

# Intelligent video and audio applications for learning enhancement

Andrzej Czyzewski · Bożena Kostek

Received: 1 April 2011 / Revised: 30 May 2011 / Accepted: 7 June 2011 /

Published online: 1 July 2011

© The Author(s) 2011. This article is published with open access at Springerlink.com

**Abstract** The role of computers in school education is briefly discussed. Multimodal interfaces development history is shortly reviewed. Examples of applications of multimodal interfaces for learners with special educational needs are presented, including interactive electronic whiteboard based on video image analysis, application for controlling computers with facial expression and speech stretching audio interface representing audio modality. Intelligent and adaptive algorithms applications to the developed multimodal interfaces are discussed.

**Keywords** Multimodal interfaces · Video processing · Audio processing · Soft computing · Heuristic algorithms · Special education needs

## 1 Introduction

As regards the usage of PC computers in college classrooms, new research shows that they can actually increase students' engagement, attentiveness, participation and learning. However, computer employed in the classroom may entail the following adverse effects:

- isolate school students,
- distract them from the teacher,
- break emotional links between pupils,
- prevent socializing during the lesson,
- change team work habits unfavorably,
- worsen eyesight acuity,
- influence negatively body posture.

---

A. Czyzewski (✉) · B. Kostek  
Multimedia Systems Department, Gdansk University of Technology,  
Narutowicza 11/12, 80-233 Gdansk, Poland  
e-mail: ac@pg.gda.pl

Current research at the Multimedia Systems Department is intended to prove the following thesis: “technology developments can lead us toward a more natural way of using computers in general, especially in classrooms”.

In order to let computers to be used in a more natural and spontaneous way, they should fulfill the following demands:

- their presence should remain unnoticed (for as much time as possible),
- they should provide fully open platform (in contrast to some recent restricted ones),
- should be operated in natural ways, e.g. by gestures,
- should interact with human senses much better.

In order to satisfy above demands, some new ways of human-computer-interfacing should be pursued. In turn, in the technology layer their realization demands solving problems requiring a simulation of intelligent thought processes, heuristics and applications of knowledge. In particular children with so called “special educational needs”, i.e. children with various types of disabilities (communication senses problems, limb paralysis, infantile paralysis, partial mental retardation and others) can potentially benefit much from the availability of intelligent solutions helping them to learn using computers. The needs of partially impaired children motivated us at the Gdansk University of Technology to develop a prototype series of multimodal interfaces some of which are presented in this paper.

## 2 Multimodal interfaces

The following milestones can be identified in the history of man-computing machine communication:

- in Ancient China the first known interface was the gills of beads;
- in the 60’s keyboards of cards perforator machines and teletypes appeared;
- when in the 70’s the first terminals appeared, the sudden need for typing occurred, as terminals accepted only such form of input data;
- the first graphical operating system was developed in the 80’s. This interface introduced us to the mouse—essentially a simple pointing device;
- the next stage were currently very popular graphical interfaces;
- fast evolution of computing power in the 90’s allowed development of a fair speech and text recognition systems.

Still it is natural human tendency to speak, gesticulate and sometimes use hand-writing, when communication is needed, thus various solutions in this domain appear on the market since 90’s until present, including tablets with touch sensitive screen and others. Nowadays natural forms of communication are the most desired and interfaces using those are known as **multimodal interfaces**.

The subject is not a new one, however, many notions related to multimodal interfaces were hitherto conceived, including the following ones:

- Man-Machine Interaction (MMI)—(during II Word War);
- Human-Computer Interaction (HCI)—(in the 70’s);

- Human-Machine Communication (HMC);
- Perceptual User Interface (PUI);
- Natural Interactive Systems (CityplaceNIS).

The term multimodal consists of two components, namely: multiplicity and modality, where modality it is the way of transferring and receiving information. There are several kinds of information in the communication, e.g.:

- natural language,
- hands gestures and movements,
- body language,
- facial expressions,
- handwriting style.

