

Sign-language-enabled information kiosk

Oya Aran, Pavel Campr, Marek Hruz, Alexey Karpov, Pinar Santemiz, and Miloš Železný

Abstract—The abstract goes here.

Index Terms—IEEEtran, journal, L^AT_EX, paper, template.

I. INTRODUCTION

THE aim of this eINTERFACE 2008 project is to design an information kiosk for deaf people that will use sign language (SL) as a main communication mean. Background of the idea is based on current research on sign language processing carried out at UWB (University of West Bohemia) and BU (Boğaziçi University). Methods for synthesis (visual animation) and recognition of sign language are under research. The project also follows the work done during previous eINTERFACE workshops (project No. 3 *Sign Language Tutoring Tool* at eINTERFACE 2006 in Dubrovnik [1] and project No. 3 *A Multimodal Framework for the Communication of Disabled* at eINTERFACE 2007 in Istanbul).

Deaf or hearing-impaired users have limited possibility of communication with hearing people, which can be problem especially in the case of communication with authorities or information providers (train connection, sales, etc.) These people also cannot use speech-based automatic information services. In these cases dialogue systems should be designed to be accessible by deaf users to solve this problem. This project aims to design the dialogue system (information kiosk) in this way.

The idea is to combine sign language recognition and synthesis tasks to develop a simple information kiosk [2] for providing information such as train connections for deaf people who use sign language. The kiosk will be an alone-standing machine equipped with a standard PC, a touch screen display, a big screen for an avatar a microphone, and one or several cameras. The cameras will capture body/facial gestures of a signing person, while the big screen will present the rendered sign language output and the touch screen will allow touch commands as an alternative input. The system (kiosk) will wait for an input from user. When the user occurs in front of the kiosk, it will start its SL recognizer. It will decode the information user is looking for, detect important signs for obtaining data needed (time, destination, etc.) and requests missing data from user. As an output, information based on this data will be produced.

Pavel Campr, Marek Hruz and Miloš Železný are with the Department of Cybernetics, Faculty of Applied Sciences, University of West Bohemia, Pilsen, Czech Republic.

Oya Aran and Pinar Santemiz are with the Computer Engineering Department, Boğaziçi University, Istanbul, Turkey.

Alexey Karpov is with the Speech and Multimodal Interfaces Laboratory, St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences, St. Petersburg, Russia.

II. SYSTEM OVERVIEW

A. The information kiosk setup

Similarly as in case of speech-based information systems, we need to design a concept for sign language-based information system. For such system, special hardware will be required to be able to acquire input data. To be able to recognize sign language (both manual gestures and oral articulation) we need to collect both detailed (facial) and whole body visual data. It is putting restrictions to the recording conditions. We present design of hardware setup for such information system as an information kiosk allowing to be used in public places, such as train station and at the same time allowing to capture data allowing best possible recognition rates.

The information system can be divided into several basic parts: central control unit, input sensors, and output part. Input sensors comprise a visual part, an acoustic part, and a touch screen part. Output is presented to the user using graphical screen. Central control unit will be typically a multimedia PC equipped with special hardware for recording and processing of the input data. Central control unit will also take care of the connection to external data sources, such as various types of databases (about train connections etc.)

Visual input sensors are made up by cameras situated in the rear part of the kiosk. Camera 1 looks from horizontal view at the user (the upper part of the body) standing in front of the kiosk. Camera 2 is situated at the top of the rear part of the kiosk. It looks downwards at the user's whole body. It enables to gather 3D information about the position of hands. Camera 3 is situated near Camera 2, but looks at the user's face. Data from Camera 3 will be used for automatic lip-reading. Whole hardware setup is depicted in Figure 1.

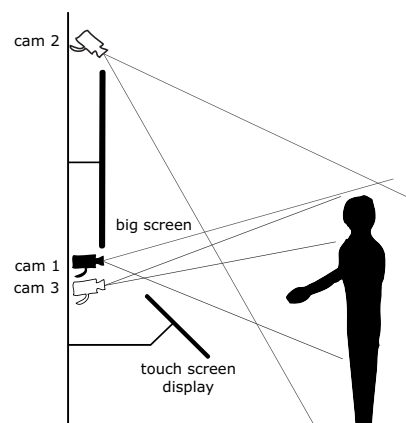


Fig. 1. Setup of cameras proposed for information kiosk for deaf users. This setup was used for recording SLR-A and SLR-B databases.

Acoustic input sensor will be a microphone situated at the top of the screen, closest to the user. It should capture voice of the user with highest possible sensitivity. Acoustic part is an optional part of the system and can be used in case when communication of hearing (and speaking) people with the system is expected. However in the case, when aurally people who can speak, will try to use voice part of the system, expected recognition rate of such speech will be very low and can be used only as an accompaniment.

Haptic input sensor will be realized as a touch screen. For some easy tasks such as selection from several choices it will be more efficient to present graphically several choices and let user select by touching the screen at the position of one of the choices instead of forcing him to use sign language, as the expression in sign language can be complicated (city names that has to be spelled etc.)

Graphical screen output will be used to guide the user through the whole information providing process, allowing him/her to quickly enter most frequently used modes, letting him to choose from several choices when appropriate, and giving him feedback for his/her actions. Design of the graphical user interface will be done with respect to the special needs of users and will be as intuitive as possible to efficiently help the user to reach his/her goal.

