Natural Human-Computer Interaction Using Static and Dynamic Hand Gestures

Guillaume Plouffe[1], Ana-Maria Cretu[2], Pierre Payeur[3]

School of Electrical Engineering and Computer Science
University of Ottawa
Ottawa, Canada
[1]gplou010@uottawa.ca, [3]ppayeur@uottawa.ca

Department of Computer Science and Engineering
Université du Québec en Outaouais
Gatineau, Canada
[2]ana-maria.cretu@uqo.ca

***Abstract*—The paper presents design and implementation issues of a natural gesture user interface for real-time hand gesture recognition in depth data collected by a Kinect sensor. The hands are first segmented based on depth and a novel algorithm improves the scanning time in order to identify the first pixel on the hand contour within this space. Starting from this pixel, a directional search algorithm allows for the identification of the entire hand contour and the *k*-curvature algorithm is then employed to locate the fingertips over it. Dynamic time warping (DTW) guides the selection of gesture candidates and also recognizes gestures by comparing an observed gesture with a series of pre-recorded reference gestures. The comparison of results with state-of-the-art approaches shows that the proposed system is similar in performance or outperforms the existing solutions for the static and dynamic recognition of signs.**

Keywords—Kinect; human-computer interaction; natural interface; dynamic time warping

I. INTRODUCTION

Natural user interfaces are subject to various requirements: they have to work in real-time, to support a natural interaction with the user (by remaining invisible and being adaptable without imposing elaborate calibration procedures and without the need of specialized and costly control equipment), to offer accurate interpretation of gestures, and to provide robustness against background clutter while allowing the user to unobtrusively interact with the application. The inherent complexity of these requirements still provides significant challenges to the research community and justifies the interest in this topic, particularly in the real-time recognition of hand gestures.

In terms of data acquisition for hand gesture recognition, possible 3D motion capture solutions at the level of the fingers include accelerometers, magnetic trackers, optical marker systems, and data gloves. These are generally expensive, require extensive calibration to the user and limit in general the natural movement. Kinect, a low-cost, off-the-shelf sensor offers a vision-based markerless motion capture solution that gathers positional information about a user motion. While a good survey for gesture recognition in general is available in [1] and one focused on depth images in [2], the following literature review summarizes the solutions that make use of RGB-D data, such as returned by the Kinect in the context of hand gesture recognition.

The approaches used for hand localization vary according to the nature of data. In color images, hand segmentation can be achieved by detecting skin color [3, 4]. In depth data, solutions are based on depth thresholding. Empirical depth thresholding solutions choose the most probable search space by trial and error and search the hand within it. Automated solutions for hand location identification use the assumption that the hand is the closest object in the scene [5, 6], or employ other reference elements in the scene, such as the head position [7]. Certain existing solutions impose the intervention of the user to calibrate the system prior to its use, for example by a wave gesture executed by the user [8]. Others impose constraints, such as the use of a colored glove [9] or a black belt on the wrist of the gesturing hand [6] in order to simplify the hand localization procedure and its segmentation. These solutions impede on the invisibility of the interface to the user and constrain her/him to adapt the behavior to the interface.

After the hand localization, appropriate features have to be selected for the recognition of gestures. These include for depth data: hand contour, hand palm center [5, 10], finger properties (e.g. count, direction) [11], hand trajectory [12], and gradient kernel descriptors [13]; while for color data, Haarlet [8], Gabor [14], and SIFT [13] descriptors, and shape distance metrics [6, 15] are used. Among the approaches proposed in the literature for the recognition of gestures from the selected features are: neural networks [8, 12], random forests [14], and hidden Markov models. Most of the existing solutions for hand gesture recognition are designed for use either on static or on dynamic gestures, for one or two hands; only very few solutions work simultaneously on both static and dynamic gestures [7, 16].

The objective of this work is to develop an improved, low complexity and real-time solution for the recognition of both static and dynamic gestures executed by one or multiple hands from depth data returned by a Kinect sensor. The interface is invisible to the user in the sense that it does not require any constraints or calibration procedure. Due to the fact that our system is based only on depth information, cluttered

978-1-4673-9175-7/15/$31.00 ©2015 IEEE

backgrounds, lighting conditions, clothing or skin color have little impact on the reported performance.

II. Proposed Solution

Fig.1 summarizes the framework of the proposed solution for static and dynamic hand gesture recognition.

