

A Gesture Recognition Method for Proximity-Sensing Surfaces in Smart Environments

Biying Fu^{1(✉)}, Tobias Grosse-Puppendahl¹, and Arjan Kuijper^{1,2}

¹ Fraunhofer IGD, Fraunhoferstr. 5, 64283 Darmstadt, Germany
{biying.fu,tobias.grosse-puppendahl}@igd.fraunhofer.de

² Technische Universität Darmstadt, Hochschulstr. 10, 64289 Darmstadt, Germany
arjan.kuijper@gris.tu-darmstadt.de

Abstract. In order to ease the daily activities in life, a growing number of sophisticated embedded systems is integrated into a users environment. People are in need to communicate with the machines embedded in the surroundings via interfaces which should be as natural as possible. A very natural way of interaction can be implemented via gestures. Gestures should be intuitive, easy to interpret and to learn. In this paper, we propose a method for in-the-air gesture recognition within smart environments. The algorithm used to determine the performed gesture is based on dynamic time warping. We apply 12 capacitive proximity sensors as sensing area to collect gestures. The hand positions within a gesture are converted into features which will be matched with dynamic time warping. The gesture carried out above the sensing area are interpreted in realtime. Gestures supported can be used to control various applications like entertainment systems or other home automation systems.

Keywords: Gesture recognition · Dynamic time warping · Capacitive proximity sensing

1 Introduction

In-the-air gesture recognition within smart environments offers a number of highly promising application scenarios. They range from increasing hygiene in public restrooms to touchless interactions with infrastructure, such as doors. In this paper, we investigate the use of a proximity-sensing surface in smart environments. Being based on capacitive sensing, it can detect human interactions within distances of up to 30 cm [4]. The surface can be attached to walls, placed within doors, or integrated underneath tables. A particular challenge is the design of computationally-cheap algorithms for recognizing gestures. In this work, we identify relevant gestures based on a number of potential use-cases and propose a generic method for gesture recognition. It employs computationally inexpensive algorithms that can be implemented on low-cost embedded systems.

Latest developments in our society lead to the use of smart technologies that simplify everyday activities in life. More and more applications in the areas of human machine interfaces are demanded, all having the same goal of sensing human interactions in a more or less natural manner. Capacitive gesture-recognition systems are able to fulfill this need while offering highly interactive system designs at low cost. A great benefit of capacitive sensing is the ability of being installed unobtrusively under any non-conductive surface.

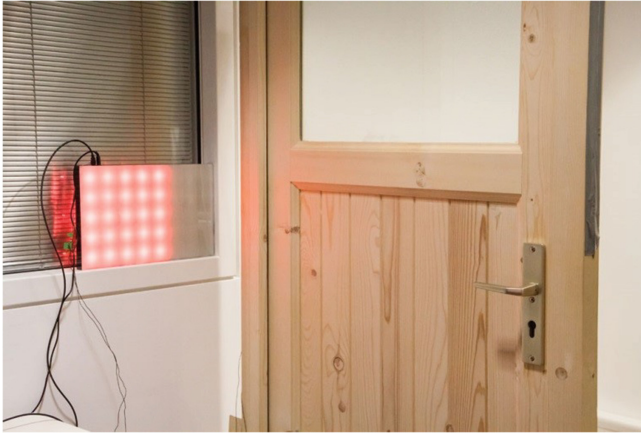


Fig. 1. A proximity-sensing surface acting as a door opener from [5]. Users are able to open and close the door based on swipe-gestures.

In order to specify the requirements for our proposed gesture-recognition method, we identified a number of use-cases.

1. *Interaction with a smart door* (shown in Fig. 1): Due to hygiene consideration in a public restroom, it will be practical to open, lock, unlock and close the door without touching the doorknob. The state of the door (e.g. locking it) can be easily changed using simple hand gestures, as introduced in [5].
2. *Controlling roller blinds*: A proximity-sensing surfaces can be used to open and close the roller. Here, vertical swipe gestures offer a natural way of controlling the appliance.
3. *Controlling Entertainment Systems*: Those supported gestures can be used to control a music player or other entertainment applications.
4. *Soft authentication in restricted areas*: By carrying out an authentication gesture, doors can be locked or unlocked. Moreover, alarms can be switched off in combination with occupancy detection.
5. *Controlling lights and illumination*: Gestures in front of a proximity-sensing surface can be employed to turn lights on and off. Moreover, circular gestures allow to dim the lights to ones needs.

