Human–Computer Interaction for Smart Environment Applications Using Fuzzy Hand Posture and Gesture Models

Annamária R. Várkonyi-Kóczy, *Fellow, IEEE*, and Balázs Tusor

***Abstract*—Ever since the assemblage of the first computer, efforts have been made to improve the way people could use machines. Recently, the usage of smart environments has become popular in order to make everyday living more comfortable and to improve the quality of living of humans. In this paper, a hand posture and gesture modeling and recognition system is introduced, which can be used as an interface to make possible communication with smart environment (intelligent space) by simple hand gestures. The system transforms preprocessed data of the detected hand into a fuzzy hand-posture feature model by using fuzzy neural networks and based on this model determines the actual hand posture applying fuzzy inference. Finally, from the sequence of detected hand postures, the system can recognize the hand gesture of the user.**

***Index Terms*—Fuzzy neural networks, hand-gesture recognition, hand-posture modeling, intelligent space, iSpace, smart environment.**

I. Introduction

TODAY, computers play an increasing role in our everyday life. The applications of intelligent systems that aim to improve the living conditions and quality of everyday life are also gaining more and more importance. In ideal case, these systems are realized in such a way that the usage of the systems becomes as easy as possible, while the presence of the system does not bother its user at all. This leads to the basic idea of "*ubiquitous computing*," which is proposed by [1].

One example for ubiquitous computing applications is the *intelligent space* (iSpace) [2], which has been developed at the University of Tokyo. iSpace is a room or area that has built-in intelligence: It can monitor the events and actions taking place in the room or area (railway station, underpass, road crossing, or even a town, etc.), it can comprehend human interactions, and it is able to react to them. To mention some examples, it can share information with the user, help in orientation, or anticipate crisis situations. The user can also give commands to the iSpace to use certain services. Therefore, the system should be easy to use for the people in it, without the need for them to learn how the system is to be used.

The most characteristic feature of the iSpace is that the intelligence is distributed in the whole space, not in the individual agents. The main advantages of this component-based architecture are that the iSpace can be easily constructed or modified by the installation or replacement of so-called distributed intelligent networked devices (DINDs) responsible for monitoring a part of the space, processing the sensed data, making local decisions, and communicating with other DINDs or agents if necessary. The agents in the space do not have to possess any complex logic.

Any room or area can be converted to an iSpace by installing DINDs into it. However, when building iSpace into an existing area, we have to keep in view that the system should be human centered, it should not be disturbing for the people who are using it, and the installation should not overly alter the area.

There are several iSpace applications currently developed or planned. These include the positioning and tracking of humans, the monitoring of the physiological functions of elderly people, and the localization and control of mobile robots, finding paths for them by using itineraries taken by people (see, e.g., [2]–[4]).

Nowadays, it has become an essential expectation toward smart environments and the iSpace to make communication with the environment possible for the user using simple natural signs. For this, one example can be the usage of hand gestures. In this paper, the authors present a hand posture and gesture modeling and recognition system that is able to determine the shape of the detected hand based on the coordinate model computed by the hand detection and tracking system proposed in [5]. Hence, creating an intuitive interface for the iSpace makes it possible for the user to communicate with the environment by using hand gestures.

Recently, several systems have been reported and implemented for hand-gesture recognition. Here, we briefly summarize three characteristic approaches that, similar to the introduced system, use images made by one or more cameras.

The system proposed by [6] is based on the idea of a divide-and-conquer strategy: It breaks hand gestures to basic hand

Manuscript received June 21, 2010; revised November 20, 2010; accepted December 20, 2010. Date of current version April 6, 2011. This work was supported in part by the Hungarian National Scientific Fund under Grant OTKA 78576 and in part by the Structural Fund for Supporting Innovation in New Knowledge and Technology Intensive Micro- and Spin-off Enterprises under Grant GVOP-3.3.1-05/1.2005-05-0160/3.0. The Associate Editor coordinating the review process for this paper was Dr. Gilles Mauris.

A. R. Várkonyi-Kóczy is with the Institute of Mechatronics and Vehicle Engineering, Óbuda University, Népszínház u. 8, 1081 Budapest, Hungary, and also with the Integrated Intelligent Systems Japanese-Hungarian Laboratory, Népszínház u. 8, 1081 Budapest, Hungary (e-mail: varkonyi-koczy@uni-obuda.hu).

B. Tusor is with the Integrated Intelligent Systems Japanese-Hungarian Laboratory, Népszínház u. 8, 1081 Budapest, Hungary (e-mail: balazs.tusor@gmail.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIM.2011.2108075

postures and movements, which can be treated as primitives for syntactic analysis based on grammars. The system consists of a two-level architecture that decouples hand-gesture recognition into two stages, i.e., the low-level hand-posture detection and tracking and the high-level hand-gesture recognition and motion analysis. For the low-level hand-posture detection, it uses a statistical approach based on Haar-like features, whereas for the high-level hand-gesture detection, it employs a syntactic approach based on the linguistic pattern recognition technique to fully exploit the composite property of hand gestures. The detected hand postures and motion trajectories are sent from the low level to the high level of the architecture as primitives for the syntactic analysis so that the whole gesture can be recognized according to the predefined grammars.

In the system proposed in [7], pseudo-two-dimensional hidden Markov models (P2-DHMMs) are used for the hand-gesture recognition. The basic idea is the real-time generation of gesture models for hand-gesture recognition in the content analysis of video sequence from a charge-coupled-device camera. To avoid the problem caused by the exponential complexity of the algorithm of fully connected 2-DHMMs, the connectivity of the network has been reduced in several ways, thus gaining P2-DHMMs, which retains all of the useful HMMs features. The models can be similarly trained to neural networks.

The fuzzy neural network architecture proposed by [8] is based on incorporating the idea of fuzzy adaptive resonance theory mapping (ARTMAP) [9] in feature recognition neural networks [10]. The inputs of the neural network consist of fuzzy membership function values, which are determined from the gray-level value of each pixels of the monochrome image. A gesture recognition fuzzy neural network is used (with four layers, excluding the input layer) for feature recognition. The implemented system offers a flexible framework for gesture recognition and can be efficiently used in scale invariant systems.