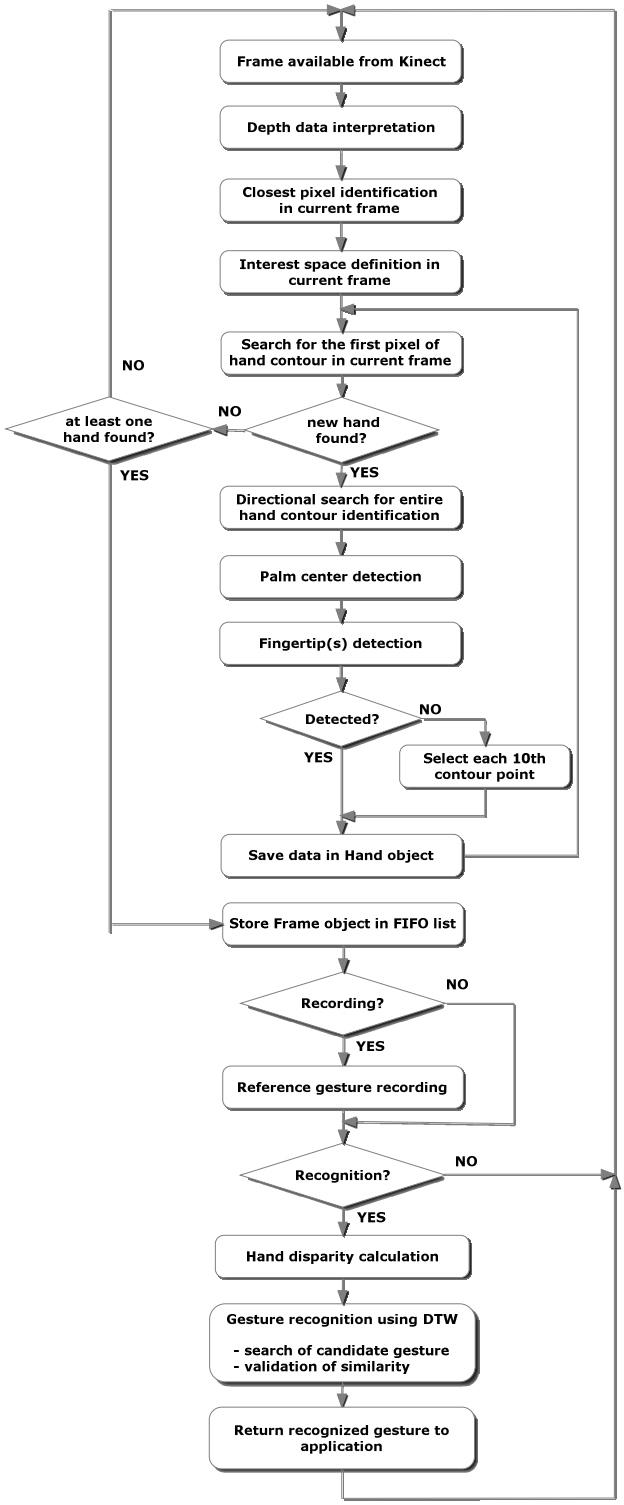


Fig. 1. Framework for static and dynamic hand gesture recognition.

Depth Data Interpretation - The Windows SDK 1.8 for Kinect returns 16-bit raw data, from which the last 13 bits correspond to the depth information. In order to only retain this information, a 3-bit shift operation is performed on the collected data. This depth information is recuperated each time a new frame becomes available, and is used in the context of this work to detect one or multiple hand contours.

Closest Pixel Identification and Interest Space Definition – A simple solution is used in order to guide the search for the hand contour. In particular, the minimal distance (closest) pixel to the Kinect is identified, based on the assumption that the hand is the closest object to the camera in the scene. As we naturally tend to keep the hands in front of our body when we gesticulate, the approach does not impose significant limitations for the user. To define the interest space in which the search for hands will occur, the depth of the identified pixel is considered to represent the minimal depth (lower bound of the interest space) and a constant (e.g. an empirically determined value of 20cm in this work) is added in order to calculate the maximum depth (upper bound of the interest space).

Search of the First Pixel of a Hand Contour- A search is then performed within the interest space to identify the first pixel that has at least one neighboring pixel that is not included in the interest space. To optimize the search algorithm proposed in [10], in this work, the search takes place in blocks of 20×20 pixels instead of blocks of 5×5 pixels and doesn't retrace each contour list of point or bounding box for every pixel inside each contour to distinguish the first point from the others. As illustrated in Fig. 2, a pixel is considered valid if it is present in the defined interest space. If a pixel is valid and if it doesn't possess any neighboring pixel that is valid, the search backtracks pixel by pixel towards an invalid neighboring pixel until a contour pixel is found. We verify if this pixel is part of an already found contour and, if not, a new potential hand is found. If at least one neighboring pixel is valid, then we are likely inside an already found hand and continue the search. Unlike the solution in [10], this approach improves the search time and it does not restrict the search to the central part of the image.