The multimodal systems can be divided into unimodal systems—those using only one modality e.g. speech recognition or text recognition or multimodal systems—those using several modalities as an input signal, e.g. speech recognition with simultaneous gesture capture. Applications of multimodal systems are widespread most in education to provide help to pupils with special needs, including:

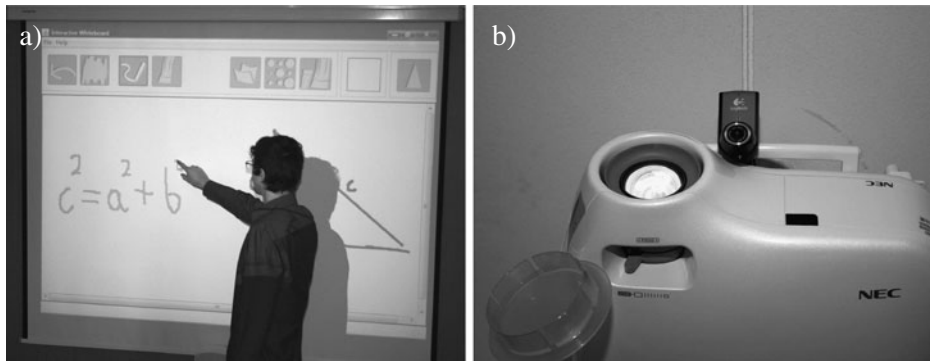
- children with attention disorders (e.g. ADHD syndrome)—multimodal interfaces give the great opportunity to improve their learning skills through stimulating different senses helping to focus attention,
- concentration training—biofeedback usage,
- educational games with multimodal interaction,
- others (some cases will be presented later on).

Currently Multimedia Systems Department is carrying out several research projects dealing with multimodal interfaces in a direct co-operation with industrial partners. Human senses: sight, hearing, touch and smell are involved. Moreover, gesture recognition with video camera image analysis is employed in many applications. The recognition based on image processing is nowadays research focus—much effort is put on eliminating the need of usage of all wire connections, sensors, gloves or other additional tools.

A common feature of all developed system is that their engineering demand non-deterministic problem solving in algorithmic and especially signal processing layers. Thus, the technology layer realization demands solving problems requiring applications of heuristics, soft computing or in general: knowledge-base systems.

### 3 Gesture-controlled interactive whiteboard

Interactive electronic whiteboards may be an effective support for students who need to see the material presented again, or are absent from school, for struggling learners, and for children with special educational needs. The disadvantage of typical electronic whiteboards is their price which is partly the result of the necessity of using electronic pens and large frames equipped with sensors. To improve whiteboard content controlling in cases the system uses a camera, vision-based gesture recognition approach can be applied. Some attempts in this domain were presented in papers



**Fig. 1** Hardware part of the system: **a** projection screen; **b** multimedia projector coupled with a webcam (Lech and Kostek 2010a)

of others researchers, e.g. Xu and Yi (2008), Maes and Mistry (2009) and Mistry and Maes (2009). Authors of the last recalled paper used a portable projector for the content displaying and a webcam. The equipment can be mounted on a helmet worn by the user. Special colorful tips are used on fingers to provide gesture controlling.

The system developed at the Multimedia Systems Department by Lech and Kostek (2010a) provides the functionality of electronic whiteboards and its essential feature is lack of the necessity of using any special manipulators or sensors. Moreover, the whiteboard content can be handled (e.g. zoomed in/out, rotated) by dynamic hand gestures. Data gloves (cyber gloves) or special tips on fingers are not needed.

The hardware part of the system is presented in Fig. 1. It is composed of a PC (dual core), a multimedia projector, a webcam and a screen for projected image. The webcam is attached to the multimedia projector in such a way that both lenses are directed at the projection screen.

