted using color information and motion information whereas the skin regions, which are in movement, are considered part of the hand region. The fingertips are extracted using high values of curvature.

The conclusion drawn in [19] is that gestures with “one” or “two” fingers are the most reliably recognized. Using two fingers or just one finger, a large set of gestures can be defined and recognized. For example, in the work [32] a robust method for defining dynamic single finger-based gesture is proposed. This shows that the set of meaningful gestures that can be defined is large enough to create a comprehensive gesture dictionary. However, the proposed dynamic gesture recognition algorithm does not exploit the full potential of the tracking system used. In [32] only one active finger gesture is used for recognition. In contrast, the present work proposes extending the usage of the tracking system to perform two or more finger-based gesture recognition. We also avoid the complex problems of model definition and elaborated classification scheme typically employed in gesture recognition [33] by designing a geometrical approach, which brings important improvements in the frame processing speed.

3 The Tracker Principle

This section contains a brief overview of the used finger tracking method [34]. The proposed approach offers significant improvements with respect to real-time capabilities in processing the foreground and environment variations (indoors, outdoors, dynamic background, illumination changes).

Based on the background subtraction technique from [7] the foreground objects are extracted, resulting in a considerable reduction of the processing data. A codebook representation of the background model is used to quantize the background model at each pixel location. The model is adaptable to illumination changes. The background model based on codebook representation allows for the fast update of the model, being easy to adapt when dealing with dynamic backgrounds.

In the next stage, the finger features are extracted. These finger features are line strips, obtained by the horizontal and vertical scanning of the segmented foreground. We use both horizontal and vertical scans in order to achieve rotation invariance of the hand tracking system.

Looking for similarities in length and thickness, and considering the centers of these segments, the algorithm identifies the fingers' positions. The slope of the line containing the identified strip centers is then computed. First, the strips are filtered using neighboring information, and afterwards are clustered in order to detect the fingers. The clustering method used in our work is the multi-scale mode filtering (MSMF) [35] algorithm, which in fact is our multi-scale version of the Mean Shift (MS) algorithm. In MSMF algorithm, the scale parameter varies during iterations, allowing a course to fine tuning approach. This strategy proved to be beneficial in avoiding false maxima. The final point reached is the maximum point of probability density and its location is our solution.

As an important improvement, the tracking algorithm presented in [34] was extended to detect also the palm of the hand. Basically, palm detection and tracking work mainly the same as the finger tracking algorithm with a difference in the scale of the extracted line. As noticeable next, palm detection is very important in our approach, as it allows defining a supplementary point that is used to design the final set of natural gestures to be recognized.

This approach [34] is based on sparse features and it is part of the recent direction in tracking, known as tracking by detection [36-37]. Using sparse features provides computational efficiency, robustness and has high accuracy. The clustering algorithm, the careful selection of the finger features and the shape information regarding the fingers enhance the robustness of this tracking method.

We demonstrated the robustness and the dynamical accuracy of the tracker against occlusion in [34].

4 Gesture Classification

The envisaged NUI uses six simple gestures, easily performed by any user. These gestures were carefully chosen, taking into consideration the anatomic features of the hand and also the envisaged touchless info-point application. Even if the number of the defined gestures is not that large, still they can be versatilely used for multiple actions. Using these gestures, a virtual mouse could be easily implemented. All these gestures imply the use of two fingers only.

The six gestures definitions are in the next section. The two fingertips based gestures newly introduced offer a convenient and natural way to manipulate the content of the aimed application. Contrasting, the three fingertips based gestures are more difficult to be performed and the number of gestures that can be defined is not superior due to the lack of accuracy in defining the geometric distances and angles.

The algorithm initially detects all fingertips, except the thumbs' fingertip, and the palm of the hand as shown in Figure 1. The thumb is not of interest in the proposed context of dynamic hand gesture recognition because it is not used to gesticulate. Besides, its relative position to the palm area is difficult to estimate.



Figure 1

The fingertip detection stage

Context awareness is also used in our approach: once two fingertips are detected, the user can start gesticulating. The user knows when to start performing a gesture (a start message is printed on the screen, as shown in Figure 2. The two fingertips must be detected in at least three consecutive frames in order to start the routine that classifies the gesture. The algorithm computes several distances and angles as detailed below.