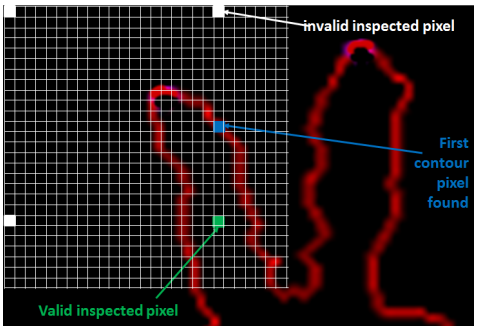
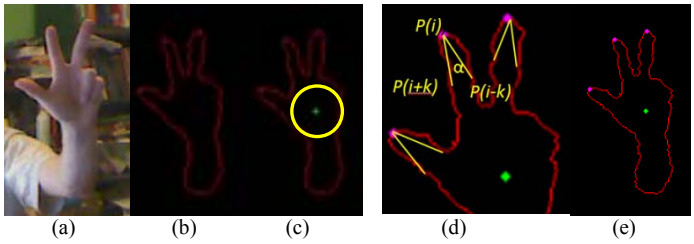


Fig. 2. Detection of the first pixel of a hand contour.

Directional Search for Entire Hand Contour Identification – Starting from the identified initial contour pixel, a directional search is performed to detect the entire contour of the hand.

The considered search directions are up-left, up-right, down-left and down-right and a direction is preferred if it has been used for the previous pixel as well. The algorithm includes also a solution to backtrack if an unknown valid configuration is encountered, as in [10]. An example of detected contour for the 3 digit sign is shown in Fig. 3b.



(a) (b) (c) (d) (e)

Fig. 3. (a) Gesture, (b) detected contour, (c) largest circle contained in the palm and identified palm center, in green, (d) angle calculation for fingertip identification and (e) identified fingertips marked by magenta dots.

Palm Center Detection – Once a hand contour is identified, the hand palm center is computed using the contour pixels as well as the interior pixels of the hand, using an approach similar to the one proposed by Cerezo [5]. The bounding box of the hand is first calculated using the minimum and maximum points on the X and Y axis respectively. The minimal distance of interior pixels, d_{min}, is then computed for each block of 5×5 pixels with respect to each contour pixel. One contour pixel in each 5 proved experimentally to be sufficient for this purpose. The interior pixel with the maximum value for d_{min} is considered to be the palm center. In simple terms, the center of the palm is the center of the largest circle that can be circumscribed in the palm (Fig. 3c).

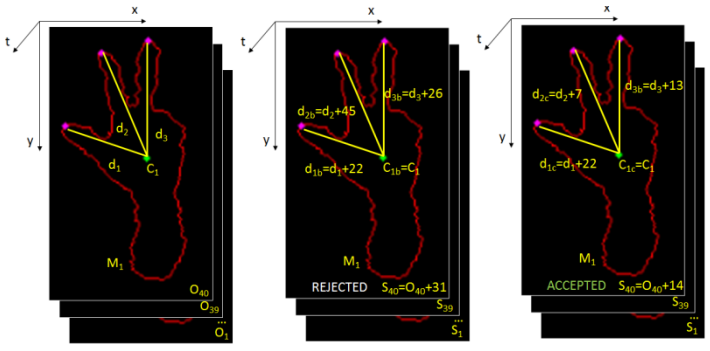
Fingertips Detection – Also once a hand contour is identified, the fingertips are detected as points over contour that respect certain properties. The k-curvature algorithm is applied for this purpose, because it was shown to provide the best results in terms of overall success rate, to support the highest range of hand rotations within which is capable to perform reliably and to have the lowest complexity in terms of specific steps when compared with other solutions in case of static hand recognition [17]. As illustrated in Fig. 3d, for each contour point, the algorithm determines the angle between two equal length measuring tapes that start at the point and end 20 points away (e.g. the step size, k is 20 pixels) in both directions. If the angle α between them (e.g. between *P(i-k)->P(i) and P(i)->P(i+k)*) in Fig. 3d has an empirically-determined value between 25° and 55°, a fingertip is considered identified at that point. The three detected fingertips for the 3 digit sign are shown in magenta in Fig. 3e. The contour, palm center and fingertip informations are stored in a new Hand object. This object is then added to the list of hands inside the Frame object (e.g. the block "Store frame object in FIFO list" in Fig. 1). If found, the next first pixel of a hand contour will be, in turn, analyzed to build an object Hand and further added to the list of hands inside the Frame object.