III. DIALOGUE CONTROL MODULE

A. Computer-driven dialogue

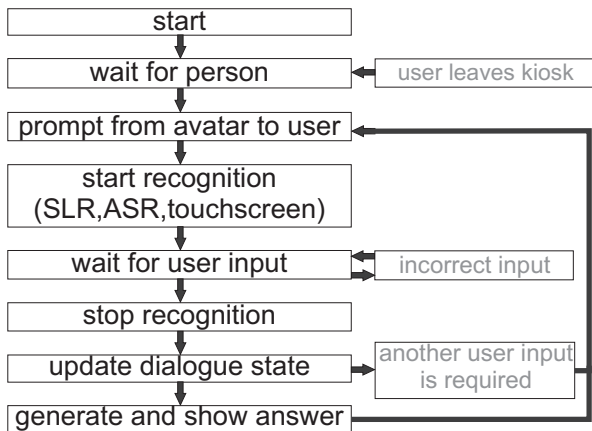


Fig. 2. Dialogue flow

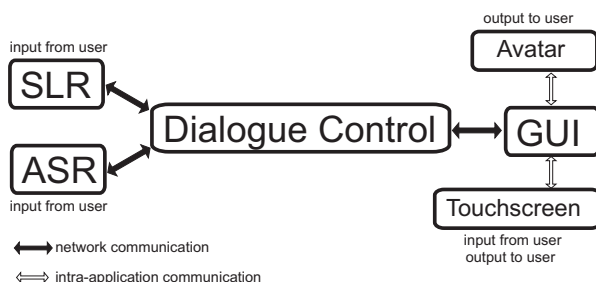


Fig. 3. Modules overview

The interaction between the user and the kiosk is managed by a computer-driven dialogue system (fig. 2). The dialogue consists of several scenarios, which are defined as a set of questions, their possible answers and answer generator. When the user selects a scenario (e.g. train connection information), the dialogue control asks the user for an answer of the first unanswered question. The answer can be entered as one- or multiple-word sentence (supported by speech recognizer, planned for SL recognizer in the future). All entered sentences must satisfy a context-free grammar defined by Backus-Naur Form (BNF). The grammar allows to generate a list of all sentences, their parts or words. All possible words, which can follow in the current sentence, are displayed in GUI (fig. 4, right). Later it allows to validate the user's answer. When all required questions are answered the dialogue system generates an answer for the user's query.

Scenarios are defined in one configuration file (YAML format) which can be easily modified. Here in an example with a part which define "departure information" scenario:

```

dialogue:
...
screens:
  layout:
    template: layout.html
  ...
departures:
  template: body_departures.html
  prompt: here you can find information about
         departures from this train station
  legend: departures
  form:
    departure:
      prompt: select departure
      label: departure
      type: grammar
      grammar: towns
      default: "town_prague."
    destination:
      prompt: select destination where you
             want to go
      label: destination
      type: grammar
      grammar: towns
    date:
      prompt: select date of departure
      label: date of departure
      type: grammar
      grammar: date
    time:
      prompt: select time of departure
      label: time of departure
      type: grammar
      grammar: time
  
```

The default language used in the system is English. Other languages are translated using internationalization configuration file which contains all necessary translations from English to other languages (Czech, Turkish, Russian) both spoken and signed (for sign synthesis).

IV. GRAPHICAL USER INTERFACE MODULE

Graphical user interface (GUI) consists of two screens. The first shows a signing avatar which guides the user through the dialogue by asking questions and presenting an answer when all questions are answered. To increase interactivity the avatar turns in the direction to the user which is achieved by face detection.

The second touchscreen shows current dialogue status. There is a list of both answered and unanswered questions.

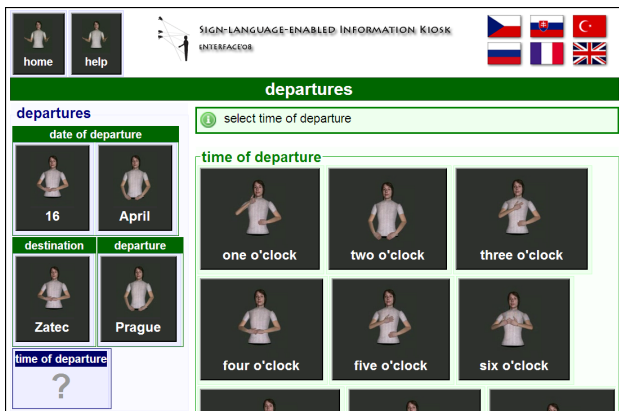


Fig. 4. Sample screen of touchscreen graphical user interface. Gray buttons and flags are clickable. Top: main menu. Left: dialogue status with answered and unanswered questions. Right: current question and all possible answers which can be selected by haptic, speech or sign-language modalities.

By clicking on an answered question this answer can be reset. This is an advantage of using graphical representation of information which is able to present more information at one time in compare to speech dialogue systems. Next part of the screen is used to present the list of all possible answers for current question. If the answers are multiple-word sentences then the list contains only words on the first places in the sentences and next words are listed later after the first is selected.

The first screen with signing avatar is rendered online and is described in section "Sign language synthesis". The second screen contains graphical user interface which is generated as a XHTML document. Particular screens are generated from XHTML templates which are filled by data generated by the dialogue control module. The signing avatars displayed in the buttons are pre-generated and showed in the XHTML document as flash animations.

V. SIGN LANGUAGE RECOGNITION MODULE

A. Database

The sign language recognition module is intended for recognizing isolated signs. For this purpose a database was created using an application `sign_capture` developed at this workshop. This application renders a sign that should be performed by the user on the screen. This approach was chosen since the users were not familiar with Signed Czech Language. After the user is ready to perform the sign the system automatically detects the movement and starts recording from the camera to a video file. After the hands are at the body and no movement is detected the capturing stops and the file is saved. In total 338 files were recorded with one male and one female signer. The database contains 50 signs from Czech signed language such as Czech towns, days, and miscellaneous signs (i.e. yes, no, information). Every sign was repeated three times by the user. While recording the conditions were long sleeves, non-skin-colored clothes, uniform background, and constant illumination.

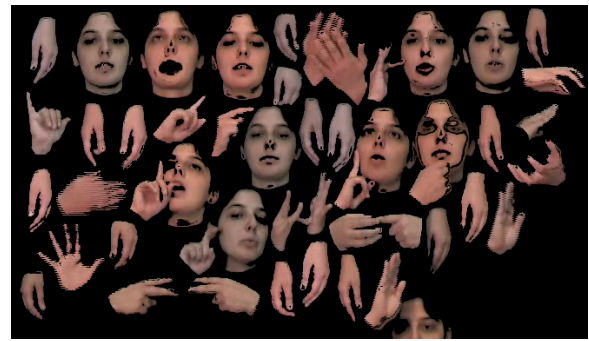


Fig. 5. Skin-color examples of one signer.