Figure 2

The system is ready for interpreting gestures

The first distance, denoted by $dist1$, is the distance between the index fingertip and the center of the palm (1); the second distance, denoted by $dist2$, represents the distance between the middle fingertip and the center of the palm (2); and finally $dist3$ represents the distance between the two fingertips (3).

$$dist1 = \sqrt{(x_{palm} - x_{index})^2 + (y_{palm} - y_{index})^2} \quad (1)$$

$$dist2 = \sqrt{(x_{palm} - x_{middle})^2 + (y_{palm} - y_{middle})^2} \quad (2)$$

$$dist3 = \sqrt{(x_{index} - x_{middle})^2 + (y_{index} - y_{middle})^2} \quad (3)$$

where (x_{palm}, y_{palm}) are the coordinates of the central point of the palm region and (x_{index}, y_{index}) are the coordinates of the central point of the index fingertip, while (x_{middle}, y_{middle}) are the coordinates of the central point of the middle fingertip.

The computed angles are: $alpha$, which is the angle between the two fingers (4); $beta$, the index angle with Ox axis (5) and $gamma$, the middle finger angle with Ox axis (6). See Figure 3 for details.

$$alpha = beta - gamma \quad (4)$$

$$beta = \arctan \frac{y_{palm} - y_{index}}{x_{palm} - x_{index}} \quad (5)$$

$$gamma = \arctan \frac{y_{palm} - y_{middle}}{x_{palm} - x_{middle}} \quad (6)$$

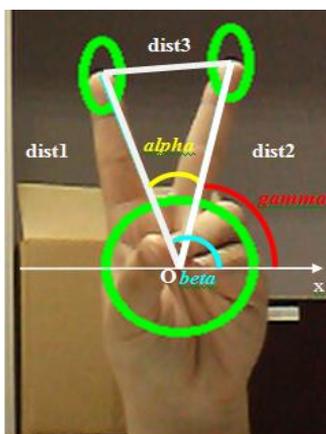


Figure 3

The computed distances and angles

Using these distances and angles the following gestures are defined: Select, clickLeft, clickRight, MoveLeft, MoveRight and DoubleClick. Each gesture is characterized by a set of parameters. These parameters are presented in Table I. It is important to observe that the distances $dist1$, $dist2$, $dist3$ are normalized with the finger thickness, so they are not influenced by the distance between the user and the camera.

Table 1

Parameters of the defined gestures

| Gesture | The parameters describing the gesture | The values for the gestures' parameters |
|-------------|---------------------------------------|---|
| Select | $alpha, dist3$ | $alpha < a, dist3 \leq \sigma$ |
| clickLeft | $alpha, dist1, dist2$ | $(dist2 < meanD2) \ \&\& \ ((meanAlpha - \tau) < alpha < (meanAlpha + \tau)) \ \&\& \ (dist1 \geq meanD1)$ |
| clickRight | $alpha, dist1, dist2$ | $(dist1 < meanD1) \ \&\& \ ((meanAlpha - \tau) < alpha < (meanAlpha + \tau)) \ \&\& \ (dist2 \geq meanD2)$ |
| MoveLeft | $alpha, beta, gamma$ | $(beta < (meanBeta - \tau)) \ \&\& \ (gamma < (meanGamma - \tau)) \ \&\& \ ((meanAlpha - \tau) < alpha < (meanAlpha + \tau))$ |
| MoveRight | $alpha, beta, gamma$ | $(beta > (meanBeta + \tau)) \ \&\& \ (gamma > (meanGamma + \tau)) \ \&\& \ ((meanAlpha - \tau) < alpha < (meanAlpha + \tau))$ |
| DoubleClick | $dist1, dist2$ | $(dist1 < meanD1) \ \&\& \ (dist2 < meanD2)$ |

In order to evaluate the temporal evolution of these distances, the mean distances $meanDi$, $i = 1..3$, are computed and later compared. This is done using only the frames associated with the performed gesture. In each new frame we compute the

$meanDi$ as the sum between $disti$, $i = 1..3$, computed in all previous frames and the current frame, divided by the number of the current frame. The temporal evolution of angles $alpha$, $beta$, $gamma$ is evaluated computing the mean values: $meanAlpha$, $meanBeta$, $meanGamma$ similar to $meanDi$. The end of gesture occurs when the fingertips are no longer tracked. The value of angle $alpha$ has a maximum range of 60° . The value for parameters a , σ and τ from Table 1 are experimentally determined.