Hand Disparity Calculation - To allow for the recognition of gestures independent of their position on the screen, we use the position of fingertips with respect to the center of the palm(s) and the relative position of the palm(s) center with respect to the palm center of the first discovered contour, as illustrated in Fig. 4a. To decide if a stored gesture should be accepted in the list of candidate gestures, we calculate the disparity between the 40th stored frame (Fig. 4b and 4c) and the 40th frame of the observed gesture (Fig. 4a), using the following calculation:

$$\Delta SO_{40} = \frac{\sum_{j=1}^{M-1} \frac{\sum_{i=1}^{N-1} \Delta d_i}{N} \times w_f + \Delta C_j \times w_p}{M} \quad (1)$$

$$= \frac{\left(\frac{(22+45+26)}{3} \times 1\right) + 0 \times 7}{1} = 31$$

where M is the number of hands detected in the image, N the number of fingers detected in the image, Δd_i is the difference in the distance from the palm center to detected fingertip i in the observed and the stored gesture frame, ΔC_j is the difference between the palm centers in the stored and observed gesture frames and w_f (e.g. 1 in current work) and w_p (e.g. 7 in current work) are the weights for the finger and palm center disparities, respectively. The resulting value is then compared to a threshold value (e.g. 30 in the current work) to decide if the stored gesture should be accepted as a candidate gesture or not. In the case of Fig. 4b it is rejected because we use a threshold of 30, while in the case of Fig. 4c is accepted. For frames with only one hand, the palm center disparity is always 0 since the position is relative to itself. For a limited number of gestures in which the fingertips are not visible or not detected, the contour points (each 10th point) are used as a basis to calculate the distance. Since this distance gets higher for gestures involving multiple hands and fingers, the calculation in eq. (1) includes a normalization operation by the number of fingers detected in an image, N, in order not to bias the recognition. For the same reason, the disparity value for an entire image is obtained by dividing the total hand disparity over all detected hands by the number of hands, M.



(a) (b) (c)

Fig. 4. (a) Hand disparity calculation for observed gesture and selection of gesture candidates: example of (b) rejected and (c) accepted gesture candidate.

Store in FIFO List – Through the implemented interface, the user is allowed to record and store a sequence of reference gestures specific to his/her application. The system enters in a

Recording state once the user clicks on the "Record Gesture" button. As previously described, this series of gestures is saved in a FIFO list. In order to ensure real-time behavior of the system, the list is limited to 40 images and contains the hand disparities values for each hand in each image. Once a sequence of new images representing a gesture being executed is made available by the Kinect, the Recognition mode can be activated by pressing the "Gesture Recognition" button to recognize the observed gesture. If the gesture is recognized, its name appears in the message zone of the interface.

Gesture Recognition Using Dynamic Time Warping – The gesture recognition module executes in the context of this work in two steps, both of which are capitalizing on the dynamic time warping algorithm [18]: (1) the search of candidate gestures based on the similarity between the observed gesture and each reference gesture and (2) the validation of the similarity between each observed version and each identified candidate reference gesture.

The solution proposed in [10] compares in step (1), the last image of an observed gesture with the last image of each reference gesture in the list of reference gestures by calculating the Euclidean distance. If the reference gesture associated with the lowest distance is within a threshold, the gesture is considered to be a possible candidate. This solution is simple and fast, but fails to distinguish between two dynamic gestures whose last image is identical, because it only compares the latter in order to recognize a gesture. This is the case for example of the dynamic letters of the ASL alphabet, *j* and *z* (see Fig. 7). To address this issue, in this work DTW is employed not only for validation, in step (2), but also as a mean to identify possible candidates in step (1). DTW computes the disparity between two data series acquired at different times. This is achieved by constructing a matrix that contains Euclidean distances at aligned points over the two sequences. This matrix is then used to calculate the minimal cost (or the shortest path) between the two series. The choice of direction can be horizontal, vertical and diagonal directions. A weight is associated with each of these directions and the shortest path has to be inferior to a threshold in order for the two sequences to be considered similar. In the context of this work, the two series represent an observed and a reference gesture, respectively, as shown in Fig. 5.