2 Related Work

Providing means for natural interaction is an important goal when designing smart environments. For example, the user could switch on a standing lamp just with a simple hand gesture when she or he enters the room. It is also possible to use the whole body for interaction, for example by analyzing postures on furniture [6]. However, in this paper we aim at recognizing gestures carried out by a human hand. Many different modalities have been applied for gesture recognition, many of them based on cameras like the Microsoft Kinect [9]. Other modalities include capacitive approaches, for example by using body-attached electric field sensors [3], or ultrasound [7].

Camera based gesture recognition uses image processing and statistical methods like HMMs or DTW to perform gesture recognition. However, computer vision approaches are computationally expensive, as the bandwidth of information is very large. Here, the challenge is to efficiently extract information needed in a short time, in order to perform live gesture recognition. Capacitive sensing on the other hand is low power and efficient. Using this modality, several gesture recognition methods have been investigated, like [2] or the Swiss-Cheese Extended Algorithm presented in [4]. In the latter work, the authors use models to eliminate areas, in which no object may exist. Object tracking is performed with a particle filter to measure which predicts the new user's hand position above the sensing area. The algorithm is able to recognize and track multiple hands in real time. As this approach may not be executed on a microcontroller, we will present an approach based on Dynamic Time Warping (DTW).

DTW is a widely used method to perform gesture recognition. In [10], the authors developed a microcontroller-optimized implementation to warp a long common subsequence with a reference sequence and feedback the spotted subsequence in real time. This implementation can also be used to detect the QRS complex in a long ECG time signal as well as to detect predefined gestures within a time sequence. Very similarly, we will use DTW as a basis for our gesture recognition method in this paper. In the next chapter, the chosen method for gesture recognition above a capacitive sensing area will be explained in detail.

3 Proposed Gesture Recognition Model

As we intend the proximity-sensing surface to be low-cost and installed ubiquitously in a user's environment, our focus lies on computationally inexpensive algorithms. The implementation was realized based on the Rainbowfish platform [4], which is depicted in Fig. 1. It consists of 12 transparent electrodes each serving as a capacitive proximity sensor. The overall proximity sensing surface of the Rainbowfish has a dimension of 40 cm \times 25 cm containing 12 rectangular transparent electrodes used for determining the position of a human hand. It is also possible to feedback live performed user actions using LED lights integrated beneath the transparent platform, which can also be seen in the depicted figure. Object localization above the sensing surface is performed using a straightforward weighted averaging method developed by [1], which offers a fast way of

position calculation. To provide a smoother localization, a 2D position estimation Kalman Filter is also implemented. The estimated position by Kalman Filter is further improved by the measurement.

The next major step is the gesture recognition and thus its interpretation, in order to make interaction between user and their environment possible. With our proposed method we can quickly and almost confidently detect a set of simple hand gestures based on the traditional dynamic time warping method. All recognizable gestures so far using single hand is listed in the Fig. 2. In the following sections the implementation will be further explained in detail aiming to give you an better impression of how gesture recognition is done.

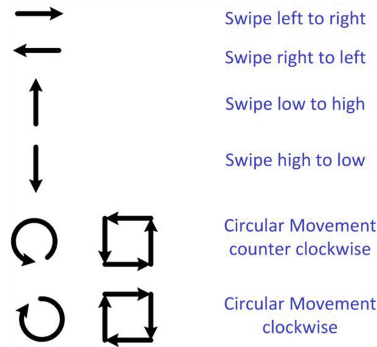


Fig. 2. Figure illustrates all confidently recognizable simple hand gestures using dynamic time warping method.

3.1 Dynamic Time Warping

The method of dynamic time warping presented in [8] is used to compare two time series, while one of them is usually based on a template database of reference hand gestures. In order to find the best match of a given time series compared to a template database, a cost function is calculated for two sequences prepared. The best match with the highest score, or the lowest cost, will be the intended hand gesture out of the predefined database. The mapping is performed in a nonlinear fashion, since the length of a performed gesture can be varied which depend on the gesture's speed. Therefore, the two time series could be non-linearly scaled in order to optimally match each other.