B. Skin Color Segmentation

Skin color is widely used to aid segmentation in finding parts of the human body [3]. We learn skin colors from a training set and then create a model of them using a Gaussian Mixture Model (GMM). This is not a universal skin color model; but rather, a model of skin colors contained in our training set. We prepared the set of training data by extracting images from our input video sequences and manually selected the skin colored pixels. We used images of both speakers under slightly different lighting conditions. In total, we processed 50 video segments. An example of training images is shown in Fig 5.

For color representation we use the RGB color space. The main reason is that this color space is native for computer vision and therefore does not need any conversion to other color space. The collected data are processed by the Expectation Maximization (EM) algorithm to train the GMM. After inspecting the spatial parameters of the data we decided to use a five Gaussian mixtures model. The straight forward way of using the GMM for segmentation is to compute the probability of belonging to a skin segment for every pixel in the image. One can then use a threshold to decide whether the pixel color is skin or not. But this computation would take a long time, provided the information we have is the mean, variance and gain of each Gaussian in the mixture. We have precomputed the likelihoods and stored them in a look-up table. The range of the likelihood is from 0 to 255. A likelihood of 128 and more means that the particular color belongs to a skin segment. With this procedure we obtain a $256 \times 256 \times 256$ look-up table containing the likelihood of all the colors from RGB color space. The segmentation is straightforward. We obtain the likelihood of the color of each pixel from the look-up table.

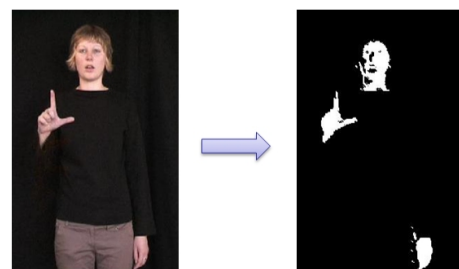


Fig. 6. Skin color segmentation example

According to the likelihood of the color, we decide whether it belongs to a skin segment or not. For each frame, we create a mask of skin color segments by thresholding the likelihood image. The result is as shown in Fig 6.

C. Tracking

1) *The Joint PF* : We use a joint PF that calculates a combined likelihood for all objects by modeling the likelihood of each object with respect to others. Figure 7 shows the pseudo-code of the joint PF. \mathbf{x}_t^n is the joint object state, as defined in Eq.1. The first step is to determine the initial states and the weights of the particles, with respect to the prior distribution. The initial distribution of the particles are determined with respect to an explicit detection step based on connected component labeling. The particles at time t are determined by the re-sampling, prediction and weight setting steps. At the re-sampling step, new particles are sampled with replacement from the weighted particles at time $t - 1$. At this step the weights of the new particles are equally assigned. The states of the re-sampled particles at time t are determined by the object dynamics and by an additional mean-shift step. The weights are determined by normalizing the joint likelihood.

- 1) Initialization: $\{(\mathbf{x}_0^n, \pi_0^n)\}_{n=1}^N$
 - 2) For $t > 0$
 - a) Re-sampling:
 $\{(\mathbf{x}_{t-1}^n, \pi_{t-1}^n)\} \rightarrow \{(\mathbf{x}_{t-1}'^n, 1/N)\}$
 - b) Prediction: $\mathbf{x}_t''^n = f(\mathbf{x}_{t-1}'^n)$
 $\{(\mathbf{x}_{t-1}'^n, 1/N)\} \rightarrow \{(\mathbf{x}_t''^n, 1/N)\}$
 - c) Mean-shift iterations: $\mathbf{x}_t^n = MS(\mathbf{x}_t''^n)$
 $\{(\mathbf{x}_t''^n, 1/N)\} \rightarrow \{(\mathbf{x}_t^n, 1/N)\}$
 - d) Weight setting: $\pi_t^n \propto z_t^n = h(\mathbf{x}_t^n)$
 $\{(\mathbf{x}_t^n, 1/N)\} \rightarrow \{(\mathbf{x}_t^n, \pi_t^n)\}, \sum_{n=1}^N \pi_t^n = 1$
 - e) Estimation: $\hat{\mathbf{x}}_t = E[\{(\mathbf{x}_t^n, \pi_t^n)\}]$

Fig. 7. Joint PF algorithm

2) *Object Description*: The state vector for a single object consists of the position, the velocity and the shape parameters. The shape parameters are selected as the width, the height and the angle of an ellipse surrounding the object. Thus, for a single object we have a seven dimensional state vector. Then, the joint particle is a single vector containing all the objects in the scene:

$$\mathbf{x}_t^n = \{\mathbf{x}_t^{n,f}, \mathbf{x}_t^{n,r}, \mathbf{x}_t^{n,l}\}^T \quad (1)$$

where f, r, l are indexes to the face, right and left hands. In the rest of the paper, we will refer to \mathbf{x}_t^n as the joint particle and $\mathbf{x}_t^{n,i}$ as the particle or as the sub-particle, alternatively.

3) *Dynamic Model*: For each object, the position and the velocity parameters are modeled by a damped velocity model and the shape parameters are modeled by a random walk model.

For multiple objects in the joint PF, the dynamic model is applied to each object. We additionally apply mean shift [4] to the sub-particles of each object independently. The MS algorithm moves the particle centers to the areas with high skin color probability. This allows us to use particles effectively, since the particles with low weights will be less likely. As a result, a PF with MS needs fewer particles than a standard PF.

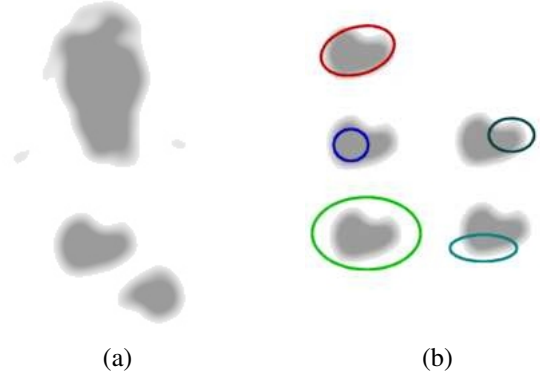


Fig. 8. (a) The thresholded image that is used in likelihood calculation, (b) hand region and different particles. The likelihood function gives the highest likelihood for the particle at the top.