In this figure, values in circles inscribed in each cell represent the hand disparity for each pair of observed and stored gestures before applying the DTW direction weight (w_h, w_v, w_d). The product of these weights with the circled values gives the value used for minimal cost computation, indicated in color on the bottom right of each cell.

The empirically determined value for the maximum Euclidean distance, in pixels, that is accepted for a candidate gesture is 30. This value ensures a good compromise between the processing time for the 40 images that are compared for each candidate gesture and the number of candidate gestures to be processed. The maximum cost accepted for a path in the DTW matrix per image is set empirically to 30. The weights associated with the directions of movement are experimentally set to 0.005 for vertical, w_v, and horizontal direction, w_h, and 0.003 to the diagonal direction, w_d, in order to favor the latter. When we perform a search in the DTW matrix for the shortest path, in the virtual space of frames (observed vs. reference gestures), as illustrated in Fig. 5, the neighboring values of static gestures are in general equal (Fig. 5a). Thus any direction in the DTW matrix could be chosen and the associated gesture could be falsely rejected based on the path taken. To avoid this problem, the search is biased (diagonal weight w_d is lower than vertical, w_v, and horizontal, w_h, weights) to favor the next image of both reference and last observed gesture performed, corresponding to a diagonal movement in the DTW matrix.

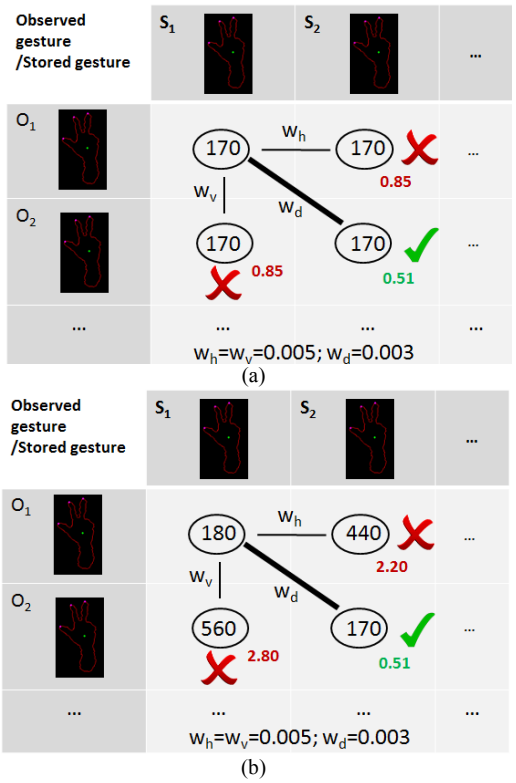


Fig. 5. Example of DTW matrix for (a) static and (b) dynamic gestures.

To improve the performance, a maximum number of 10 horizontal and vertical movements are allowed. This results in a faster elimination of divergent candidate gestures. In order to select the winning gesture among the gesture candidates, the one with the smallest disparity value (and that respects the threshold) is chosen.

III. Experimental Results

In order to test the proposed approach we have used the set of static sign digits from 1 to 9, the ASL alphabet, from which the static signs are illustrated in Fig. 6 and the dynamic ones in Fig. 7, and a series of static two hand gestures proposed for controlling a game in [4], illustrated in Fig. 8. The figures

show the detection of the contour, palm and fingertips for the static gestures and also the trajectory in case of dynamic gestures, as shown in Fig. 7. The tests do not take place in controlled conditions. As one can notice in the figures, the background is heavily cluttered.