Following this approach brings in one constraint: the first element and the last elements of both time series should be mapped together, which is the so called boundary condition. Suppose we have two time series $A = (a_i)$ with Index $i = 1..N$ and $B = (b_j)$ with Index $j = 1..M$, whereas the length of both sequences could be different. We are looking for an optimal path between these two sequences with the smallest score, whereat (a_1, b_1) and (a_N, b_M) should be mapped together. The concept is illustrated in Fig. 3. The score matrix of

dimension $N \times M$ can be built comparing elements of both time series with each other. The path through the score matrix will always be the sum of the smallest score differences. Possible scores can be built using the Euclidean distance, some error measures or other self-defined scores adapted to the individual need.

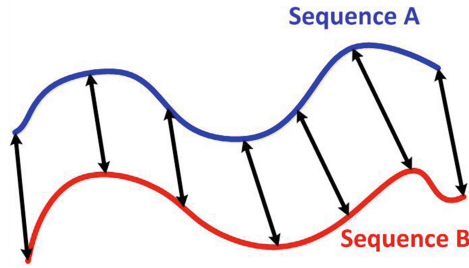


Fig. 3. Figure illustrates the method of dynamic time warping. Two different sequences A and B are aligned to each other in an optimum path using minimum score.

3.2 Implementation

As described in the introduction, a time series of hand positions will be sampled in time into a discrete sequence. Depending on the duration above the sensing area, the gesture can be of different lengths. Each short sequence within the gesture is converted into features, which is used to conduct the time warping method in order to interpret the performed gesture. In the following paragraph, the feature extraction will be explained in detail. The feature representation is illustrated in Fig. 4 with a simple circular chart diagram. The radial component of this circular chart represents the velocity component of the consisting part of a gesture. One single gesture is sampled in consisting hand positions above the sensing area. From one sample point to the successive sample point the velocity component will be calculated. If it is below a certain threshold, it will be interpreted as an indecisive slow movement and will be represented with the character *Z*. Otherwise, the angular movement of the velocity component will be calculated and mapped adequately to the appropriate angular character. The start of the gesture is set, if the user's hand is above the sensing area and thus the starting command will be filled with a character *S* symbolizing the start of this gesture. The ongoing gesture is evaluated as long as the gesture can be recognized and the final termination of the determined gesture can be set by leaving the sensing area. As soon as the user's hand leaves the sensing area, the end character *E* will be added to the command stream. An *E* can also be generated when the hand remains above a certain point for a longer time. This ensures that there is no obligation of leaving the surface with the hand. The definition of the used character can be found in Table 1.

Table 1. The meaning of the characters used in the dynamic time warping method.

Character	Definition
S	Start of a gesture
Z	Slow velocity component between two successive parts of a gesture
D	Angular component indicating horizontal movement from left to right
B	Angular component indicating horizontal movement from right to left
C	Angular component indicating vertical movement from high to low
A	Angular component indicating vertical movement from low to high
E	End of a gesture

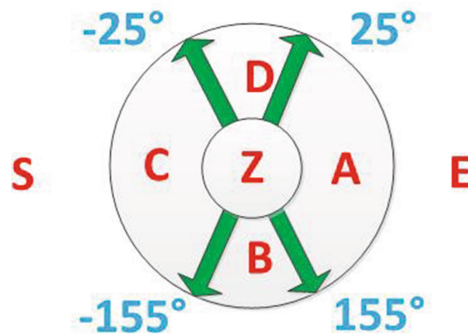


Fig. 4. The figure illustrates the way how the tangent of relative movement is mapped to the respective string character. A character *S* will be added, when the user's hand is detected on the sensing area for the first time and the character *E* will be added when no object is above the sensing plane. As long as the relative movement is small, the character *Z* is added, otherwise the other characters in the circle chart will be added accordingly.

The graphical interpretation of the angular distribution with respect to their corresponding string characters can be seen in Fig. 4. Due to the geometric property of the sensing area, where the length is broader than the width, it is reasonable to chose the angular distribution such that it is in favor of the horizontal movement. Caused by the larger x-axis with respect to the y-axis, the user has more freedom and precision by performing horizontal swipes.

An exemplary template for horizontal gesture moves from left to the right can be represented by a sequence like *SDDDE*, whereas real-world may also contain noise such as *SDDZZDDDE*. Therefore, the temporally stretched real-world strings will be compared with all possible reference command strings. The reference gesture with the lowest score and thus the highest matching score is the intended user gesture. One special cost function and it's distance function can be seen in the Figs. 5 and 6.