4) *Appearance Model*: We use the skin color probability image, which has a positive probability for skin color pixels and zero probability for other colors.

To calculate the likelihood of a single object, we make two measurements based on the ellipse that is defined by the state vector of the particle:

- A : The ratio of the skin color pixels to the total number of pixels inside the ellipse.
- B : The ratio of the skin color pixels to the total number of pixels at the ellipse boundary.

These two ratios are considered jointly in order to make sure that our measurement function gives high likelihood to particles that contains the whole hand without containing many non-hand pixels [5]. If we do not take the ellipse boundary into account, smaller ellipses are favored and particles tend to get smaller. We design our measurement function Eq.2 to return a high likelihood when A is as high as possible and B is as low as possible:

$$z_t^{n,i} = \begin{cases} 0 & , \text{if } A < \Phi_p \\ 0.5 \cdot A + 0.5 \cdot (1 - B) & , \text{otherwise} \end{cases} \quad (2)$$

where $z_t^{n,i}$ denotes the likelihood of a single object, i , for the particle n .

The first line in Eq.2 is required to assign low likelihood to particles that have zero or very few skin color pixels. Otherwise these particles will receive 0.5 likelihood value even if they do not contain any skin color pixels. The equation takes its highest value when there are no skin colored pixels at the boundary ($B = 0$), and when all the inner pixels are skin colored. Figure 8b shows the grey-level hand image and possible particles. The particle at the top receives the highest likelihood by Eq.2 where as the other particles receive lower likelihoods.

5) *Joint Likelihood Calculation*: A joint particle is a combination of sub-particles which refer to the objects we want to track, i.e. two hands and the face (Eq.1). We calculate the joint likelihood with respect to the following:

- 1) The likelihood of a single object based on the appearance model;

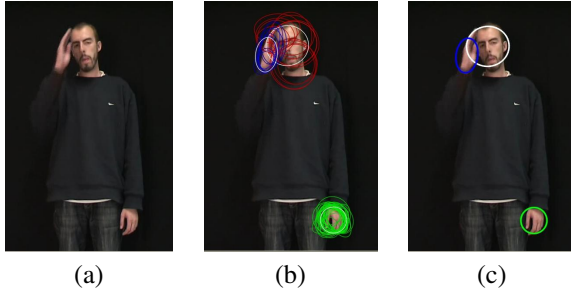


Fig. 9. (a) Original image, (b) Particle distribution with joint PF, (c) Estimated hand positions

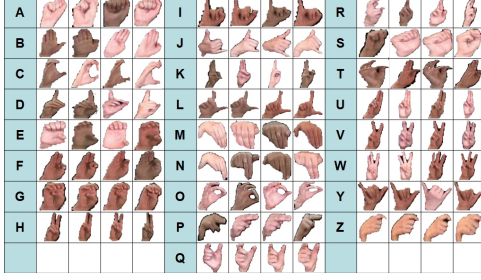


Fig. 10. Finger alphabet of the Czech sign language

- 2) The distance of each object to the other objects to handle interactions. Assign low likelihood if the sub-particles are close to each other;
- 3) Additional constraints on respective object locations. This criterion is needed to prevent wrong object assignments especially after occlusions.

We define partial likelihoods for each criterion above and calculate the joint likelihood by the multiplication of the partial likelihoods.

The joint likelihood is calculated for the objects that stay in the scene. If any of the objects disappears, it is excluded from the likelihood calculations. We assume an object has disappeared if all of the sub-particles of that object have zero weight.

D. Feature Extraction

In order to recognize the signs, other than the tracking features we also need shape information. In order to describe the shape, we implemented five algorithms and tested their performances over the set of finger alphabet in Czech Sign Language, which can be seen in Fig 10.

After segmenting and tracking of the hands, we obtain the segmented hand image for each sign. Using this image, we find the contour and the gray scale image of the hand. Then we find DCT coefficients from the gray scale image [6], Hu moments from the segmented hand image [7], and Fourier descriptors from the contour points of the hand shape [8]. The feature extraction methods can be seen in Fig 11.

From the points on the contour, we get the following:

- Complex coordinates
- Distances of the points from the centroid of the hand
- Angles between two consecutive points

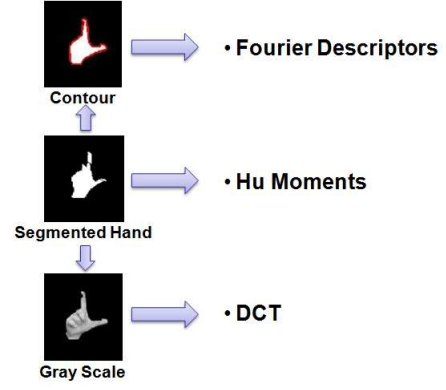


Fig. 11. Feature extraction methods

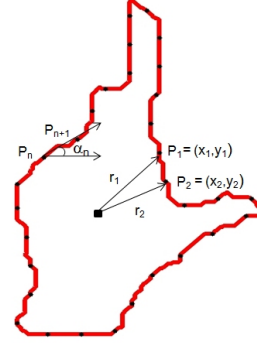


Fig. 12. The procedure to obtain the complex coordinates, central distances and the angle between consecutive points

Then we calculate Fourier descriptors using these features. The procedure of obtaining Fourier descriptor features can be seen in Fig 12.

In the first method, to eliminate the effect of translation, we first subtract from each coordinate the center, compute the Fourier transform of the complex coordinate and find its absolute value. So, if the point is $P_i = (x_i, y_i)$ and the center of the hand is $P_c = (x_c, y_c)$, then the first feature is

$$f_i^1 = \text{abs} \left[\text{fft} \left((x_i - x_c) + i(y_i - y_c) \right) \right]$$

In the second method, we find the distance of the boundary points from the centroid of the shape. So,

$$f_i^2 = \text{abs} \left[\text{fft} \left(\sqrt{(x_i - x_c)^2 + (y_i - y_c)^2} \right) \right]$$

In the third method, we first compute the angles between two consecutive points and compute the fft features using the formula

$$f_i^3 = \text{abs} \left[\text{fft} \left(\arctan \frac{y_{i+1} - y_i}{x_{i+1} - x_i} \right) \right]$$

In the fourth method, we compute the DCT features from the gray scale image of the hand [6]. Then we trace the upper-left corner of the DCT image diagonally and choose the first 15 features and eliminate the first feature, which is the DC component.