### 3.1 Kalman filtering

To provide reliable hand position tracking each captured frame is appropriately processed using Kalman filters. Considering the necessity of eliminating distortions introduced by camera lens, perspective transformations and impact of light on displayed image is performed. The image processing methods used have been described in earlier papers (Lech and Kostek 2010a, b). Handling the whiteboard contents is based on recognizing dynamic hand gestures in the processed images being the result of subtracting the camera frames from the displayed images. There are 13 gestures predefined in the system. Eight gestures involve only one hand and five—both hands simultaneously. Three of the one-hand gestures are so-called grouped gestures and can be composed of other five gestures performed in an appropriate order. Each gesture is associated with the default computer action performed by emulating key pressing or mouse button clicking.

Hand movements are modelled by motion vectors designated on a few successive camera frames. Each vector  $\vec{u} = [u_x, u_y]$  is analyzed in the Cartesian coordinate system regarding speed and direction (Fig. 2; Lech and Kostek 2010a).

Two parameters of motion vectors, i.e. speed and direction, were used as a basis for gesture interpretation mechanism. Speed for motion vector within the time interval  $t_i - t_{i-1}$ , denoted as  $v_{ij}$ , where  $j = i - 1$ , was calculated according to (1). Direction for particular motion vector  $\vec{u}_{ij} = [u_x^{ij}, u_y^{ij}]$  was denoted as an angle  $\alpha_{ij}$  in relation to angle  $\varphi_{ij}$  between  $\vec{u}_{ij}$  with origin at  $[0, 0]$  and versor of y-axis, according to (2) and (3).

$$v_{ij} = \frac{\sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}}{t_i - t_{i-1}} \left[ \frac{px}{s \cdot 10^{-1}} \right] \quad (1)$$

where:  $x_i$  and  $x_{i-1}$  are  $x$  coordinates of the upper vertex of rectangle bounding the hand shape (for the left hand the coordinates denote the left upper vertex of the rectangle and for the right hand they denote the right upper vertex of this rectangle) at time  $t_i$  and  $t_{i-1}$ , respectively, and  $y_i$ ,  $y_{i-1}$  are  $y$  positions of hand at time  $t_i$  and  $t_{i-1}$ , respectively;

$$\varphi_{ij} = \frac{180^\circ \cdot \alpha_{ij} \cos \frac{u_y^{ij}}{|\vec{u}_{ij}|}}{\pi} [^\circ] \quad (2)$$

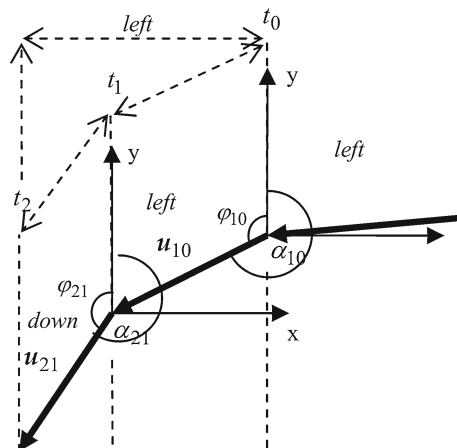
$$\alpha_{ij} = \begin{cases} \varphi_{ij}, & u_x^{ij} \geq 0 \\ 360^\circ - \varphi_{ij}, & u_x^{ij} < 0 \end{cases} \quad (3)$$

For the velocity  $v_{ij}$ , also vertical and horizontal velocities are computed using trigonometric identities for angle  $\varphi_{ij}$  in relation to angle  $\alpha_{ij}$ . The obtained horizontal and vertical velocities are expressed by (4) and (5), respectively.