E	1	1	1	1	1	1	1	1	1	0
D	1	0	0	0	1	1	0	0	0	1
D	1	0	0	0	1	1	0	0	0	1
D	1	0	0	0	1	1	0	0	0	1
D	1	0	0	0	1	1	0	0	0	1
S	0	1	1	1	1	1	1	1	1	1
	S	D	D	D	Z	Z	D	D	D	E

Collected Example Sequence

Fig. 5. The cost function for the collected performed gesture on the x-axis with the reference string on the y-axis is depicted. The cost is 1, if the character is mismatched.

E	Inf	5	1	1	1	1	2	3	3	3	2
D	Inf	4	0	0	0	1	2	2	2	2	3
D	Inf	3	0	0	0	1	2	2	2	2	3
D	Inf	2	0	0	0	1	2	2	2	2	3
D	Inf	1	0	0	0	1	2	2	2	2	3
S	Inf	0	1	2	3	4	5	6	7	8	9
	U	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
		S	D	D	D	Z	Z	D	D	D	E

Collected Example Sequence

Fig. 6. The dist function is depicted in the figure. The yellow path follows the best alignment from the end of the sequence backwards to the beginning of the sequence (Color figure online).

With following assumptions, I used two additional weighting functions to further improve the cost of the dynamic time warping method, which are both of temporary and spatially natures. Since the sensing area is large, the gesture performed in the middle of the sensing area should be more intended and precise as on the boarder of the sensing area. Therefore the spatial weighting function will be given by the Eq. 1.

$$w(x, y) = 1 - A \cdot \exp \left(-\frac{(x - \frac{L}{2})^2}{2} - \frac{(y - \frac{W}{2})^2}{2 \cdot \sigma_y^2} \right) \quad (1)$$

The uncertainty in the y direction is larger, since as mentioned previously the geometric dimension of the x direction is larger than the y direction. In Eq. (1) L means the length

$$L = x_{max} - x_{min}$$

and W means the width

$$W = y_{max} - y_{min}$$

of the sensing area and A is a constant factor. The penalty is the smallest in the middle of the sensing area and enlarged at both sides as can be seen from Fig. 7. Furthermore I presume that the gesture in the middle of the time sequence is more intended and precise than at the beginning or at the end of a gesture. Suppose the length of the command sequence is L , then the weighting function can be give by Eq. 2.

$$w(n) = 1 - \frac{1}{\sqrt{2\pi}} \cdot \exp \left(-\frac{(n - \frac{L}{2})^2}{2} \right) \quad (2)$$

In Eq. 2 the index n stands for the index of character collected in time and L is the number of the overall gesture collected so far. The penalty is larger at

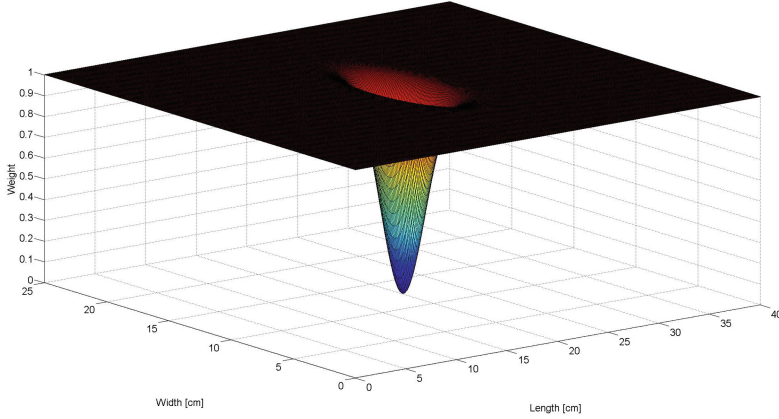


Fig. 7. The spatial weighting function in Eq. 1 with $W = 25$ cm, $L = 40$ cm and $\sigma_y = 1.6$.

the beginning and the end of a sequence, while the penalty is zero exactly in the middle of the sequence.

The software realization can be found in the flow chart in Fig. 8. The capacitive sensors keep actively measuring the activities above the sensing area. Once it detects the presence of a user's hand, the start character S will be added to the command sequence. Afterward it keeps reading sensor values to update the gesture. The corresponding string command keeps adding to the existing command sequence. The algorithm keeps detecting the gesture performed by the user in realtime, as long as the user's hand does not leave the sensing area. As soon as the user's hand leaves the sensing area, the last gesture will be analyzed and afterward the command sequence will be cleared, such that the system will be ready for a new gesture.