Finally in the last method, we compute the Hu moments from the segmented hand image [7]. In order to find the Hu

Method	Accuracy	Std Deviation
FFT-Complex	22%	2,05%
FFT-Centroid	23%	2,51%
FFT-Angle	17%	3,70%
DCT	75%	3,92%
Hu Moments	30%	2,54%

TABLE I
RECOGNITION RESULTS USING THE SET OF CZECH SIGN LANGUAGE
FINGER ALPHABET

moments, we first compute the internal moments of order $p+q$ using the formula

$$m_{pq} = \iint (x - x_c)^p (y - y_c)^q dx dy$$

where x, y are the pixel coordinates belonging to the hand. Then we normalize the moments for eliminating the scale factor

$$n_{pq} = \frac{m_{pq}}{m_{00}^{\frac{p+q}{2}+1}}$$

We use seven Hu moments which are invariant to rotation. The first six are also reflection invariant whereas the seventh moment is skew orthogonal invariant, which is useful in distinguishing mirror images.

$$\begin{aligned} S_1 &= n_{20} + n_{02} \\ S_2 &= (n_{20} + n_{02})^2 + 4n_{11}^2 \\ S_3 &= (n_{30} - 3n_{12})^2 + (n_{03} - 3n_{21})^2 \\ S_4 &= (n_{30} + n_{12})^2 + (n_{03} + n_{21})^2 \\ S_5 &= (n_{30} - 3n_{12}) \cdot (n_{30} + n_{12}) \cdot \\ &\quad ((n_{30} + n_{12})^2 - 3(n_{03} + n_{21})^2) \\ &\quad - (n_{03} - 3n_{21}) \cdot (n_{03} + n_{21}) \\ &\quad \cdot (3(n_{30} + n_{12})^2 - (n_{03} + n_{21})^2) \\ S_6 &= (n_{20} - n_{02}) \cdot ((n_{30} + n_{12})^2 - (n_{03} + n_{21})^2) \\ &\quad + 4n_{11}^2 \cdot (n_{30} + n_{12}) \cdot (n_{03} + n_{21}) \\ S_7 &= (3n_{21} - n_{03}) \cdot (n_{30} + n_{12}) \cdot \\ &\quad ((n_{30} + n_{12})^2 - 3(n_{03} + n_{21})^2) \\ &\quad - (n_{30} - 3n_{12}) \cdot (n_{03} + n_{21}) \\ &\quad \cdot (3(n_{30} + n_{12})^2 - (n_{03} + n_{21})^2) \end{aligned}$$

In our experiment, we use a database containing 241 images from the Czech sign language finger alphabet. In this database there are 23 classes which are some of the letters in the alphabet. Each class contains six to 18 samples. To create our training set, we select five samples from each class randomly and put the remaining samples to the test set. We form six folds in total, where we guarantee that each sample is put at least once to the training set. So, in each fold training set includes 115 samples and test set includes 126 samples. We use 1-nearest neighbor method as a classifier. The recognition results can be seen in Table V-D.

According to our results, DCT obtained significantly better results than the others. The reason for that is DCT also takes into account the gray scale texture information in the hand region whereas the others only use shape information.

Some signs have a similar shape, so using only shape information it is difficult to differentiate between them. Some examples can be seen in Fig 13.

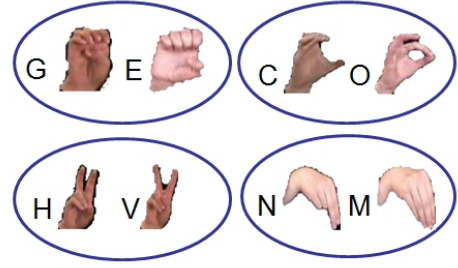


Fig. 13. Examples of images that are misclassified by the DCT method

E. Sign Recognition

For the purpose of recognition we use Hidden Markov Model (HMM). The signs are modeled as an 8-state HMM (two of these states are non-emitting). Each state is modeled as one Gaussian. This was due to the relative low amount of data. The structure of the HMM can be seen in Fig. [?].

Fig. 14. The structure of HMM used for modeling the signs.

VI. SIGN LANGUAGE SYNTHESIS MODULE

A. Feedback Animation

1) *Animation Model*: The feedback animation module can be divided into a module for a rendering of the animation model and to a module for a forming of the animation trajectories (a trajectory generator). 3D geometric animation model of the avatar is in compliance with the H-Anim standard. Currently the animation model covers 38 joints and body segments. Each segment is represented as textured triangular surface, 16 segments is used for fingers and the palm, one for the arm and one for the forearm, totally 1372 vertices and 1772 triangles per hand. The thorax and the stomach are together represented by one segment, 182 vertices and 360 triangles. The talking head is composed from seven segments, totally 4692 vertices and 9243 triangles.

The body segments are connected by the avatar skeleton. One joint per segment is sufficient for this purpose. A controlling of the skeleton is carried out through the rotation of segments (3 DOF per joint). The rotation of the shoulder, elbow, and wrist joints are completed from 3D positions of the wrist joints by the inverse kinematics.

The animation of the talking head is performed by a local deformation of the triangular surfaces. It is primarily used for the animation of the avatar's face and the tongue [9]. The triangular surfaces are deformed according to influence zones defined on the triangular surface by spline functions constructed from several 3D control points. The collection of these points is currently taken by 9 animation parameters. The rendering of the animation model is implemented in C++ and OpenGL code.