$$v_{ij}^x = v_{ij} \sin \alpha_{ij} \quad (4)$$

$$v_{ij}^y = v_{ij} \cos \alpha_{ij} \quad (5)$$

**Fig. 2** Motion vectors created for semi-circular hand movement in the left direction (Lech and Kostek 2010a)



The Kalman filter (Kalman 1960) is used for estimating the state  $s$  of a system from a series of noisy measurements. The notation  $\hat{s}_{t|t-1}$  represents the estimate of  $s$  at time  $t$  given observations up to, and including at time  $t-1$ . Hence, the predicted state  $\hat{s}_{t|t-1}$  at time  $t$  is related to the state at time  $t-1$  according to the following equation:

$$\hat{s}_{t|t-1} = F_t \hat{s}_{t-1|t-1} + w_{t-1} + B_{t-1} u_{t-1} \quad (6)$$

where  $F_t$  is the transition matrix,  $w_t$  is the process noise for time evolution drawn from a zero mean multivariate normal distribution with the covariance  $Q_t$ , and  $B_{t-1}$  is an optional control matrix applied to control vector  $u_{t-1}$ . The control vector denotes an input that changes often and it is not part of the state vector because it cannot be estimated. The updated state estimate is based on the prediction and observation (measurement) according to the following equation:

$$\hat{s}_{t|t} = \hat{s}_{t|t-1} + K_t \cdot (z_t - H_t \hat{s}_{t|t-1}) \quad (7)$$

where  $K_t$  is the optimal Kalman gain, expressed by (8),  $z_t$  is the measurement, and  $H_t$  is the observation model which maps the true state space into the observed space:

$$K_t = P_{t|t-1} H_t^T S_t^{-1} \quad (8)$$

The variable  $P_{t|t-1}$  is the predicted (a priori) covariance and  $S_t$  is the residual covariance, expressed by:

$$S_t = H_t P_{t|t-1} H_t^T + R_t \quad (9)$$

where  $R_t$  is the observation noise covariance.

In the presented system Kalman filtering was used to smooth the movement trajectory resulting in raising gesture recognition effectiveness and improving accuracy of writing/drawing on the whiteboard. The filtering was implemented using the OpenCV library (Bradski and Kaehler 2008).

The state of the system (i.e. hand position) at the given moment is expressed by  $(x, y)$  position, vertical velocity and horizontal velocity according to (10);

$$s_t = [x_t, y_t, v_t^x, v_t^y] \quad (10)$$

The true state at time  $t$  is evolved from the state at time  $t-1$  by a function of speed and so the transposition matrix takes values as follows:

$$F = \begin{bmatrix} 1 & 0 & dt & 0 \\ 0 & 1 & 0 & dt \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (11)$$

where  $dt$ , expressed by (12), is a time modification of the speed depending on the camera frame rate  $f_{FR}$  and the number of frames  $n_{t_0}^{r_1}$ , based on which a singular motion vector is created (Lech and Kostek 2010a):

$$dt = c \cdot \frac{n_{t_0}^{r_1}}{f_{FR}} \quad (12)$$

**Table 1** Comparison of grouped gesture recognition efficacy without and with Kalman filters [%] (done through repetition of 20 gestures performed by 20 people)

Gesture	Without Kalman filter	With Kalman filter
Full screen	91.19	90.99
Quitting full screen	91.78	86.57
Closing application	62.96	88.89

The constant  $c$  resulted from the chosen speed unit is used to scale the speed values and is equal to 10. The obtained frame rate equals 15 fps and the singular motion vector based on three successive frames  $dt$  equals 2. Thus, applying the transition matrix to the state at the time step  $t-1$  results in the predicted state as follows:

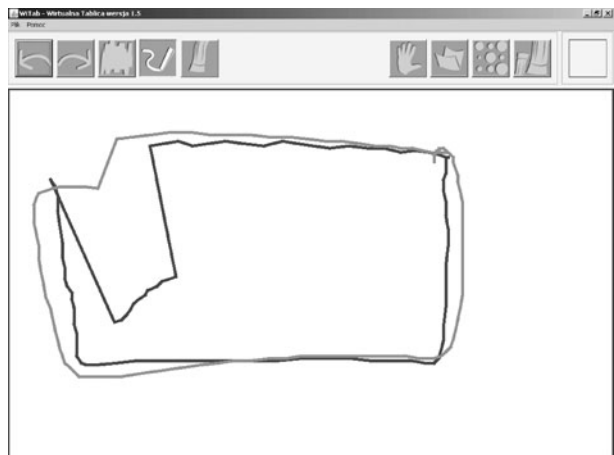
$$\hat{s}_{t|t-1} = \begin{bmatrix} x_{t|t-1} = x_{t-1|t-1} + 2 \cdot v_{t-1|t-1}^x \\ y_{t|t-1} = y_{t-1|t-1} + 2 \cdot v_{t-1|t-1}^y \\ v_{t|t-1}^x = v_{t-1|t-1}^x \\ v_{t|t-1}^y = v_{t-1|t-1}^y \end{bmatrix} \quad (13)$$

The measurement matrix  $H_t$  is initialized to identity, as well as the posteriori error covariance  $P_{t|t}$ . The process noise covariance  $Q_t$  and the observation noise covariance  $R_t$  are set to diagonal matrices with values that are equal to  $10^{-5}$  and  $10^{-1}$ , respectively.

A comparison of grouped (i.e. one-hand gestures combined with other gestures) gestures recognition efficacy without and with Kalman filters is presented in Table 1 and a visual assessment of the related results can be seen in Fig. 3.

Analyzing the image drawn (Fig. 3) one can notice that Kalman filters provided smoother drawing. However, for the grouped gestures in some cases the recognition efficacy has decreased (see Table 1). Such a situation was due to the fact that grouped gestures performed by the users composed of fast movements with rapid changes of direction by  $90^\circ$ . In such a case Kalman filter predictions change the angles which

**Fig. 3** Comparison of rectangular shapes created in poor light conditions without the Kalman filtering (*darker line*) and with the Kalman filtered hand position tracking (*brighter line*)



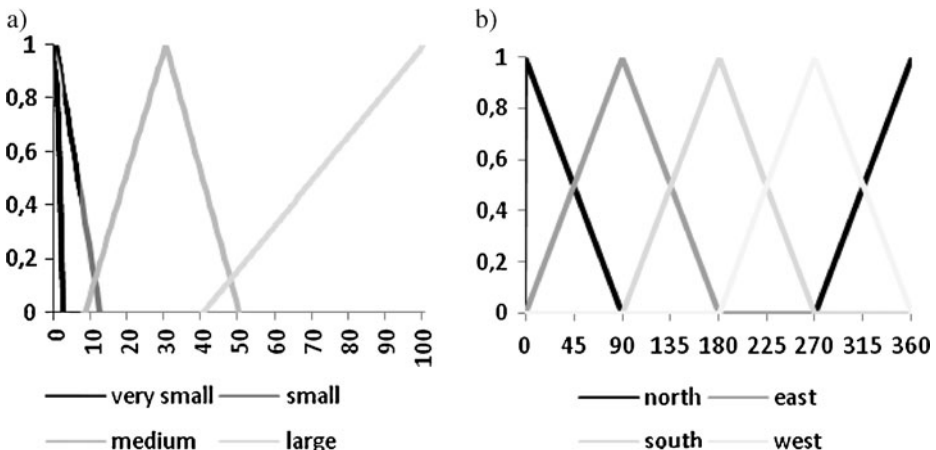
resulted in not classifying one of the component gestures. In this case for the higher recognition efficacy, the grouped gestures should be performed more slowly.

### 3.2 Fuzzy logic-based interpreter

Representing a gesture as a singular change of speed and direction over particular time interval often led to interpreting it as moving hand up—in the beginning phase of the movement—or as moving hand down—in the ending phase. Therefore, the movement trajectory in the second approach has been modeled by motion vectors created for points at time moments  $t_1$  and  $t_2$ , in relation to the moments  $t_0$  and  $t_1$ , respectively, as presented in Fig. 2 and gestures were analyzed considering a possibility of a local change of direction. Time intervals  $t_1 - t_0$  and  $t_2 - t_1$ , expressed in the number of frames retrieved from a camera, depend on camera frame rate (Lech and Kostek 2010a).