The six gestures can be seen in Figure 4 a to f.

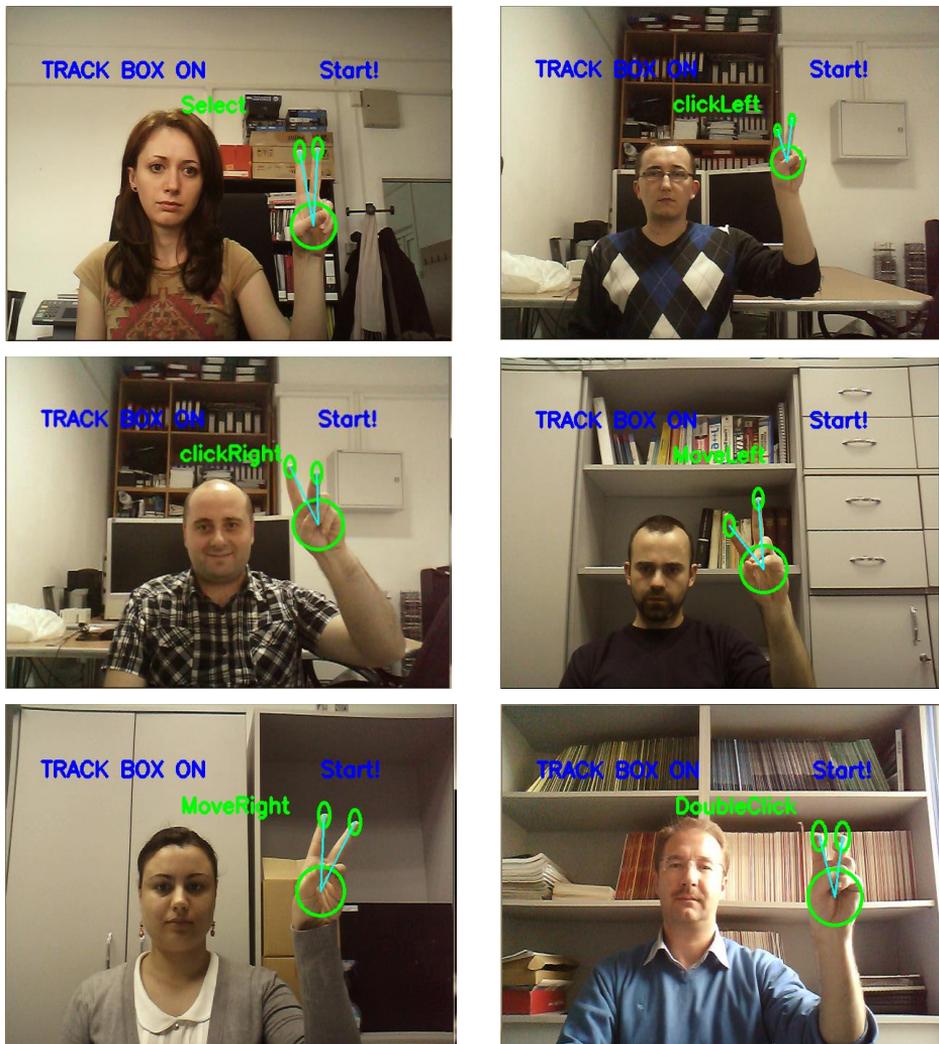


Figure 4
The defined gestures

5 Experimental Results

In order to test our method, ten subjects, both male and female, were asked to perform the six gestures. The subjects were quickly trained to perform the gestures. They had no difficulties in learning how to gesticulate correctly.

To test the robustness of our framework various environments were chosen. More exactly, the experiments were conducted in rooms with different illumination settings and complex backgrounds. In order to increase the difficulty of the recognition task, we deliberately chose to have skin-like objects in the background scene. Also the hand overlaps those objects. Considering the typical operating distance for an info-point application, the subjects were placed at a distance of 40 cm up to 120 cm from the camera. In all our experiments the value of parameter a was 25, the value of parameter σ was 2 and the value of parameter τ was equal to 10 degrees.