The approach presented in this paper has several novelties and advantages compared with known techniques. As a feature model, it introduces a new *fuzzy hand-posture model* (FHPM). For hand-posture recognition, a modified *Circular Fuzzy Neural Network* (CFNN) architecture is proposed together with a reduced time training procedure. As a result, compared with other solutions, the robustness and reliability of the hand-gesture identification is improved, and the complexity and training time of the used neural nets are significantly decreased. For details, also see [11].

The rest of this paper is organized as follows: In Section II, the authors give a brief overview of the procedure that detects the human hand and produces the input data for the new recognition system proposed in this paper. Section III presents the fuzzy hand-posture model, which is an intuitive hand-posture model that is used for modeling the human hand. In Section IV, an overview of the proposed system is given together with the detailed description of its parts and of the functioning of the system. In Section V, an experimental system is briefly shown, and with its help, we analyze the performance of the new technique and make comparison with other hand gesture recognition methods. Finally, Section VI concludes this paper and outlines possible improvements.

II. Hand Detection and Data Preprocessing

The comfortability of the way of communication between humans and the environment is of vital importance from the point of view of the usability of the system. The more natural the interaction is, the wider the applicability can be. Because of this, we decided to use one of the basic human talents, i.e., coordinated complex moving of the hands, for communication. The idea is that after the sensors of the iSpace detect and separate the hands of the humans, the intelligence of the system determines the postures and movements and translates them to the desired actions.

For hand tracking and recognition, the system presented in [5] is used. It detects the human hand by using two cameras, which are monitoring the same scene from two different positions and preprocesses the data into a 3-D coordinate model. The output of the hand tracking and recognition system, i.e., the 3-D model of the hand consisting of the spatial location of the feature points of the detected hand, serves as the input for the hand-posture detection system proposed in this paper.

The procedure works the following way: First, it locates the areas in the pictures of the two cameras where visible human skin can be detected using histogram back projection [12]. This method locates areas with a given color model. Considering Hue Saturation Value (HSV)

$$\text{backproj}(x, y) = H\left(\text{hue}(x, y)\right) \tag{1}$$

where backproj means the single channel backprojection image, H is the hue histogram, and hue is the single channel hue plane of the input image. This means that to a given color model (histogram) of a given object (human skin), as a result, the probability distribution of skin is obtained.

The next step is the extraction of feature points considering curvature extrema, peaks (e.g., fingertips), and valleys (such as the roots of the fingers) (see [13] and [14]). After that, the selected feature points are matched in a stereo image pair. The matching separately occurs for the peaks and the valleys using the fuzzy-based matching algorithm proposed by [14]. It works in the following way: First, the candidate pairs (i.e., points lying within a given fuzzy neighborhood of their corresponding epipolar lines) are located. Then, for each candidate point pair detected in the stereo image pair, the sum of the differences of their neighborhoods weighted by a 2-D fuzzy set is calculated, i.e.,

$$\sum_{\substack{x,y\in w\\ x',y'\in w'}} |I(x,y) - I'(x',y')| \cdot \mu_a(x',y') \cdot \mu_b(x',y') \tag{2}$$

where $I(x, y)$ and $I'(x', y')$ mean the intensities in the left and right images, respectively, and μ_a and μ_b denote the membership functions used as weighting functions. The environments of the feature points are denoted by w and w'. The pairing that yielded the smallest sum of the weighted difference is considered to be matching.

For improving the quality of the hand model and that of the posture identification, aside from the main feature points, we use further so-called subdivision points as well. The subdivision points are determined in the left frame as one or more

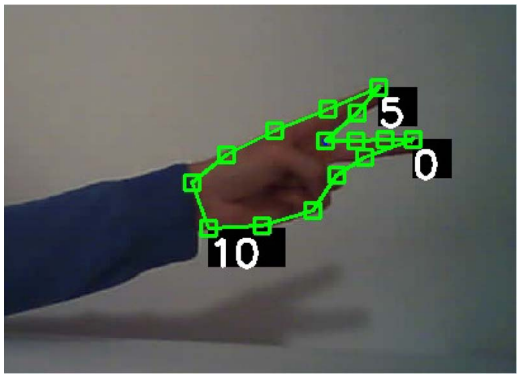


Fig. 1. Coordinate model of a detected hand posture.

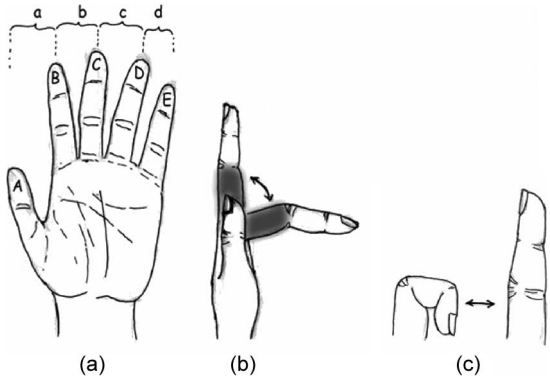


Fig. 2. Three types of fuzzy feature sets.

division points of the sections determined by two consecutive feature points. The subsections have the same length within one section, but not necessarily all of them are of the same length (here, the length of a section means the number of contour points it includes). The number of subdivision points, for each section, is proportional to the length of that section. The total number of subdivision points depends on the number of feature points: These two values, together, should yield a constant number, which equals the number of inputs of the CFNNs, as described in Section III. In case of complete lack of feature points, the whole contour is equidistantly subdivided. The subdivision points of the left frame are then matched against contour sections on the right frame, again, using the fuzzy point-matching algorithm [14].

Finally, the 3-D coordinates of the feature points are calculated using the known camera matrices (for further details, see [5]).

The result of the procedure consists of 15 spatial coordinate points. Fig. 1 shows an example for the visualized results of the procedure. From the detected spatial coordinate model, our system determines the hand posture and recognizes the hand gesture from the identified sequence of hand postures.

III. FHPM

In order to efficiently distinguish different hand postures, we have developed the FHPM. It describes the human hand by fuzzy hand feature sets.