4 Validation and Interpretation

Based on a user study conducted with 10 different test persons, we evaluated the feasibility of our proposed method. Each test person was supposed to execute the presented gestures given in Fig. 2. Each gesture was performed ten times above the sensing area. The result is evaluated and summarized in the confusion matrix, which is shown in Table 2.

From the confusion matrix given in Table 2, we can see that the circular movements can be detected with a true positive rate of more than 98 %, while the other simple linear gestures can be assigned a true positive rate of more than 90 % as well. It is quite apparent, that the performed circular movements clockwise or anticlockwise are recognized with very high accuracy, while the simple linear movements are less accurate, but still with a detection rate of over 90 %. Simple linear movements are less error prone, since the capacitive sensing

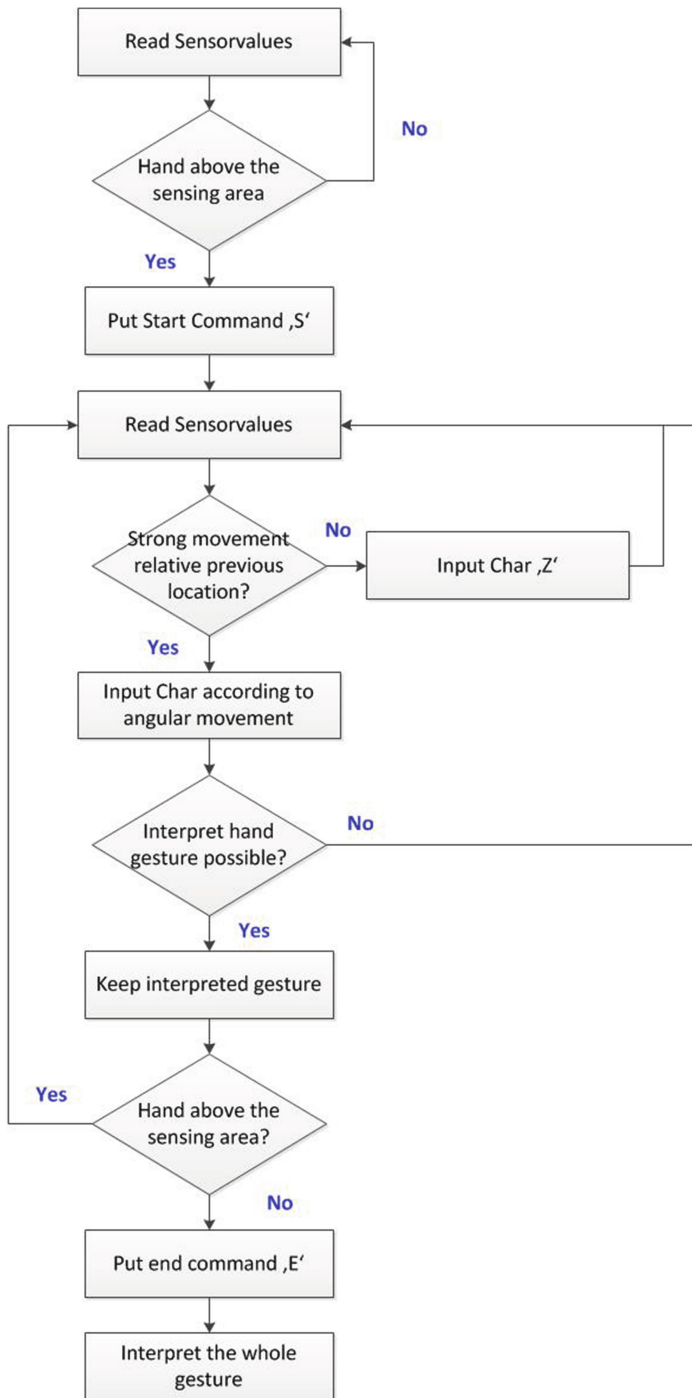


Fig. 8. Figure illustrates the flow chart of the software implementation.

Table 2. The table shows the confusion matrix.

True classified	\Rightarrow	\Leftarrow	\Uparrow	\Downarrow	\circlearrowright	\circlearrowleft
\Rightarrow	92		5	6		
\Leftarrow		96			1	1
\Uparrow	4	3	95	3		
\Downarrow	4	1		90		
\circlearrowright				1	98	1
\circlearrowleft					1	98

area is too sensitive, that it measures every tiny movements above the sensing area. The recognition rate is high, if the gesture are clearly performed.