The proposed algorithm was implemented using OpenCV 2.4 library and compiled using Visual Studio 2010. The proposed solution ran smoothly using Windows 7 and a Dell Vostro 3500 notebook hardware platform based on Intel® Core™ i3 CPU, M370 @ 2.40 Ghz, 3.00 GB RAM. The images were acquired with the 640 x 480 VGA integrated webcam of the notebook.

The recognition rate for each gesture was experimentally determined (see Table 2) taking into consideration a total number of 50 image sequences (10 Individuals x 5 backgrounds) for each of the 6 gestures. The total recognition rate was about 95.66%, comparable with other previously reported results [12], [31].

Table 2
Average rate of our hand posture detection

| Gesture | Recognition rate |
|----------------|-------------------------|
| Select | 100% |
| clickLeft | 100% |
| clickRight | 100% |
| MoveLeft | 92% |
| MoveRight | 96% |
| DoubleClick | 86% |

The confusion matrix is shown in Table 3.

Table 3
The confusion matrix for the six classes

| | Select | clickLeft | clickRight | MoveLeft | MoveRight | Double Click |
|------------|--------|-----------|------------|----------|-----------|--------------|
| Select | 100 | 0 | 0 | 0 | 0 | 0 |
| clickLeft | 0 | 100 | 0 | 0 | 0 | 0 |
| clickRight | 0 | 0 | 100 | 0 | 0 | 0 |
| MoveLeft | 0 | 8 | 0 | 92 | 0 | 0 |
| MoveRight | 0 | 0 | 4 | 96 | 0 | 0 |
| DoubleClik | 8 | 4 | 2 | 0 | 0 | 86 |

The tests results performed in challenging illumination conditions are shown in Figure 5.

Another set of tests were conducted in order to check the robustness to occlusions and pose/perspective/distance to the camera variations. The results can be seen in Figures 6 and 7.