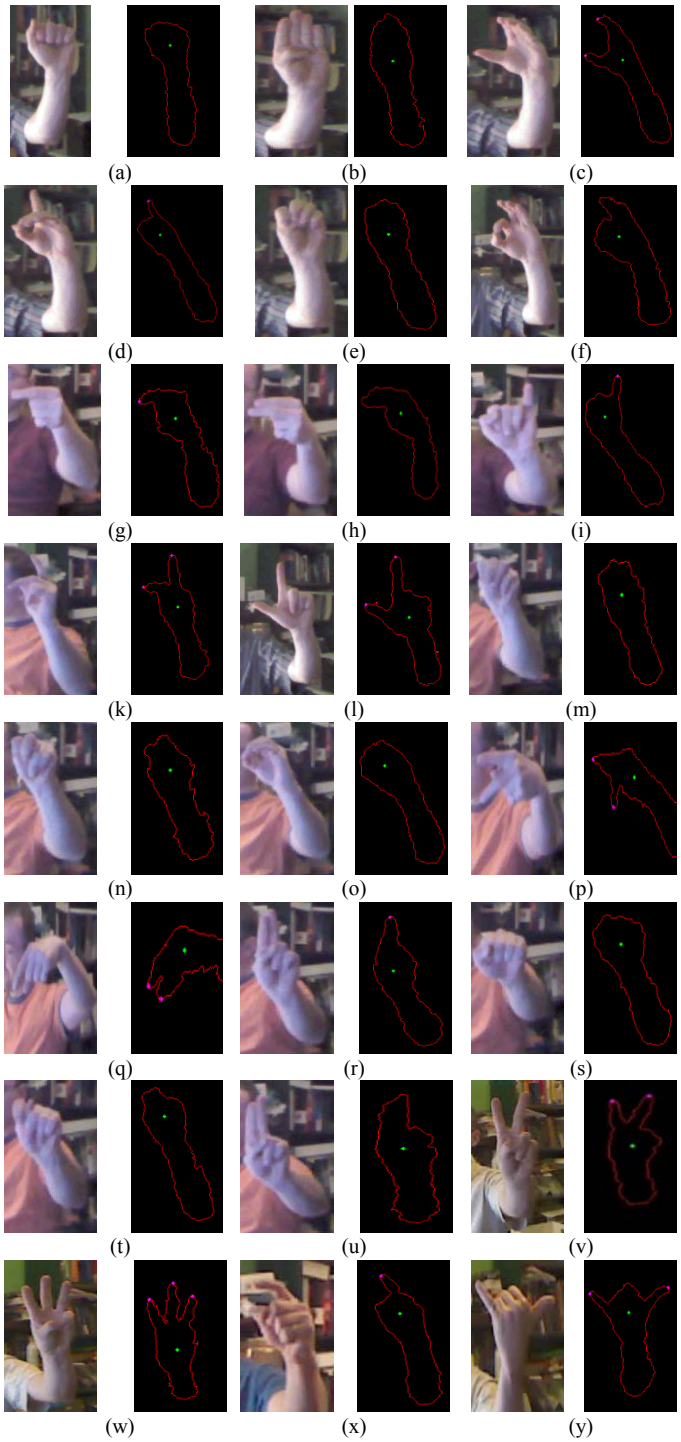


Fig.6. Static American sign alphabet gestures (a)-(i) and (k)-(y).

It can be observed that, in general, the proposed solution is able to correctly locate the points of interest over the hand surface as well as its contour. Fingertips can be missing in particular for gestures involving heavily bending of fingers or fingers that touch each other. In these cases, the *k*-curvature algorithm fails to separate them; this situation is compensated with the use in this case of the contour information.

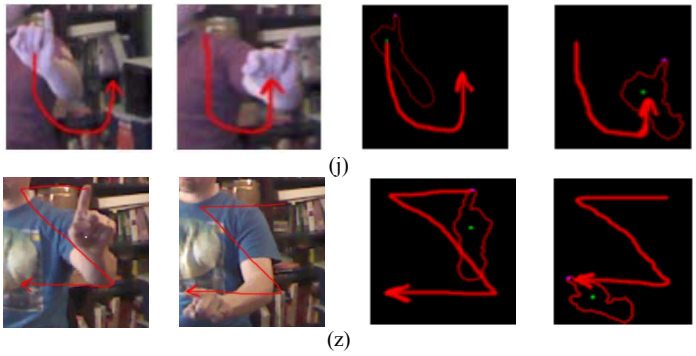


Fig.7. Dynamic American sign alphabet gestures (j) and (z).

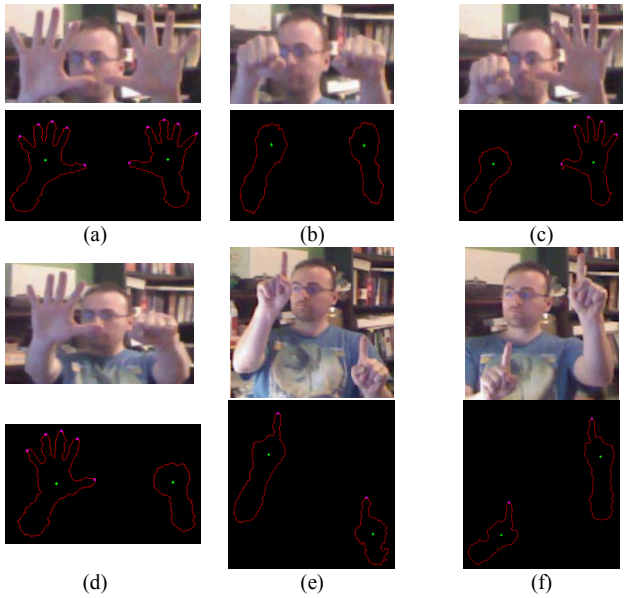


Fig. 8. Two hand gestures to control a game: (a) both palms, (b) both fists, (c) left fist, right palm, (d) left palm, right fist, (e) left hand above right hand, and (f) right hand above left hand.