Fuzzy rules were defined based on speed and direction of motion vector over time interval  $t_2 - t_1$  and  $t_1 - t_0$  separately for left and right hand. Eight linguistic variables were proposed, i.e.: speed of left and right hand over time interval  $t_2 - t_1$ , speed of left and right hand over time interval  $t_1 - t_0$ , direction of left and right hand over time interval  $t_2 - t_1$ , direction of left and right hand over time interval  $t_1 - t_0$ , denoted as  $v_{21}^L, v_{21}^R, v_{10}^L, v_{10}^R, d_{21}^L, d_{21}^R, d_{10}^L, d_{10}^R$ , respectively. Four linguistic terms were used for speed, i.e.: *very small*, *small* (denoted later as *vsmall*), *medium* and *large*, represented by triangular membership functions as shown in Fig. 4a. The membership functions were identical for all four variables representing speed. For direction the following terms were used: *north*, *east*, *south* and *west* and also in this case triangular functions were employed as shown in Fig. 4b.

The zero-order Takagi–Sugeno fuzzy inference model (Sugeno 1985) which bases on singletons was used to express discrete rule outputs representing gesture classes. The output of the system was the maximum of all rule outputs. When this value was lower than 0.5 a movement was labelled as *no gesture*. This enabled to efficiently



**Fig. 4** Fuzzy membership functions for linguistic variables: speed (a) and direction (b)



**Table 2** Gesture recognition effectiveness for the system employing fuzzy inference and without a module of fuzzy inference, for one hand gestures (%)

	With fuzzy logic						No fuzzy logic					
	Left	Right	Up	Down	Hand steady	No gesture	Left	Right	Up	Down	Hand steady	No gesture
Left	95.0	0.0	2.3	2.6	0.0	0.1	89.5	0.0	4.9	5.6	0.0	0.0
Right	0.0	94.2	2.9	2.7	0.0	0.2	0.0	89.6	5.8	4.6	0.0	0.0
Up	0.9	0.5	98.6	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0
Down	2.2	0.9	0.0	96.9	0.0	0.0	0.0	0.0	0.0	99.8	0.0	0.2
Hand steady	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	73.3	16.7

solve the problem of classifying meaningless transitions between each two gestures to one of the gesture classes. The total number of rules equaled 30. Two examples of rules expressed in FCL code are given below (Lech and Kostek 2010a):

```
// beginning phase of hand movement in the left direction (for semi-circular motion) for left hand
RULE 1: IF directionLt0 IS north AND directionLt1 IS west AND speedLt0 IS NOT small AND
speedLt1 IS NOT small AND speedRt0 IS vsmall AND speedRt1 IS vsmall THEN gesture IS g1;
// rotate left
RULE 29: IF directionLt0 IS south AND directionLt1 IS south AND directionRt0 IS north AND
directionRt1 IS north AND (speedLt1 IS NOT vsmall AND speedLt0 IS NOT vsmall) AND (speedRt1
IS NOT vsmall AND speedRt0 IS NOT vsmall) THEN gesture IS g7;
```

The first rule describes the beginning phase of semi-circular left hand movement from right to the left side. Therefore,  $d_{10}^R$  is north and  $d_{21}^R$  is west. Since the gesture involves left hand only, the speed of the right hand should be very small. If the right hand is not present in an image, 0.0 values are given as an input to the fuzzy inference system for variables  $v_{21}^R$  and  $v_{10}^R$ . The second rule represents the gesture associated with rotating the displayed object. During the gesture performing, the left hand moves down and the right hand moves up. No local change of direction is allowed. For this reason, both  $d_{21}^L$  and  $d_{10}^L$  are south and  $d_{21}^R$ ,  $d_{10}^R$  are north. While making gestures involving both hands, speed of each hand movement can be lower than when performing a single hand gesture. Therefore, contrary to the first rule the second one allows for small speed.