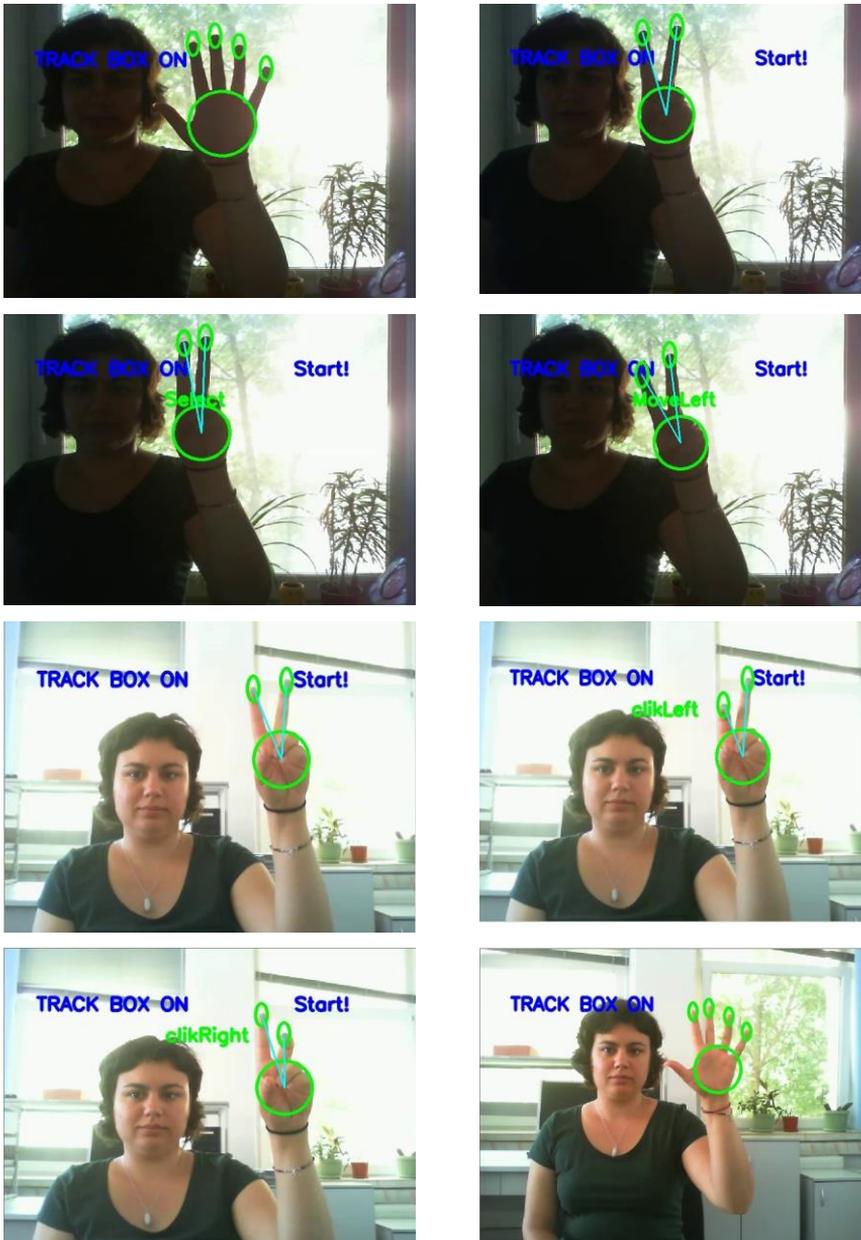


Figure 5
Experiments with different illumination conditions

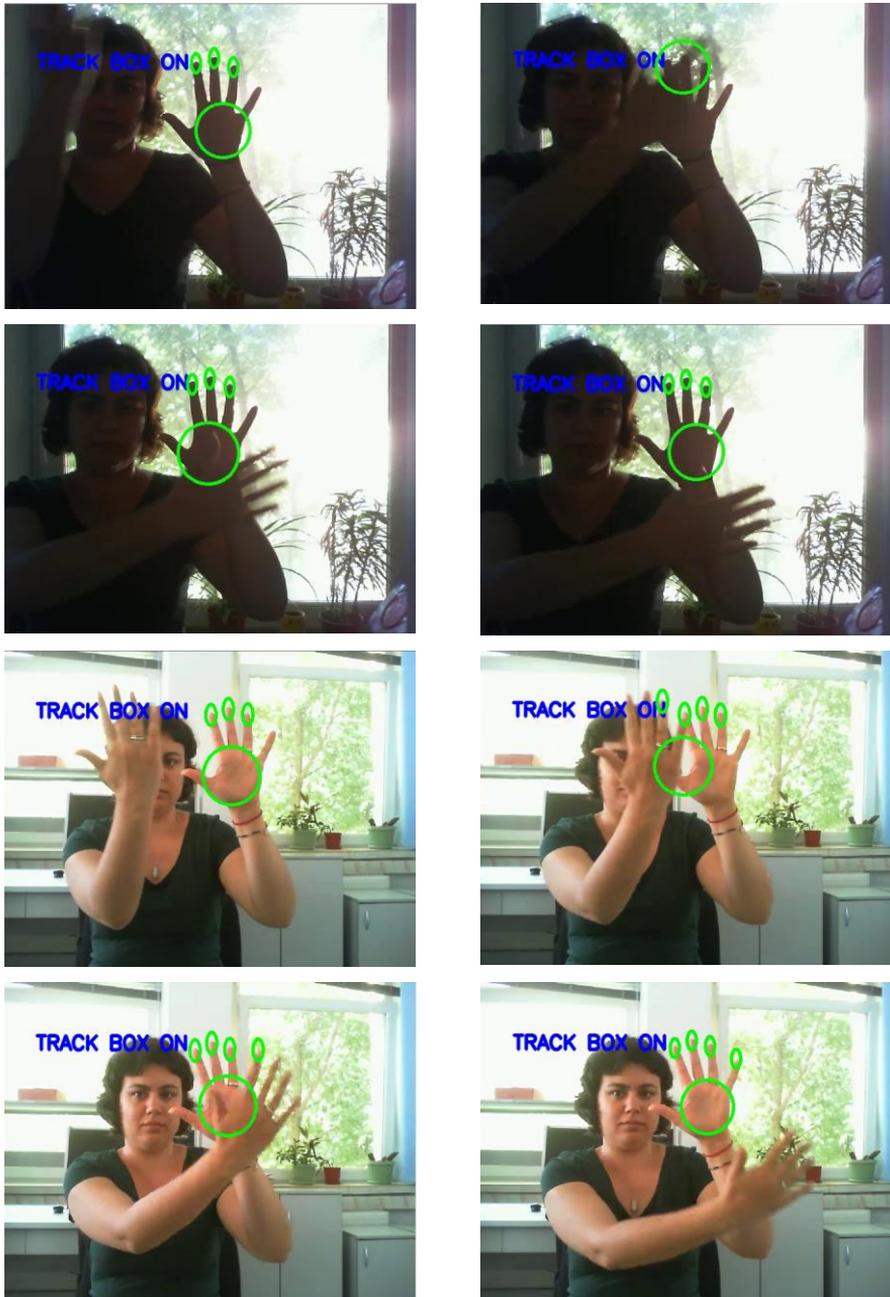


Figure 6

Experiments with occlusions in different illumination conditions

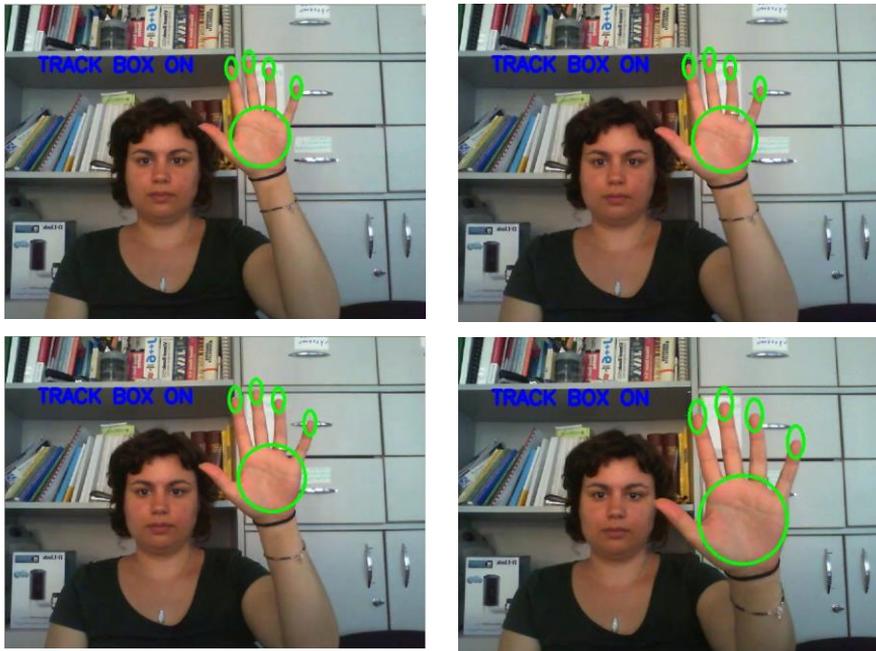


Figure 7

Experiments related to scale adaptation

Conclusion

This paper describes a novel vision-based real-time approach for recognizing basic two fingers gestures. Our method does not imply the acquisition of additional (usually more expensive) hardware, e.g., Leap Motion [38] or Kinect controller [39] nor wearable devices like in the case of Myo Armband [40] but employs a common 2D video camera integrated in most laptops or an inexpensive USB connected external web cam. Finger features are first extracted by looking for similarities in length and thickness. For this step, the MS algorithm with a multi-scale filtering capability is used. Then, the fingertips are also tracked by using the palm region of the hand. This enables us to define a supplementary point that is used to design the final set of natural gestures to be recognized. The main characteristics of the proposed framework for fingertips detection and tracking are robustness and high accuracy with reduced computational costs. For the classification step, the scale-invariant distances and angles are calculated. By analyzing the position of just two fingers, various interface actions, such as clicking and moving, are recognized. The system performances were analyzed further with respect to the real-time capability and the recognition rate. Experimentally, we have proven that the proposed approach provides a good classification rate (95.66% for a total of 300 test images) along with tight timing constraints (an average response time of 30 ms has been determined), which

enables humans to perceive interactivity. Thus the proposed architecture is suitable to support different types of applications, e.g. touchless info-point machine or smart TVs. As a future development, we estimate that the current framework could be easily extended to work within a 3D vision system.