Three different types of fuzzy feature sets had been appointed, each one describing a certain type of features of the given hand posture (see Fig. 2). The first set consists of four fuzzy features; they describe the distance between the fingertips of each adjacent finger. Both the second and the third sets consist of five–five fuzzy features: The former describes the bentness of each finger, whereas the latter describes the relative angle between the bottom finger joint and the plane of the palm of the given hand. Thus, every feature set type is an answer to a certain question as follows.

- How far are fingers X and Y from each other?
- How big is the angle between the lowest joint of finger W and the plane of the palm?
- How bent is finger z?

Each feature is marked by a linguistic variable that can only have one of the three following values: small, medium, or large.

Fig. 3. Classic hand posture for "victory."

[We also tested models having two and five fuzzy sets/universe. The choice of applying three fuzzy sets in each universe is the result of finding an optimum, taking into consideration the rate of correct identification (which is slightly higher in the case of the application of only two sets/universe where the "distance" between two neighboring linguistic values is bigger) and also the possible increase in performance caused by the flexibility of the higher number of possible models (in this respect, applying five fuzzy sets/universe overperforms the others).]

With the proposed structure, a hand posture can be efficiently described by these 14 features in a way that is easily readable and interpretable by human users as well. For each FHPM, the 14 linguistic variables (which are used to describe the 14 features previously introduced) are stored in the *ModelBase*. With this model, we can theoretically distinguish 3^{14} different hand postures of which number is much higher than the realistically conceivable need of even the most complex sign languages. However, the 3^{14} different possibilities make the system very flexible with respect to the dissimilarities of the distinguished hand postures, thus increasing the reliability of the recognition procedure.

In the following, an example is presented to demonstrate how a hand posture can be described by the FHPM features.

Fig. 3 shows the well-known victory sign. The relative position of the fingers from each other and their bentness is clearly visible. From that, we can determine the values of all the 14 linguistic variables. Table I shows the values of the variables, which are divided into the three different feature groups.

IV. Hand Posture and Gesture Identification System

For the hand-posture recognition, the proposed system uses fuzzy neural networks and fuzzy inference. The idea is based on transforming the coordinate model of the detected hand into

TABLE I
FEATURES FOR HAND POSTURE "VICTORY"

Feature group	Feature	Value
Relative distance between adjacent fingers	***a***	Large
	b	Medium
	c	Small
	d	Small
Relative angle between the lowest joint of each finger and the plane of the palm	***A***	Medium
	B	Small
	C	Small
	D	Large
	E	Large
Relative bentness of each finger	***A***	Medium
	B	Large
	C	Large
	D	Small
	E	Small

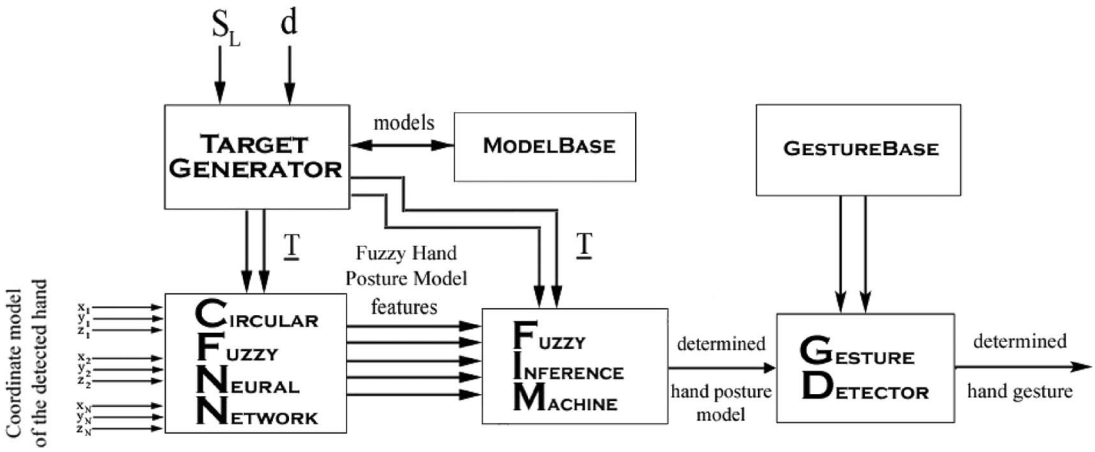


Fig. 4. Block diagram of the proposed hand posture and gesture identification system.

an FHPM by *CFNNs* and then to identify it by applying fuzzy inference.

The system consists of the following six modules: 1) ModelBase; 2) *GestureBase*; 3) *Target Generator*; 4) CFNNs; 5) *Fuzzy Inference Machine* (FIM); and 6) *Gesture Detector*.

Fig. 4 shows the architecture of the system. The system receives the coordinate model of the detected hand as input, transforms it into an FHPM using CFNNs, and then determines the hand's shape from the FHPM with the usage of the FIM module. The Gesture Detector module observes the output of the FIM and searches for matches with predefined patterns in the gesture base. The hand gesture is identified in case of fuzzy matching with any stored hand gesture or is refused if the hand gesture is unknown for the gesture base.

A. ModelBase

The features of the models are stored in the ModelBase as linguistic variables (with possible values of small, medium, and large). One of the easiest realizations of the ModelBase can be, for example, using extensible-markup-language files. Fig. 5 shows an example for two predefined models.

B. GestureBase

The GestureBase contains the predefined hand gestures. Each hand gesture is identified by its name and is described by up to five string values, which are existing FHPM names. The string values are listed in the *sequence* parameter. The parameter

```
<base>
    <model>
        <name> Open_hand </name>
        <value_a> large large large large </value_a>
        <value_b> small small small small small </value_b>
        <value_c> large large large large large </value_c>
    </model>
    <model>
        <name> Fist </name>
        <value_a> small small small small </value_a>
        <value_b> medium large large large large </value_b>
        <value_c> medium small small small small </value_c>
    </model>
    <model>
        <name> Three </name>
        <value_a> medium medium medium small </value_a>
        <value_b> small small small large large </value_b>
        <value_c> large large large small small </value_c>
    </model>
</base>
```

Fig. 5. Example for the ModelBase with three defined models.

```
<base>
    <gesture>
        <name> Classic </name>
        <quantity> 3 </quantity>
        <sequence> Open_hand Fist Three </sequence>
    </gesture>
</base>
```

Fig. 6. Example for the GestureBase with one defined gesture model.

quantity denotes how many FHPMs the given hand-gesture model consists. Fig. 6 shows an example for the realization of the GestureBase.