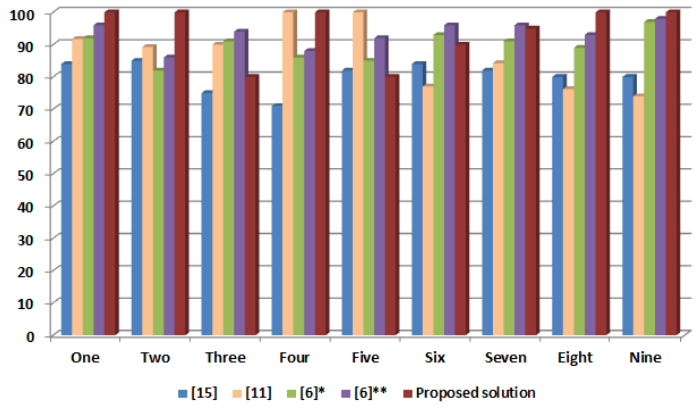


Fig. 9. Average recognition rates for digit signs (*threshold decomposition solution, **near-convex decomposition solution).

For the digit sign gestures, as it can be observed in Fig. 9 and for the ASL alphabet, in Table I, the proposed solution offers a performance similar to the existing literature, with an average recognition rate of 93.9% for the digit signs (best reported average rate in the literature being 93.2%) and of 89.9% for the alphabet (best reported average rate in the literature being 90.8%).

TABLE I. AVERAGE RECOGNITION RATES FOR THE ASL ALPHABET

Gesture	[14]	[16][a]	[13][b]	[13][c]	[16][d]	Proposed solution
a	75	81	89	94	90	100
b	83	89	95	98	96	100
c	57	83	92	95	91	100
d	37	83	87	91	90	100
e	63	77	82	90	90	100
f	35	87	94	97	94	100
g	60	87	86	92	91	60
h	80	92	92	95	95	100
i	73	91	94	96	97	80
j	N/A	N/A	N/A	N/A	87	100
k	73	73	84	91	89	100
l	43	93	93	97	94	100
m	87	68	77	88	90	100
n	17	65	74	82	88	50
o	23	76	78	97	87	83
p	13	76	88	93	90	100
q	77	74	88	93	88	100
r	63	73	76	84	91	100
s	17	73	76	88	83	100
t	7	57	68	82	84	40
u	67	74	82	88	86	80
v	87	76	83	90	91	100
w	53	79	91	95	93	80
x	20	79	78	87	90	100
y	77	88	94	96	97	90
z	N/A	N/A	N/A	N/A	100	73
Average	**49.5**	**72.9**	**78.5**	**84.6**	**90.8**	**89.9**

[a] Random forest, [b] Depth only, [c] Depth and color, [d] SVM

Finally Fig. 10 compares the performance with the solution of Zhu and Yuan [4] for the set of two hand gestures for controlling a game reported in their work (Fig. 8). The performance of the proposed solution is clearly superior.

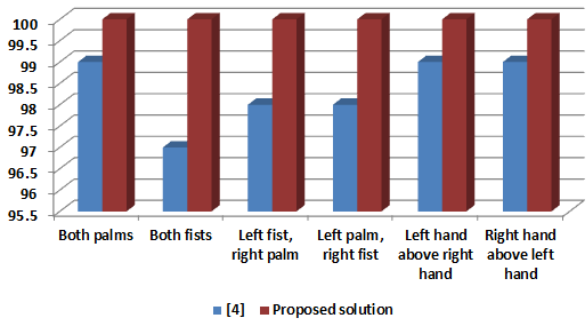


Fig. 10. Average recognition rates for two hand gestures to control a game

IV. CONCLUSION

The paper presented the development of a novel approach for gesture recognition using only depth images collected by a Kinect sensor. Starting from the closest pixel in the scene, the initial pixel of the hand contour is found using an improved block search scheme and a directional search is performed for the identification of the entire hand contour. The *k*-curvature algorithm is then employed to locate the fingertips over the contour and the dynamic time warping algorithm is used to select a candidate gesture as well as to recognize gestures by comparing them with a series of pre-recorded reference gestures. The system achieves an average performance of 94.6% for one or two hand static and dynamic gestures, with which it is able to deal simultaneously.