To have a more precise expression of how the method works, we use the precision and recall matrix (Table 3).

Table 3. The table shows the precision and recall matrix.

Gestures	Precision	Recall
Swipe Right	0.89	0.92
Swipe Left	0.98	0.96
Swipe Upward	0.91	0.95
Swipe Down	0.95	0.90
Circular Clockwise	0.98	0.98
Circular Anticlockwise	0.99	0.98

5 Conclusion and Outlook

In this paper, the proposed gesture recognition was successfully realized using dynamic time warping method. A user study is conveyed and the results are evaluated and summarized. It showed that the circular movements clockwise or anticlockwise can be detected with very high accuracy, while the simple linear movements are somehow not so error prone. But all in all, the allowed gestures can be detected with quite high certainty in real time. The implementation is simple and can be coded on a simple micro-controller. In the near future, further goal is to expand gesture recognition with both hands accomplishing different more complicated gestures on the left and right side of the sensor board. We hope to allow more complex interactions with the environment, such as turning a virtual key gestures, and further more natural gestures performed with both hands.

References

1. Braun, A., Hamisu, P.: Using the human body field as a medium for natural interaction. In: Proceedings of the 2Nd International Conference on Pervasive Technologies Related to Assistive Environments, PETRA 2009, pp. 50:1–50:7. ACM, New York (2009). <http://doi.acm.org/10.1145/1579114.1579164>
2. Braun, A., Hamisu, P.: Designing a multi-purpose capacitive proximity sensing input device. In: PETRA 2011, pp. 151–158 (2011). <http://dl.acm.org/citation.cfm?doid=2141622.2141641>
3. Cohn, G., Morris, D., Patel, S., Tan, D.: Humantenna: Using the body as an antenna for real-time whole-body interaction. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 2012, pp. 1901–1910. ACM, New York (2012). <http://doi.acm.org/10.1145/2207676.2208330>
4. Grosse-Puppenthal, T., Beck, S., Wilbers, D.: Rainbowfish: Visual feedback on gesture-recognizing surfaces. In: CHI 2014 Extended Abstracts on Human Factors in Computing Systems, CHI EA 2014, pp. 427–430. ACM, New York (2014). <http://www.opencapsense.org/fileadmin/opencapsense-org/publications/chi2014.pdf>
5. Grosse-Puppenthal, T., Beck, S., Wilbers, D., Zeiß, S., von Wilmsdorff, J., Kuijper, A.: Ambient gesture-recognizing surfaces with visual feedback. In: Streitz, N., Markopoulos, P. (eds.) DAPI 2014. LNCS, vol. 8530, pp. 97–108. Springer, Heidelberg (2014). http://dx.doi.org/10.1007/978-3-319-07788-8_10
6. Große-Puppenthal, T.A., Marinc, A., Braun, A.: Classification of user postures with capacitive proximity sensors in AAL-environments. In: Keyson, D.V., Maher, M.L., Streitz, N., Cheok, A., Augusto, J.C., Wichert, R., Englebienne, G., Aghajan, H., Kröse, B.J.A. (eds.) Aml 2011. LNCS, vol. 7040, pp. 314–323. Springer, Heidelberg (2011). http://dx.doi.org/10.1007/978-3-642-25167-2_43
7. Gupta, S., Morris, D., Patel, S., Tan, D.: Soundwave: Using the doppler effect to sense gestures. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 2012, pp. 1911–1914. ACM, New York (2012). <http://doi.acm.org/10.1145/2207676.2208331>
8. Kruskal, J.B., Liberman, M.: The symmetric time-warping problem: from continuous to discrete. In: Sankoff, D., Kruskal, J.B. (eds.) Time Warps, String Edits, and Macromolecules - The Theory and Practice of Sequence Comparison, chap. 4. CSLI Publications, Stanford (1999)
9. Pheatt, C., Wayman, A.: Using the xbox kinect™ sensor for gesture recognition. J. Comput. Sci. Coll. **28**(5), 226–227 (2013). <http://dl.acm.org/citation.cfm?id=2458569.2458617>
10. Roggen, D., Cuspinera, L.P., Pombo, G., Ali, F., Nguyen-Dinh, L.-V.: Limited-memory warping LCSS for real-time low-power pattern recognition in wireless nodes. In: Abdelzaher, T., Pereira, N., Tovar, E. (eds.) EWSN 2015. LNCS, vol. 8965, pp. 151–167. Springer, Heidelberg (2015)