C. Target Generator

The Target Generator is used to calculate the desired target parameters for the CFNNs and the FIM using the ModelBase. The input parameters of the Target Generator module are as follows.

1) d is the identification value (ID) of the model in the ModelBase. It can be simply the name of the given model or an identification number.
2) S_L is a linguistic variable for setting the width of the triangular fuzzy sets, thus tuning the accuracy of the system.

In our experiments, we used triangular-shaped fuzzy sets with centers set to the following numerical values: 1) small $=$ 0.3; 2) medium $=$ 0.5; and 3) large $=$ 0.7.

The other tuning parameter of the fuzzy sets is their width. This parameter is explicitly settable through a linguistic variable S_L. In the case of S_L chosen to small, medium, or large, the widths of the fuzzy feature sets equal to 0.33, 0.66, and 1, correspondingly.

Equation (3) is used to compute the ends of the interval that represents the given fuzzy feature set, i.e.,

$$(a, b) = (\text{center} - 0.2 \cdot \text{width}, \text{center} + 0.2 \cdot \text{width}) \qquad (3)$$

where a, b denotes the lower and upper ends of the interval, *center* corresponds to the numerical value of the given feature, and *width* equals to the numerical value of the linguistic variable S_L (see Fig. 7).

D. CFNNs with Interval Arithmetics

For the task of converting the coordinate model of the given hand to an FHPM, we have developed *CFNNs* based on fuzzy neural networks with interval arithmetic proposed by [15]. For training the CFNNs, a standard backpropagation algorithm has been used. The novelty of our network lies in its modified topology, as is shown in this subsection.

The networks have fuzzy numbers in their weights, biases, and in the output of each neuron. Each fuzzy number is represented by the interval of its alpha cut at value 0, i.e., by the base of the triangular fuzzy set. Thus, each fuzzy number is determined by two real numbers: the lower and the upper ends of the given interval. Fig. 8 shows the basic topology of the fuzzy neural networks.

Since the coordinate model consists of 15 3-D coordinates, the neural networks have 45 inputs. In order to increase the robustness, instead of using one network with 14 output layer neurons, three different networks are used with the only difference in the number of the output layer neurons. The first network computes the first group of fuzzy features, having four neurons in the output layer. The second and third networks similarly have five–five neurons in the output layer. Furthermore, all the networks consist of one hidden layer with 15–15 neurons.

In order to enhance the speed of the training session and to take advantage of the fact that the input data consists of coordinate triplets, the topology of the network has been modified. Originally, each input was connected to all hidden layer neurons. In the modified network topology, only nine inputs (corresponding to three adjacent coordinate triplets) are connected to each.

This way, every hidden layer neuron processes the data of three adjacent coordinates. The topology between the hidden layer and the output layer neurons has not been changed. This way, the network has been realigned into a *circular* network (see Fig. 9). For the sake of better visibility, not every connection is shown in Fig. 9. In the outer circle layer, the input coordinates can be found, in the middle circle, the hidden layer neurons are placed, and in the inner circle, the output layer neurons of the network are located.

These reductions cause a dramatic decrease in the required training time, whereas the precision and accuracy of the networks are not affected.

E. Fuzzy Reasoning Based Classification

The last step of the hand-posture identification procedure is the fuzzy-reasoning-based classification. This part compares the output of the CFNNs to all the FHPMs stored in the ModelBase and chooses the model that corresponds the most to the model presented by the CFNNs.

The algorithm works as follows: For each model in the ModelBase, we calculate the intersection value between their fuzzy feature sets and the fuzzy feature sets of the given FHPM, thus gaining β_i for each of its features. The minimum of β_i shows the correspondence ratio between each model and the given FHPM. The maximum value of the correspondence ratios indicates which model corresponds the most to the detected hand posture.

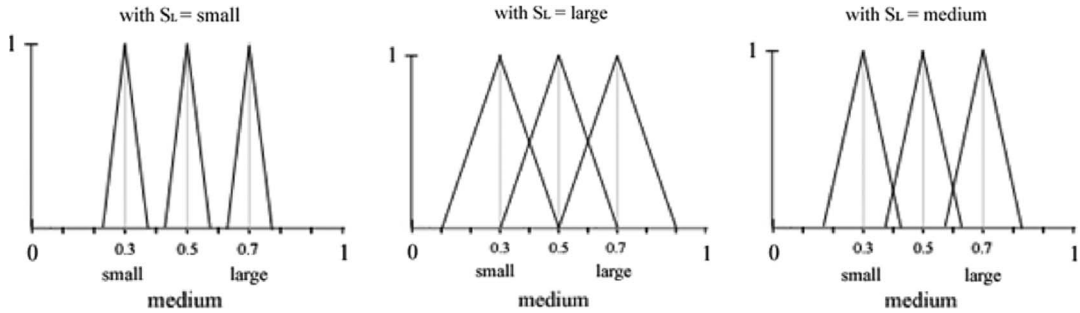


Fig. 7. Possible settings of the triangular-shaped fuzzy feature sets.

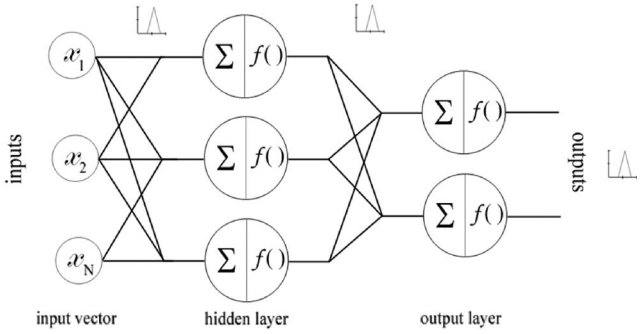


Fig. 8. Topology of the fuzzy neural networks.