2) *Trajectory Generator*: Firstly for the manual component of signed speech, the trajectory generator performs the syntactic analysis of the symbolic string on the input HamNoSys string and creates parse tree structure. Currently the trajectory

generator uses 374 parse rules. However, it is difficult to define rules and actions for all symbol combinations to cover the entire notation variability. We made a few restrictions in order to preserve maximum degree of freedom. In this assumption, the annotation of the sign have a good meaning for the user familiar with HamNoSys as well as signs are obvious enough for the transformation to the avatar animation. Next, the structurally correct string is decomposed to nodes determined by parsing rules. Two key frames data structure to distinguish the dominant and non dominant hand are used. The data structure of the key frame is composed from items specially designed for trajectory generation purpose (Figure 15 on the left). Next step of trajectory forming is filling of the leaf nodes from the symbol descriptors stored in the definition file (Figure 15 on the right). The definition file covers 138 HamNoSys symbols.

		<i>hamsym.dat</i>
Location	Pointer segment of body	HAMSYM \times
	Index of pointer segment	fingor 180.0 90.0 0.0
	Location segment	HAMSYM \circ
	Array of location index	fingor 0.0 0.0 45.0
	Index to array of location	
Action	Distance from location segment	HAMSYM \equiv
	Relative change (x,y,z)	locsegname hanim_15
	Type of the motion	idxloc 121 398 21 123
	Size of amplitude	whichidloc 2
	Amplitude gain	distance 0.4
Orientation	Turning of motion amplitude	HAMSYM \sim
	Angles for circle sector	typemov zigzag
Handshape	Orientation of wrist	turn 0.0
	Handshape vector	amplit 1.0
	Finger flexion	HAMSYM \pm
	Mask for finger selection	change 0.2 0.0 0.0
	Thumb shape	

Fig. 15. Left panel: The list of all items. Right panel: An example of the items stored in the definition file.

39 rule actions were added in manner that one rule action is connected with each parse rule. The parse tree is processed by several tree walks. The initial tree walks put together the items of the key frames according to the type of the rule actions. The reduced tree is joined and transformed to the trajectories accordance with the timing of the particular nodes. Finally, the final trajectories for both hands are contained in the root node. The number used symbols, parse rules, and actions used in the system is summarized in Figure 16. The final step is the conversion of the trajectories into the avatar animation and synchronization with articulatory trajectories generated by the selection of articulatory targets [9].

3) *Evaluation of Synthesized Sign Speech by Deaf*: A subject evaluation of quality of synthesized sign speech has been performed with the manual component and the non-manual component. The manual and non-manual component has been generated by system described in previous subsections. The perception test by deaf children from primary school is scored on the isolated signs. Two experiments either is composed from two tests were designed. The second test followed the first one after three weeks with identical procedure. Five deaf pupils were chosen from the preliminary class and the first class (5-6 years) as participants for the first experiment and six deaf pupils from the sixth and seventh class (11-13 years)

Block	HamNoSys Symbols		Parser	
	#Base	#Auxiliary	#Rules	#Actions
Symmetry	4	0	8	8
Handshape	12	11	35	6
Finger and palm orientation	26	3	34	5
Location	30	17	85	14
Action	40	42	204	15
Link of blocks	0	0	8	1

Fig. 16. The statistic of symbols, rules, and actions used by the HamNoSys parser.

for the second experiment.

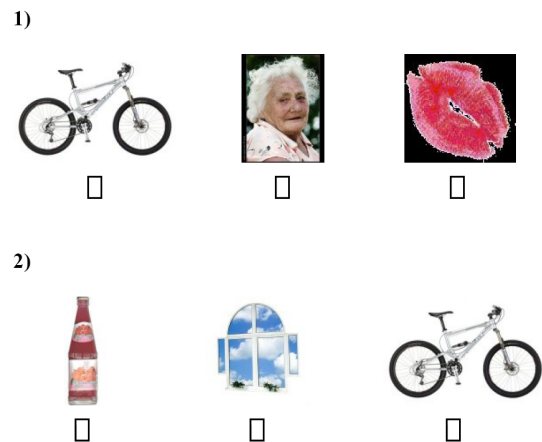


Fig. 17. An example of two choices from the sheet used for filling answers in multiple-choice test. Three choices are offered in form of three illustrations.

Test material contains the synthesized animation of isolated sign recorded to the video files. 15 signs from videotapes used in the curriculum of the preliminary class were collected, thus the signs would be know all participated pupils. The sign editor has been used for HamNoSys notation of the sings. Next, we have prepared sheets for the multiple-choice test with three response options (one correct response) composed from randomly arranged pictures of the tested words, Figure ??.

Procedure The presentation of the records consists in a sequential projection of the tested signs on the wall in the classroom by the data projector. At the beginning of the experiments, five extra non-scored words were presented to demonstrate various options of the study. The picture of signed character on the wall was approximately 30 cm high. The pupils were not familiar with the tested words before the experiments and the scribing was prevented. The procedure of first and second experiment was the same. The only difference was in the second experiment when older pupils did not use the sheets of multiple-choice test. This step was taken because these students already achieved in the first test good results. Therefore they recognize the signs without a choice.

Results Tests have been scored as follows. Pupil got for the correct answer one point and for the wrong answer none point. The similarity of some signs was not taken into account. We tested the hypothesis that pupils filled the multiple-choice test by a chance. For this purpose, we have used one-sample and one-sided t-test. The planned comparisons are carried out for

both tests of the first experiment and for the first test of the second experiment, ($\alpha = 0.01$). The results show significantly better understanding of the signed speech than a chance (three options, the chance level 33.3%, $p < 0.01$). The average 80% success was achieved in the first test of first experiment ($t(4) = 6.6$, $p = 0.0014$) and 85% for the second test ($t(3) = 17.91$, $p < 0.001$). Better results are achieved in the second experiment with older pupils. There are for the first test on average 95% correct answers ($t(5) = 27.59$, $p < 0.001$) and the retesting without the possibility of choosing, on average 80% correct answers. The means of achieved scores are summarized in Table. ??.

TABLE II
SIGN LANGUAGE SYNTHESIS UNDERSTANDING EXPERIMENTS

Experiment	1	2
Test 1	80%	95%
Test 2	85%	80%

VII. AUTOMATIC SPEECH RECOGNITION MODULE

A speaker-dependent automatic speech recognition (ASR) system was developed and embedded into the information kiosk as an optional alternative to the interactive GUI and the interface based on sign language recognition. ASR system is multilingual one and able to recognize voice commands both in English and Czech. The lexicon of ASR contains 101 diverse words for each language (Czech towns, digits, names of months and weekdays, etc.)