References

- [1] G. Simion, C. Căleanu, “A ToF 3D Database for Hand Gesture Recognition”, 10th International Symposium on Electronics and Telecommunications (ISETC), Timisoara, Romania, 2012, pp. 363-366
- [2] F. Malassiotis, N. Tsalakanidou, V. Mavridis, N. Giagourta, N. Grammalidis, M.G Strintzis, “A Face and Gesture Recognition System Based on an Active Stereo Sensor,” In: Proceedings 2001 ICIP, Thessaloniki, Greece, 7-10 Oct. 2001, Vol. 3, pp. 955-958
- [3] M. Bray, Koller-Meier, L. V Gool, “Smart Particle Filtering for 3D Hand Tracking”, in Sixth IEEE International Conference on Automatic Face and Gesture Recognition, Los Alamitos, CA, USA, 2004, pp. 675-680
- [4] P. Breuer, C. Eckes, S Muller, “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, 2007, pp. 247-260
- [5] 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), pp. 334-343, 2008
- [6] 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, pp. 1-18
- [7] K. Kim, T. H. Chalidabhongse, D. Harwood, L. Davis, “Real-Time Foreground-Background Segmentation Using Codebook Model,” in *Real-time Imaging*, Vol. 11, No. 3, pp. 167-256, 2005
- [8] M. de La Gorce, D. J. Fleet, N. Paragios, “Model-based 3D Hand Pose Estimation from Monocular Video” *IEEE Transactions on Pattern Analysis And Machine Intelligence*, Vol. 33, No. 9, pp 1793-1805, 2011
- [9] Y. Wu, Y. L. Lin, T. S. Huang, “Capturing Natural Hand Articulation,” in: Eighth IEEE International Conference on Computer Vision, 2001, pp. 426-432
- [10] B. Stenger, P. R. S. Mendonca, R. Cipolla, “Model-based 3D Tracking of an Articulated Hand” *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 310-315, 2001
- [11] J. M. Kim, W. K. Lee, “Hand Shape Recognition Using Fingertips,” 5th Int. Conf. Fuzzy Syst. Knowledge Discovery, 2008, pp. 44-48

- [12] D. Lee, Y. Park, "Vision-based Remote Control System by Motion Detection and Open Finger Counting," *IEEE Trans. Consumer Electron.*, Vol. 55, No. 4, pp. 2308-2313, 2009
- [13] L. Bretzner, I. Laptev, T. Lindeberg, "Hand Gesture Recognition Using Multiscale Color Features, Hierarchical Models and Particle Filtering," in *Proceedings of Int. Conf. on Automatic face and Gesture recognition*, Washington D.C., 2002, pp. 423-428
- [14] M. Donoser, H. Bischof, "Real Time Appearance-based Hand Tracking," *Pattern Recognition*, 2008, *ICPR 2008*, 19th International Conference on, Vol., No., pp. 1-4, 2008
- [15] R. Belaroussi, M. Milgram, "A Real Time Fingers Detection by Symmetry Transform Using a Two Cameras System," *LNCS*, Vol. 5359, 2008, pp. 703-712
- [16] O. Gallo, S. M. Arteaga, J. E. Davis, "A Camera-based Pointing Interface for Mobile Devices," *IEEE Int. Conf. Image Processing*, 2008, pp. 1420-1423
- [17] L. Song, M. Takatsuka, "Real-Time 3D Finger Pointing for an Augmented Desk," In *Australasian conference on User interface*, Vol. 40, Newcastle, Australia, 2005, pp. 99-108
- [18] C. von Hardenberg, F. Bérard, "Bare-Hand Human-Computer Interaction", *Proceedings of the ACM workshop on Perceptive User Interfaces*, Orlando, Florida, USA, 2001, pp. 1-8
- [19] K. Oka, Y. Sato, H. Koike, "Real-Time Fingertip Tracking and Gesture Recognition", *IEEE Computer Graphics and Applications*, pp. 64-71, 2002
- [20] S. Malik, C. McDonald, G. Roth, "Finger Detection via Blob Finding Flood fill: Hand Tracking for Interactive Pattern-based Augmented Reality" *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR'02)*, 2002
- [21] P. Premaratne, Q. Nguyen, "Consumer Electronics Control System Based on Hand Gesture Moment Invariants," *Computer Vision, IET*, Vol. 1, No. 1, pp. 35-41, 2007
- [22] L. Yun Liu, G. Zhijie, Yu Sun, "Static Hand Gesture Recognition and its Application based on Support Vector Machines," *Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, SNPD '08. Ninth ACIS International Conference on*, 2008, pp. 517-521
- [23] L. Taehee, T. Hollerer, "Handy AR: Markerless Inspection of Augmented Reality Objects Using Fingertip Tracking," *Wearable Computers*, 2007 11th *IEEE International Symposium on*, 2007, pp. 83-90
- [24] J. MacLean, R. Herpers, C. Pantofaru, L. Wood, K. Derpanis, D. Topalovic, J. Tsotsos, "Fast Hand Gesture Recognition for Real-Time Teleconferencing Applications," *Int. Workshop Recognition, Analysis Tracking of Faces Gestures Real Time Syst.*, 2001, pp. 133-140