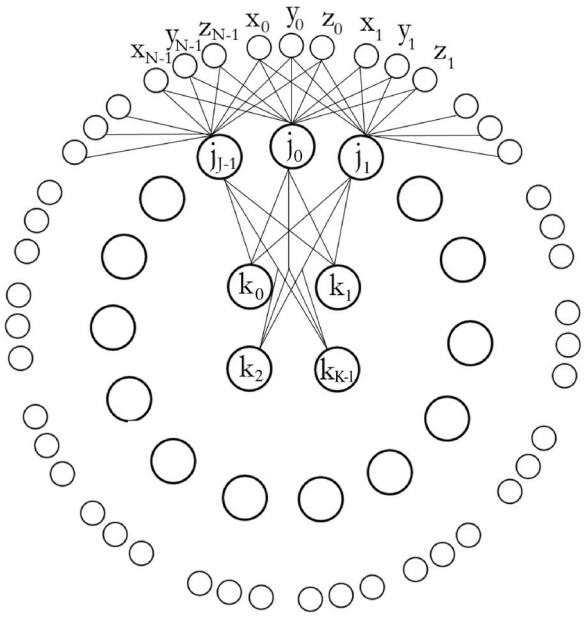


Fig. 9. Topology of the CFNNs.

F. Gesture Identification

The ID of each recognized hand-posture model is put in a queue, which is monitored by the Gesture Detector module. It searches for hand-gesture patterns predefined in the Gesture-Base, and in case of matching, it gives the ID (in our realization: the name) of the detected hand gesture as the output of the system, i.e., it identifies the gesture.

Here, we would like to remark that the reliability of the identification system can be improved by applying a "fault tolerant" sign language: The high number (3^{14}) of possible FHPMs offers an easy way to choose the meaningful hand postures (i.e., the elements of the sign language) in such a way that there is a dissimilarity in more than two fuzzy features between any two used models, or in the same way, when there is a $d_H > 2$ Hamming distance among the meaningful hand-posture series. In this case, a detected unknown hand posture or hand-posture sequence caused by false detection(s) can be corrected to the nearest known hand posture/gesture. This can be done by the following way: From the detected hand posture, a *similar* hand posture can be created by changing one fuzzy feature value to its neighboring value (in the case of three linguistic values: small to medium; medium to small or large; large to medium) and then run the FIM with the new hand posture as input. This is repeated for all the possible *similar* hand postures, the result of the inference will be the hand-posture model that corresponds the most to any of the *similar* hand postures. By this way, one false detection can be corrected if $d_H = 3$ is ensured, two false detection can be corrected if $d_H = 5$, etc.

G. Training Session

For increasing the speed of the training of the CFNNs, we have developed a new training procedure. In order to decrease the quantity of the training samples, instead of directly using the training data in the training phase, they are first clustered, and the CFNNs are trained by the centers of the obtained clusters (that are still 45 real numbers or 15 spatial coordinate points).

The clustering procedure (see Fig. 10) is based on the k-means method with a simple modification: Comparing a given sample to the clusters one by one, the sample gets assigned to the first cluster with the distance of its center to the given sample less than an arbitrary value. If there is no such cluster, then a new cluster is appointed with the given sample assigned to it. The clustering is used on only one type of hand model at one time, so the incidental similarity between different types of hand models will not cause problems in the clustering phase.

V. Experimental System and Results

A. Image Processing Component

We have built an experimental setup for testing and analyzing the performance of the image processing component. The

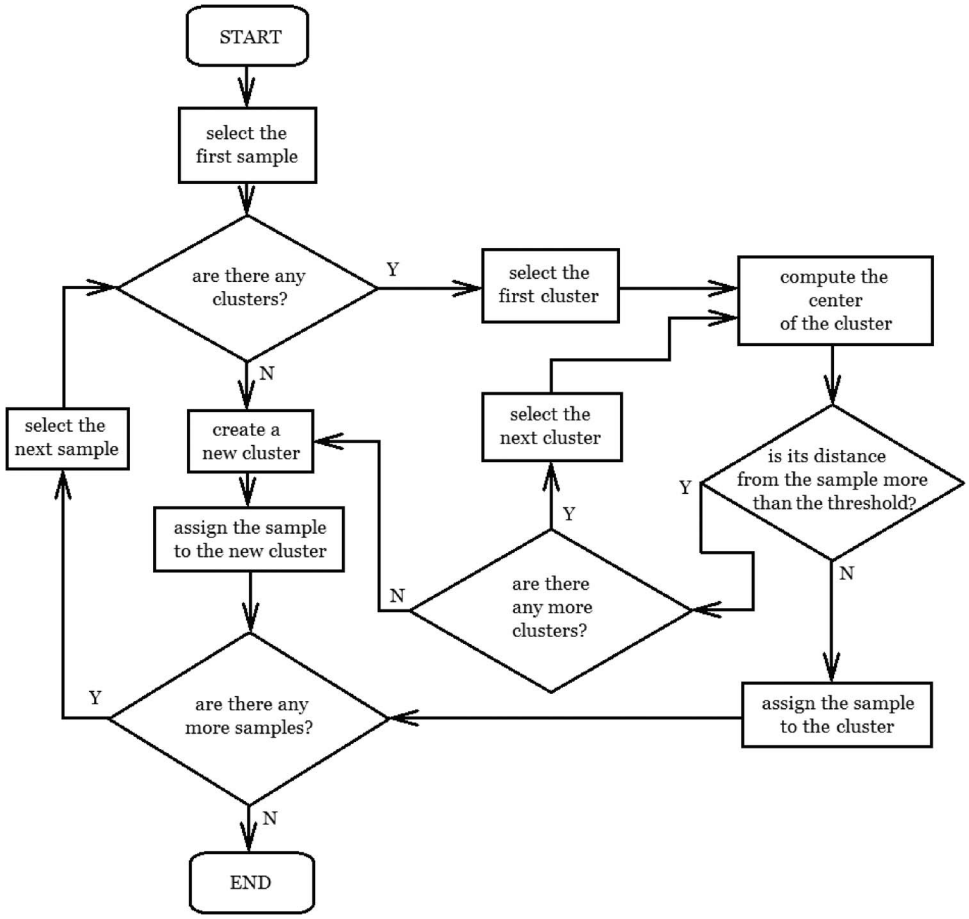


Fig. 10. Flowchart of the classification method.