Again 20 persons took part in tests. Each person was asked to repeat each gesture 18 times. Among these 18 repetitions 10 middle gesture representations were chosen. Since the system analyzes motion vectors for time intervals  $t_2 - t_1$  and  $t_1 - t_0$  in relation to each obtained camera frame, among each gesture representations there were many assignments to the particular gesture class. Sample results of a comparison between fuzzy rule-based recognition and recognition based on fixed thresholds with the analysis of the global motion vector change are presented in Table 2.

#### 4 Facial expression processing—the LipMouse application

The multimodal interface engineered at the Multimedia Systems Department to be used with people with hand motor disabilities had to fulfill the following demands:

- can be used by people with impaired hands movements,

- should enable to use lip movements and gestures,
- the application runs on a standard PC computer, controlled by video camera image processing,
- all movements of mouth (head) are converted to movements of the screen cursor.

The main task of the LipMouse (see Fig. 5) is to detect and analyze images of the user's mouth region in a video stream acquired from a web-camera. All movements of mouth (or head) are converted to movements of the screen cursor. Various parameters regarding speed of the cursor movement may be set according to the user's preferences. The LipMouse also detects three mouth gestures: opening the mouth, sticking out the tongue and forming puckered lips. Each gesture may be associated with an action, which may be freely chosen by the user.

Translation between the current mouth position and the screen cursor movements is determined by three parameters: threshold  $t$ , sensitivity  $s$  and acceleration  $a$ , and is given by equation (in vertical and horizontal direction separately):

$$\Delta x = \begin{cases} a(\Delta p - t)^2 + s(\Delta p - t), & \Delta p > t \\ -a(\Delta p + t)^2 + s(\Delta p + t), & \Delta p < -t \\ 0, & -t \leq \Delta p \leq t \end{cases} \quad (14)$$

where  $\Delta x$  denotes the resulting screen cursor movement distance (in screen pixels) and  $\Delta p$  is the distance between the mouth position and the reference position. The mouth position shift  $\Delta p$  (in horizontal and vertical direction), is calculated as follows:

$$\Delta p_x = \frac{m_x - r_x}{w}, \quad \Delta p_y = \frac{m_y - r_y}{w} \quad (15)$$



**Fig. 5** Main window of the LipMouse application

where  $(m_x, m_y)$  denotes the current mouth position (the center of the mouth region upper boundary) in video frame pixels,  $(r_x, r_y)$  is the reference mouth position and  $w$  denotes the current mouth region width. Normalization of the mouth position shift by the mouth width assures that a screen cursor moves in the same way independently of the user's face distance from the camera.

The threshold  $T$  is the minimal mouth shift from the reference position that is required for the screen cursor to start moving. The greater the threshold is, the more the user's head needs to be turned in order to move the screen cursor. Sensitivity  $s$  and acceleration  $a$  determine directly how the mouth shift value is translated into screen cursor movement speed. The greater values of these parameters are, the faster the screen cursor moves at the same mouth shift from the reference position.

Lip gesture recognition is performed by an artificial neural network (ANN) (Dalka and Czyzewski 2010a). Each image frame is classified independently. The number of the ANN inputs corresponds to the number of lip image features and is equal 168 or 171, respectively, depending on the chosen variant of lip region extracted (Dalka and Czyzewski 2010b). A feed-forward ANN with one hidden layer is used to detect lip gestures. Based on initial experiments, the number of neurons in the hidden layer was set to eight. There are four outputs from the ANN, each one is related with one type of gestures recognized by the ANN. Three of them are: opening the mouth, sticking out the tongue and forming puckered lips, the fourth one returns a natural, neutral facial expression, this means that no real lip gesture is present.