- [25] B. Stenger, A. Thayananthan, P. Torr, R. Cipolla, "Modelbased Hand Tracking Using a Hierarchical Bayesian Filter," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 28, No. 9, pp. 1372-1384, 2006
- [26] J. Alon, V. Athitsos, Y. Quan, S. Sclaroff, "A Unified Framework for Gesture Recognition and Spatiotemporal Gesture Segmentation," *IEEE Trans. Patt. Anal.Mach. Intell.*, Vol. 31, No. 9, pp. 1685-1699, 2009
- [27] H. Ying, J. Song, X. Ren, W. Wang, "Fingertip Detection and Tracking Using 2D and 3D Information," *Proc. 7th World Congress Intell. Control Autom.*, 2008, pp. 1149-1152
- [28] S. C. Crampton, M. Betke, "Counting Fingers in Real Time: A Webcam-Based Human-Computer Interface with Game Applications," *Proc. Conf. Universal Access Human-Comput. Interaction*, 2003, pp. 1357-1361
- [29] A. M. Burns, B. Mazzarino, "Finger Tracking Methods Using EyesWeb," *Proceedings of the 6th international conference on Gesture in Human-Computer Interaction and Simulation*, Vol. 3881, 2006, pp. 156-167
- [30] L. Gui, J. P. Thiran, N. Paragios, "Joint Object Segmentation and Behavior Classification in Image Sequences", *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Minneapolis 2007*, pp. 1-8
- [31] D. Lee, S. Lee, "Vision-based Finger Action Recognition by Angle Detection and Contour Analysis" *ETRI Journal*, Vol. 33, No. 3, pp. 415-422, 2011
- [32] C. David, V. Gui, P. Nisula, V. Korhonen, "Dynamic Hand Gesture Recognition for Human-Computer Interactions," *Proc. of 6th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI)*, 2011, pp. 165-170
- [33] J. Davis, M. Shah, "Recognizing Hand Gestures", *European Conference on Computer Vision*, 1994, pp. 331-340
- [34] V. Gui, G. Popa, P. Nisula, V. Korhonen, "Finger Detection in Video Sequences Using a New Sparse Representation," *Acta Technica Napocensis Electronics and Telecommunications*, Vol. 52, No. 1, 2011
- [35] V. Gui, "Edge Preserving Smoothing by Multiscale Mode Filtering". *European Conference on Signal Processing, EUSIPCO'08, Lausanne*, pp. 25-29, 2008
- [36] S. Gu, Y. Zheng, C. Tomasi, "Efficient Visual Object Tracking with Online Nearest Neighbor Classifier". *The 10th Asian Conference on Computer Vision, ACCV Queenstown, New Zealand*, 2010
- [37] J. Santner, C. Leistner, A. Sarti, T. Pock, H. Bischof, "PROST Parallel Robust Online Simple Tracking", *IEEE CVPR 2010*, pp. 723-730
- [38] <https://www.leapmotion.com/>
- [39] www.xbox.com
- [40] <https://www.myo.com/>