A. Signal processing and feature extraction

The audio signal is captured by a microphone of a headset and sampled at 16 KHz with 16 bits on each sample using a linear scale. The system is intended for the distant talking and human-computer interaction, so the microphone was selected to be able to capture speech signal at the distance of 2 meters from a speaker with an acceptable SNR. The signal is divided into the frames and cepstral coefficients are computed for the 25 ms overlapping frames with 10 ms shift between adjacent frames applying the bank of triangular filters calculated according to the Mel-scale frequencies 3:

$$Mel(f) = 2595 \log_{10} 1 + \frac{f}{700} \quad (3)$$

Mel-frequency cepstral coefficients (MFCCs) are calculated from the log filterbank amplitudes using the discrete cosine transform. So the audio speech recognizer system calculates 13 MFCCs (including 0-th coefficient) as well as estimates the first and second order derivatives that forms an observation vector of 39 components. The acoustical modeling is based on left-right continuous Hidden Markov Models (HMMs) [10], applying mixtures of Gaussian probability density functions that are defined according to the equation 4:

$$\mathcal{N}(o; \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^n |\Sigma|}} e^{-\frac{1}{2}(o-\mu)^T \Sigma^{-1} (o-\mu)} \quad (4)$$

where \mathcal{N} is a Gaussian with mean vector μ and covariance matrix Σ , and n is the dimensionality of the observable

vector o . HMMs of phonemes have three meaningful states and two “hollow” states intended for concatenation of the models of phonemes in the models of words. Each word of the vocabulary is obtained by concatenation of context-independent phonemes (triphones). As the base technology for realization of speech recognizer Hidden Markov Models Toolkit (HTK) [11], developed by the Cambridge University Engineering Department, was used. It is a free available toolkit including C source codes. HTK 3.4 toolkit is employed on all the levels of audio signal processing. Modeling of speech by HTK includes two main stages:

- Training HMMs of acoustical items using a phonetically labeled speech corpus.
- Speech recognition in on-line or off-line modes.

B. System’s training and evaluation

In order to train the speech recognizer a speech corpus was recorded in office conditions using the distant talking directed microphone. About 1000 utterances of two users were recorded and used for training HMMs of phonemes. Totally we have recorded above 10 minutes of speech data from each speaker, these data were labeled semi-automatically in the terms of phoneme sets. It is required to notice that the phonemic alphabets, for instance SAMPA alphabets [12] are rather different for English and Czech, the latter contains more phonemes. Totally 41 different phonemes are used in transcriptions of the lexicon, some words of which have several variants of pronunciations taking into account peculiarities of the speakers. 20 % training utterances were manually labeled by the WaveSurfer software, the rest of the data were automatically segmented by the Viterbi forced alignment method with the flat start [11]. In general the stage of acoustical models training includes the following steps:

- Manual transcription of a lexicon of an applied domain
- Creation of a grammar or a statistical language model (for instance, bi-gram or tri-gram model)
- Preparation of a training speech corpus
- Coding the speech data (feature extraction)
- Definition of topology of HMMs (prototypes)
- Creation of initial HMMs by flat start
- Re-estimation of HMMs parameters of monophones using a labeled speech corpus and Baum-Welch algorithm
- Re-estimation of HMMs of triphones by Baum-Welch algorithm
- Mixture splitting

The speech decoder uses Viterbi-based token passing algorithm [11]. The input phrase syntax is described in a simple grammar that allows to recognize one command in a hypothesis. The audio speech recognizer operates quite fast (less than 0.5xRT) so the result of speech recognition is available almost immediately after detection of speech end by an energy-based voice activity detector. The performance of ASR was evaluated by another speech data, collected in the same office conditions as the training part. Totally XXX phrases were pronounced by two male speakers, the word recognition rate (WRR) was XXX %. This rate is acceptable for our task since ordinary single microphone is used for

distant speech capturing providing quite low SNR. It was found that SNR for audio signal is under 20 db because of far position (about 2 meters) of the speaker in front of the kiosk and the microphone. In further research a microphone array and corresponding digital signal processing methods for speaker localization and noise elimination are supposed to be applied [13].

VIII. EVALUATION OF THE SYSTEM

Experiments, statistics...

Only features from tracking (ellipse: x, y, width, height, angle, velocity.x, velocity.y). HMM has 8 states (two of them are non-emitting), 30 Gaussian mixtures per state, one training iteration.

No. parameters	Method	Recognition rate
21	x	33.68%
21	PCA	42.86%
15	PCA	28.57%
12	PCA	34.69%
9	PCA	33.67%
6	PCA	37.76%
4	PCA	23.47%
21	PCA & ICA	8.16%
15	PCA & ICA	7.14%
12	PCA & ICA	16.33%
9	PCA & ICA	31.63%
6	PCA & ICA	39.80%
4	PCA & ICA	16.33%

Features from tracking (ellipse: x, y, width, height, angle, velocity.x, velocity.y) and DCT coefficients of the hands. HMM has 8 states (two of them are non-emitting), one Gaussian per state, three training iteration. Tracking failed with 15 files.

No. parameters	Method	Recognition rate
49	x	81.63%
49	PCA	78.57%
39	PCA	78.57%
29	PCA	76.53%
19	PCA	70.41%
9	PCA	47.96%

IX. CONCLUSION

Because the kiosk should be usable by all hearing-impaired people (even who cannot read) we designed the GUI in a way that all important text labels are accompanied by an animation of signing avatar with corresponding sign. This new graphical user interface component can be used as a clickable button. This component and whole GUI layout is designed as a XHTML web page, so that the whole application could be used online, but without ASR and SLR components, and controlled only by a mouse.

The biggest usability problem of the kiosk design is the SL recognition where the user have to start and finish the performed sign in the initial position. This expectation must be explained to the user in the early beginning of the dialogue.