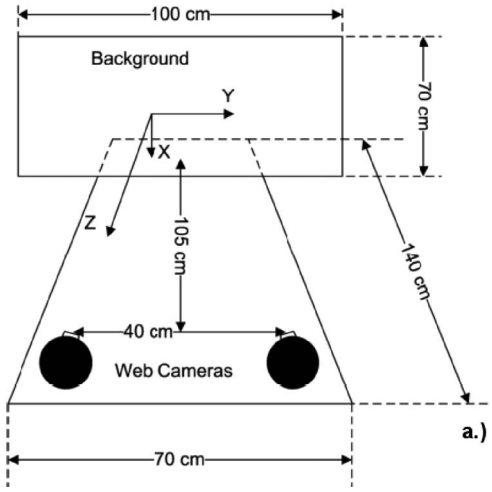


Fig. 11. Experimental setup.

hardware part of the testing system consists of two web cameras (Genius Look 316) connected to an average personal computer (PC) (central processing unit: AMD Athlon 64 4000+, 1 GB DDR RAM, Windows XP+SP3 operating system). The cameras are located at one end of a table. At the other end, a homogeneous background is mounted (see Fig. 11). The image processing algorithms are running on the PC and are implemented in C++ using *OpenCV* [16]. For camera calibration, the *Camera Calibration Toolbox for MATLAB* has been adopted [17].

Throughout the tests, the following parameters were used: The input frames were 8 bits/channel static red/green/blue images, at a resolution of 640 $\times$ 480 pixels. The working images (i.e., the backprojection and the grayscale representation for fuzzy matching) were single-channel images, with the same parameters. After backprojection, only pixels having brightness (i.e., "value," in the HSV color space) between 10 and 200 were retained. The thresholding parameter, before contour component extraction, was 60. In the fuzzy matching algorithm, the weighted difference was calculated over an area of 121 $\times$ 121 pixels, having the maximal weight at the central area of 80 $\times$ 80 pixels. The points lying within 20 pixels of the epipolar lines were considered to be matching candidates, and others were discarded. The experimental setup is shown in Fig. 11.

In order to assess the quality of the spatial model, 100 uniform samples of each hand posture were captured at various distances from the homogeneous background (starting at $d = 0$ cm, i.e., hand adjacent to the background, until $d = 40$ cm, stepping by 5 cm), which is a total of 5400 samples. The yielded spatial models were evaluated in formal and informal ways. The *formal approach* consisted of fitting a plane to the spatial model (which, in this case, was neither translated nor scaled) and calculating the average distance of the spatial points from this plane (here, the fitting plane is defined as the plane for which the squared sum of distances to the spatial points is minimal).

The *informal solution* consisted of manually examining the spatial model by converting it to the format of a third-party

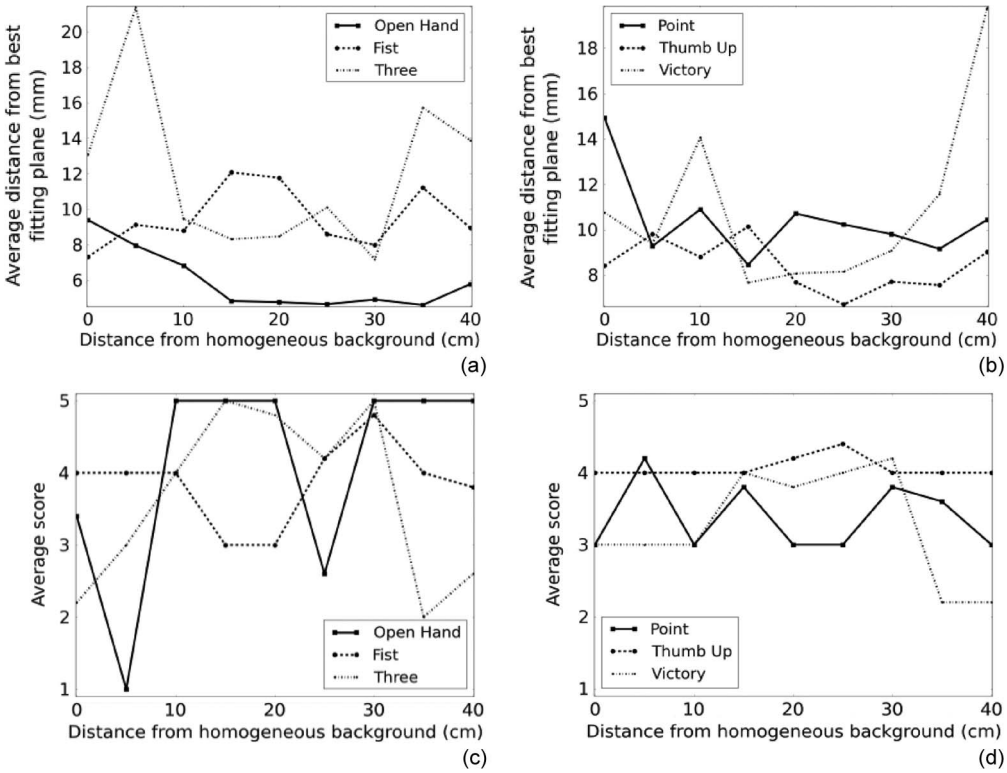


Fig. 12. Average distances and scoring of each measurement set: (a) Distance for postures "Open Hand," "Fist," and "Three." (b) Distance for postures for postures "Point," "Thumb up," and "Victory." (c) Scoring of postures "Open Hand," "Fist," and "Three." (d) Scoring of postures "Point," "Thumb up," and "Victory."

program, i.e., *Anim8or* [18]. Every 20th sample was examined in this way, and an integer score between 1 (worst) and 5 (best) was assigned to them. The results of both methods are shown in Fig. 12. In general, it is true, that sets having a lower average distance were found to be of "better" quality by the informal examination as well. For instance, models for posture "*Three*" at $d = 5$ cm contained more inaccuracies than those at $d = 10$ cm. In the case of posture "*Three*," at $d = 35$ cm and $d = 40$ cm, one point was in a wrong position in the right frame, yielding a considerable displacement also in the spatial model. In another case, however, the distance from the fitting plane was not useful for detecting an inaccuracy. All the examined samples of posture "*Open Hand*" at $d = 5$ cm, and three of them at $d = 25$ cm, exhibited a particular error: There were no spatial points representing the contour between the ring and the little finger, causing the two fingers to "blend together." However, since all of them were lying near the fitting plane, a low average distance was reported.