REFERENCES

[1] S. Mitra and T. Acharya, "Gesture Recognition: A survey," *IEEE Transactions on Systems, Man, and Cybernetics*, pp. 311–324, 2007.

[2] J. Suarez and R.R. Murphy, "Hand Gesture Recognition with Depth Images: A Review", *IEEE Int. Symp. Robot and Human Interactive Communication*, pp. 411-417, Paris, France, 2012.

[3] C.-C. Hsieh and D.-H. Liou, "Novel Haar Features for Real-Time Hand Gesture Recognition", *Real Time Image Processing*, vol. 10, pp. 357-370, 2015.

[4] Y. Zhu and B. Yuan, "Real-Time Hand Gesture Recognition with Kinect for Playing Racing Video Games", *Int. Joint Conf. Neural Networks*, pp. 3240-3246, Beijing, China, 2014.

[5] F. Trapero Cerezo, *3D Hand and Finger Recognition using Kinect,* Universidad de Granada, Spain, 2013.

[6] Z. Ren, J. Yuan, and Z. Zhang, "Robust Hand Gesture Recognition Based on Finger-Earth Mover's Distance with a Commodity Depth Camera", *Proc. ACM Int. Conf. Multimedia*, pp. 1093-1096, 2011.

[7] M. Correa, J. Ruiz-del-Solar, R. Verschae, J. Lee-Ferng, and N. Castillo, "Real time Hand Gesture Recognition using a Range Camera", J. Baltes et al. (Eds.), *RoboCup 2009*, LNAI5949, pp. 46-57, 2010.

[8] M. Van den Bergh, D. Carton, R. De Nijs, N. Mitsou, C. Landsiedel, K. Kuehnlenz, D. Wollherr, L. Van Gool, and M. Buss, "Real-time 3D Hand Gesture Interaction with a Robot for Understanding Directions from Humans", *IEEE Int. Symp. Robot and Human Interactive Communication*, pp. 357-362, Atlanta, USA, 2011.

[9] M. Schroder, C. Elbrechter, J. Maycock, R. Haschke, M. Botsch, and H. Ritter, "Real-Time Hand Tracking with a Color Glove for the Actuation of Anthropomorphic Robot Hands", *IEEE-RAS Int. Conf. Humanoid Robots*, pp. 262-269, Osaka, Japan, 2012.

[10] D.J. Ryan, *Finger and gesture recognition with Microsoft Kinect,* Master Thesis, University of Stavanger, Norway, 2012.

[11] Y. Li, "Multi-Scenario Gesture Recognition Using Kinect", *IEEE Int. Conf. Computer Games*, pp. 126-130, 2012.

[12] K. Rimkus, A. Bukis, A. Lipnickas, and S. Sinkevicius, "3D Human Hand Motion Recognition System", *IEEE Int. Conf. Human System Interaction*, pp. 180-183, Sopot, Poland, 2013.

[13] K. Otiniano-Rodriguez and G. Camara-Chavez, "Finger Spelling Recognition from RGB-D Information using Kernel Descriptor", *IEEE Conf. Graphics, Patterns and Images*, Arequipa, pp. 1-7, 2013.

[14] N. Pugeault and R. Bowden, "Spelling it out: Real-time ASL fingerspelling recognition". *Conf. Consumer Depth Cameras for Computer Vision*, Barcelona, Spain, 2011.

[15] Z. Ren, J. Meng, and J.Yuan, "Depth Camera Based Hand Gesture Recognition and its Applications in Human-Computer-Interaction", *IEEE Conf. Information, Comm. and Signal Proc.*, pp. 1-5, 2011.

[16] F. Pedersoli, S. Benini, N. Adami, and R. Leonardi, "XKin: An Open Source Framework for Hand Pose and Gesture Recognition", *Visual Computer*, vol. 30, pp.1107-1122, 2014.

[17] M. Vanko, I. Minarik, and G. Rozinaj, "Evaluation of Static Hand Gesture Algorithms", *Int. Conf. Systems, Signals, and Image Processing*, Dubrovnik, Croatia, pp. 83-86, 2014.

[18] E. Keogh and C. A. Ratanamahatana, "Exact Indexing of Dynamic Time Warping", *Knowledge and Information Systems*, vol. 7, no. 3, pp. 358-386, Springer, 2005.