The other parts of the kiosk proved that they can be used without any major usability difficulties. This is achieved by using similar concept of the dialogue as is used in web page browsing which is well-known and the users don't have to think how to control the kiosk.

ACKNOWLEDGMENT

This project is supported by The Scientific and Technological Research Council of Turkey (TUBITAK) project 107E021, Bogazici University project BAP-03S106.

This research was supported by the Grant Agency of Academy of Sciences of the Czech Republic, project No. 1ET101470416 and by the Ministry of Education of the Czech Republic, project No. ME08106.

We would like to express our great thanks to Zdeněk Krňoul (University of West Bohemia, Pilsen) for contribution to this project by providing 3D avatar for sign language synthesis.

REFERENCES

- [1] O. Aran, I. Ari, A. Benoit, A. H. Carrillo, F.-X. Fanard, P. Campr, L. Akarun, A. Caplier, M. Rombaut, and B. Sankur, "Sign Language Tutoring Tool," *Proceedings of eINTERFACE 2006, Summer Workshop on Multimodal Interfaces, Dubrovnik, Croatia, 2006*.
- [2] M. Železný, P. Campr, Z. Krňoul, and M. Hruz, "Design of a multimodal information kiosk for aurally handicapped people," in *SPECOM 2007 proceedings, Moscow, Russia, 2005*, pp. 751–755.
- [3] V. S. V. Vezhnevets and A. Andreeva, "A survey on pixel-based skin color detection techniques," in *Graphicon, 2003*, pp. 85–92.
- [4] E. Maggio and A. Cavallaro, "Hybrid particle filter and mean shift tracker with adaptive transition model," in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2005*.
- [5] O. Aran and L. Akarun, "A particle filter based algorithm for robust tracking of hands and face under occlusion," in *IEEE 16th Signal Processing and Communications Applications (SIU 2008), 2008*.
- [6] R. C. Gonzales and R. E. Woods, *Digital Image Processing*. Prentice-Hall, 2001.
- [7] M. Hu, "Visual pattern recognition by moment invariants," *IRE Trans. Inf. Theory*, vol. 8, p. 179–187, 1962.
- [8] D. S. Zhang and G. Lu, "A comparative study of three region shape descriptors," in *Proc. of the Sixth Digital Image Computing Techniques and Applications (DICTA02), 2002*, pp. 86–91.
- [9] Z. Krňoul and M. Železný, "A development of Czech talking head," in *Proceedings of ICSPL 2008*, in press, 2008.
- [10] R. Rabiner and B. Juang, "Fundamentals of speech recognition," in *New Jersey: Prentice-Hall, Englewood Cliffs, USA, 1993*.
- [11] S. Y. et al., "The htk book," in *HTK Version 3.4, Cambridge University Engineering Department, 2006*.
- [12] "Sampa alphabet, <http://www.phon.ucl.ac.uk/home/sampa/>."
- [13] S. Brandstein and D. Ward, "Microphone arrays," in *Springer Verlag, 2000*.



Oya Aran Oya Aran received the BS and MS degrees in Computer Engineering from Boğaziçi University, Istanbul, Turkey in 2000 and 2002, respectively. She is currently a PhD candidate at Boğaziçi University working on dynamic hand gesture and sign language recognition under the supervision of Prof. Lale Akarun. Her research interests include computer vision, pattern recognition and machine learning. She is a student member of the IEEE.



Pavel Campr Pavel Campr was born in 1981 in the Czech Republic. He graduated in cybernetics from the University of West Bohemia (UWB) in 2005. As a Ph.D. candidate at the Department of Cybernetics, UWB, his research interests focus on hand gesture and sign language recognition, computer vision, machine learning and multimodal human-computer interaction. He is participating in the research project MUSSLAP. He is also teaching assistant and maintainer of the departmental website.

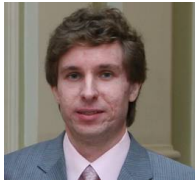


Miloš Železný was born in Plzeň, Czech Republic, in 1971. He received his Ing. (=M.S.) and Ph.D. degrees in Cybernetics from the University of West Bohemia, Plzeň, Czech Republic (UWB) in 1994 and in 2002 respectively.

He is currently a lecturer at the UWB. He has been delivering lectures on Digital Image Processing, Structural Pattern Recognition and Remote Sensing since 1996 at UWB. He is working in projects on multi-modal speech interfaces (audio-visual speech, gestures, emotions, sign language). He is a member of ISCA, AVISA, and CPRS societies. He is a reviewer of the INTERSPEECH conference series.

PLACE
PHOTO
HERE

Marek Hružík ...was born...



Alexey Karpov Alexey A. Karpov received the M.S. Diploma from St. Petersburg State University of Airspace Instrumentation and Ph.D. degree in computer science from St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS), in 2002 and 2007, respectively. His main research interests include automatic speech and speaker recognition, text-to-speech, multimodal interfaces based on speech and gestures, audio-visual speech processing. Currently he is a senior researcher of Speech and Multimodal Interfaces Laboratory of SPIIRAS.

He has been the (co)author of more than 60 papers in refereed journals and International conferences, for instance, Interspeech, Eusipco, TSD, etc. His main research results are published by the Journal of Multimodal User Interfaces and by the Pattern Recognition and Image Analysis (Springer). He is a coauthor of the book "Speech and Multimodal Interfaces" (2006, in Russian), and a chapter in the book "Multimodal User Interfaces: From Signals to Interaction" (2008, Springer). He has also been involved in EU SIMILAR Network of Excellence as well as several research projects funded by EU INTAS association and Russian scientific foundations. He was a winner of the 2-nd Low Cost Multimodal Interfaces Software (Loco Mummy) Contest (2006, Brussels). Dr. Karpov is a member of organizing committee of series of International conferences "Speech and Computer" SPECOM, as well as member of the EURASIP, ISCA and OpenInterface associations.



Pınar Santemiz Pınar Santemiz received the B.S. degree in mathematics from Boğaziçi University, Istanbul, Turkey, in 2006. She is currently an M.Sc. student in the Department of Computer Engineering, Boğaziçi University. Her research interests are in the areas of computer vision, sign language analysis, machine learning, and pattern recognition.