The time needed for each step has been measured by instrumenting the source code before and after the main steps and also at the beginning and at the end of the whole iteration. Then, this iteration was run 200 times on ten different samples of each posture, and the total time spent during each type of step was averaged. The results are shown in Fig. 13. As expected, it has been found that the bottleneck of the procedure is constituted by steps, which need to match distinct points against whole sections. Due to these operations, the average time needed by an iteration is about 0.78 s, and the maximal value was as high as 1.23 s. In order to use this system in real-time applications, ways to optimize the performance must be investigated.

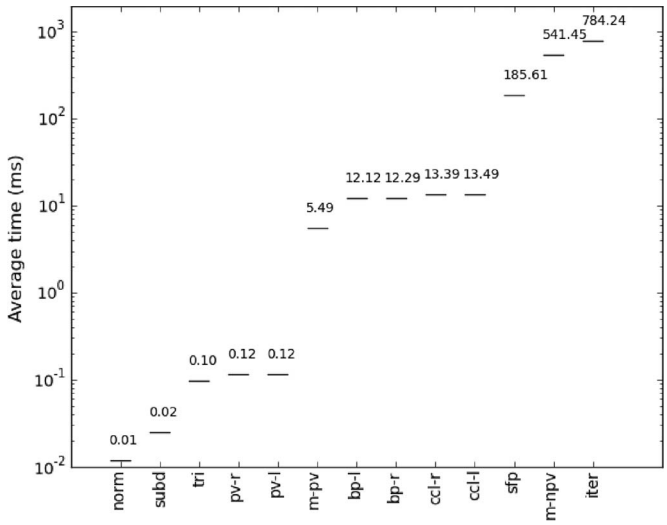


Fig. 13. Average times for various steps. norm: normalization of spatial points. subd: contour subdivision. tri: stereo triangulation. pv-{l, r}: peak and valley location on the left and right frame. m-pv: peak and valley matching. bp-{l, r}: back projection and discarding pixels by brightness (on the left and right frame, respectively). ccl-{l, r}: connected component localization after thresholding (on the left, and the right frame, respectively). sfp: selecting subdivision points as feature points. m-npv: matching subdivision points to contour sections (i.e., nonpeak and valley matching). iter: time for a complete iteration.

B. FHPM-based Hand Recognition

The experimental options for the training session of the CFNNs (and clustering) were the following:

- Learning rate: 0.8
- Coefficient of the momentum method: 0.5

TABLE II
INITIAL RESULTS OF THE TRAINING, INFERENCING WITH DIFFERENT S_L VALUES

Name of the	Quantity of chosen samples	Quantity of clustered	Quantity of testing set	Correctly identified	%
Open hand	20	10	80	75	93.7
Fist	20	13	80	78	97.5
Three	20	4	80	78	97.5
Thumb-up	20	7	80	78	97.5
Point	20	4	80	77	96.25
Victory	20	6	80	75	93.7

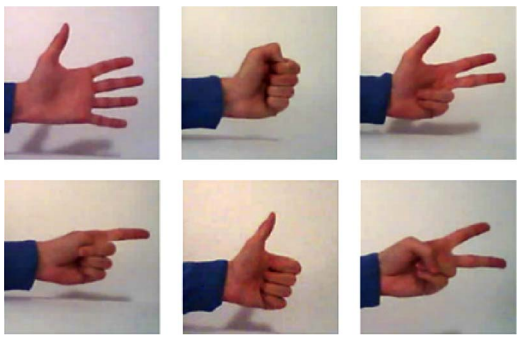


Fig. 14. Used hand postures.

- Error threshold: 0.1
- S_L: small
- Clustering distance: 0.5

Table II summarizes the identification results of a simple initial experimental system that is currently able to differentiate among six hand models (Open hand, Fist, Three, Thumb up, Point, and Victory). For better visualization, Fig. 14 shows these hand postures. The set of the generated coordinate model samples has been split into two separate sets, i.e., training and testing. The training set consists of 20 samples from each of the three FHPMs. The clustering procedure reduced the quantity of the training sets; thus, instead of 120, only 42 coordinate models were trained to the CFNNs. The networks are trained with choosing S_L = small. The FIM could identify the models in the testing set (with 80 samples of each model) with an average accuracy of 94.8%. (The testing set does not contain the training set.)

While the CFNNs were trained with the S_L = small value, in the inference phase, S_L = large has been used, which increased the accuracy of the FIM.

C. Analysis of the Results

The performance of the proposed system has been analyzed by the implemented experimental setup (see Fig. 11). This system works with six predefined hand postures (see Fig. 14) and gestures (as an example, see Fig. 6) consisting any series composed of these six postures. However, the number of used hand postures can be considered little, compared with the usual number of elements of the traditional sign languages; this is not the case. If we increase the number of meaningful hand postures, only the training time of the CFNNs will increase, and the reliability of the identification can be kept at the same level.

However, the correct identification rate proved to be in average above 96%, currently with a limitation of the need of using a homogenous background. This is possibly the most limiting factor of the application.

The reliability of the hand posture and gesture recognition can be further improved by applying the fault tolerant hand-posture and hand-gesture set definition concepts outlined in Section IV-F.

VI. CONCLUSION AND FUTURE WORK

In this paper, the concept and implementation of a hand posture and gesture modeling and recognition system have been

introduced. This interface makes human users to be able to control smart environments by hand gestures. The presented system is able to classify different simple hand postures and any hand gestures that consist of any combination of the predefined hand postures.

In our future work, we will extend the set of FHPMs. The hand-gesture identification method will be improved as well since the actual implementation of the Gesture Detector does not take into account the position of the hand, only the shape of it.

REFERENCES

[1] M. Weiser, "The computer for the twenty-first century," *Sci. Amer.*, vol. 265, no. 3, pp. 94–104, 1991.

[2] J.-H. Lee and H. Hashimoto, "Intelligent space," in *Proc. Int. Conf. IROS*, vol. 2, pp. 1358–1363.

[3] G. Appenzeller, J.-H. Lee, and H. Hashimoto, "Building topological maps by looking at people: An example of cooperation between intelligent spaces and robots," in *Proc. Intell. Robots Syst.*, 1997, vol. 3, pp. 1326–1333.

[4] J.-H. Lee, K. Morioka, N. Ando, and H. Hashimoto, "Cooperation of distributed intelligent sensors in intelligent environment," *IEEE/ASME Trans. Mechatronics*, vol. 9, no. 3, pp. 535–543, Sep. 2004.

[5] A. A. Tóth and A. R. Várkonyi-Kóczy, "A hand gesture controlled interface for intelligent space applications," in *Proc. 13th IEEE Int. Conf. INES*, Brigdetown, Barbados, Apr. 16–18, 2009, pp. 239–244.

[6] Q. Chen, M. D. Cordea, E. M. Petriu, A. R. Várkonyi-Kóczy, and T. E. Whalen, "Human-computer interaction for smart environment applications using hand gestures and facial expressions," *Int. J. Adv. Media Commun.*, vol. 3, no. 1/2, pp. 95–109, Jun. 2009.

[7] N. D. Binh, E. Shuichi, and T. Ejima, "Real-time hand tracking and gesture recognition system," in *Proc. Int. Conf. Graph., Vision Image Process.*, 2005, pp. 362–368.

[8] N. D. Binh and T. Ejima, "Hand gesture recognition using fuzzy neural network," in *Proc. ICGST Int. Conf. Graph., Vision Image Process.*, Cairo, Egypt, 2005, pp. 1–6.

[9] G. A. Carpenter, S. Grossberg, N. Markuzon, J. H. Reynold, and D. B. Rosen, "Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps," *IEEE Trans. Neural Netw.*, vol. 3, no. 5, pp. 698–713, Sep. 1992.

[10] B. Hussain and M. R. Kabuka, "A novel feature recognition neural network and its application to character recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 1, pp. 98–106, Jan. 1994.

[11] B. Tusor and A. R. Várkonyi-Kóczy, "Circular fuzzy neural network based hand gesture and posture modeling," in *Proc. IEEE I^2MTC*, Austin, TX, May 2010, pp. 815–820, [CD-ROM].

[12] G. R. Bradski, "Computer vision face tracking for use in a perceptual user interface," *Intel Technol. J.*, vol. Q2, pp. 1–15, 1998.

[13] CSC2503F Project Report S. Malik, Real-Time Hand Tracking and Finger Tracking for Interaction2003.

[14] A. R. Várkonyi-Kóczy, "Autonomous 3D model reconstruction and its intelligent applications in vehicle system dynamics: Theory and applications, Parts I-II," in *Proc. 5th IEEE Int. SISY*, Subotica, Serbia, Aug. 24–25, 2007, pp. 19–24.

[15] H. Ishibushi and H. Tanaka, "Fuzzy neural networks with fuzzy weights and fuzzy biases," in *Proc. IEEE Neural Netw. Conf.*, San Francisco, CA, 1993, vol. 3, pp. 1650–1655.

[16] G. Bradski and A. Kaehler, Intel OpenCV Library2009. [Online]. Available: http://opencv.willowgarage.com/wiki

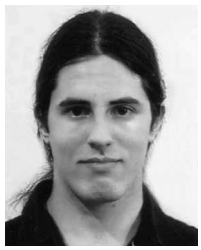
[17] J. Y. Bouguet, Camera Calibration Toolbox for Matlab2008. [Online]. Available: http://www.vision.caltech.edu/bouguetj/calib_doc

[18] R. S. Glanville, Anim8or2008. [Online]. Available: http://www.anim8or.com

Annamária R. Várkonyi-Kóczy (M'94–SM'97–F'07) was born in Budapest, Hungary, in 1957. She received the M.Sc. degree in electrical engineering, the M.Sc. degree in mechanical engineer-teacher, and the Ph.D. degree from the Technical University of Budapest, Budapest, in 1981, 1983, and 1996, respectively, and the D.Sc. degree from the Hungarian Academy of Sciences, Budapest, in 2010.

She was a Researcher with the Research Institute for Telecommunication, Budapest, for six years, followed by four years with the Group of Engineering Mechanics, Hungarian Academy of Sciences. From 1991 to 2009, she was with the Department of Measurement and Information Systems, Budapest University of Technology and Economics, Budapest. Since 2009, she has been a Full Professor with the Institute of Mechatronics and Vehicle Engineering, Óbuda University, Budapest. She is also founding professor and project leader with the Integrated Intelligent Systems Japanese-Hungarian Laboratory, Budapest. Her research interests include digital image and signal processing, uncertainty handling, soft computing, anytime, and hybrid techniques in complex measurement, diagnostics, and control systems.

Dr. Várkonyi-Kóczy was the past Vice-President of the Hungarian Fuzzy Association, is an elected member of the Hungarian Academy of Engineers, a member of the John von Neumann Computer Society and the Measurement and Automation Society (Hungary).



Balázs Tusor was born in Kerepestarcsa, Hungary, in 1987. He received the B.Sc. degree in computer engineering in 2010 from Budapest University of Technology and Economics, Budapest, Hungary, where he is currently working toward the M.Sc. degree (specialization on intelligent systems).

He is also a Student Researcher with the Integrated Intelligent Systems Japanese-Hungarian Laboratory, Budapest. His research interests include intelligent space and application of neural networks, fuzzy control, hybrid soft computing techniques, and artificial intelligence methods.