In order to minimize the number of false-positives, post processing of the ANN output vector  $o$  is performed in order to improve reliability of classification. The maximum value of ANN output  $o_{\max}$  is converted according to the following equation:

$$o'_{\max} = \frac{o_{\max}}{\sum_{i=0}^3 o_i} \cdot o_{\max}, \quad o_i \in [0, 1] \quad (16)$$

If  $o'_{\max}$  is greater or equal to the threshold  $T$ , a gesture connected with the output  $o_{\max}$  is returned as the recognized gesture; otherwise, the neutral gesture is returned, which means that no real gesture is detected. This method assures that if the neural network output does not exceed a certain threshold value, no gesture is detected in order to minimize false-positives ratio. It can be noticed that  $T = 0$  turns off the ANN output post-processing.

The ANN is trained with a resilient backpropagation algorithm (RPROP) (Riedmiller and Braun 1993). Training data are acquired during the calibration phase which is required at the beginning of every session with the application. During each stage, 60 frames containing gesture images are gathered (video rate is 15 fps). Feature vectors obtained from these frames form training vectors (80% of all vectors) and testing vectors (every fifth vector). This means that total 192 feature vectors (48 for every gesture) are used for ANN training and 48 vectors are used to test the ANN after training (12 for every gesture). Five neural networks are trained based on the same data and the one with the smallest error rate of validation vector classification (with post-processing threshold  $T = 0.5$ ) is used for lip gesture recognition.

In order to facilitate lip gesture recognition by ANN, an algorithm for determining region of the image containing lips must be very precise and has to be robust against

head movements in the vertical and horizontal directions. In order to locate lips, a series of face image transformations is performed. First, an image is smoothed with Gaussian filter and converted from RGB color space into the CIE LUV space (Leung et al. 2004). One should recall here that the CIE LUV color space is designed to be perceptually uniform, meaning that a given change in value corresponds roughly to the same perceptual difference over any part of the space.  $L$  component of the CIE LUV, i.e. luminance, is approximately uniformly spaced but is quite indicative of the actual visual differences. Chrominance components are  $U$  and  $V$  for CIE LUV. It should also be mentioned that the RGB-representable colors occupy only part of the LUV color space. Furthermore, the smoothed image is also transformed with DHT (Discrete Hartley Transform) (Moran and Pinto 2007), according to the formula:

$$\begin{bmatrix} C_1 \\ C_2 \\ C_3 \end{bmatrix} = \begin{bmatrix} 0.5774 & 0.5774 & 0.5774 \\ 0.5774 & 0.2113 & -0.7887 \\ 0.5774 & -0.7887 & 0.2113 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (17)$$

The  $U$  component of the LUV space and the third component  $C_3$  of the DHT transform are used for further processing, because they provide distinct, linear separation of lip and non-lip areas.

There are 171 lip image features used for lip gesture recognition. They can be divided into four groups. The first one is used only when the first invariant  $V_1$  of the lip region extracted is chosen and it contains three parameters: the width and the height of the ellipse approximating the lip shape and the angular eccentricity of the ellipse which is given by the following equation:

$$e = \arccos\left(\frac{a}{b}\right) \quad (18)$$

where  $a$  is the shorter, and  $b$ —the longer axis of the ellipse.

The second group of parameters is formed by the normalized, 20-point luminance histogram of the lip region. The third group contains Hu sets of invariant image moments (Hu 1962). Four sets of Hu moments are calculated based on four equal-sized, non overlapping luminance images the lip region is divided into. Each Hu set contains seven parameters which returns a total number of 28 features in the third group (Dalka and Czyzewski 2010b).

The last group of parameters is based on co-occurrence matrices.  $T$  co-occurrence matrix, also referred to as a co-occurrence distribution, is defined over an image to be the distribution of co-occurring values at a given offset (Haralick et al. 1973; Clausi 2002). It is commonly used as a texture description. A co-occurrence  $N \times N$  matrix  $C$  defined over an  $n \times m$ ,  $N$ -color image  $I$ , parameterized by a spatial offset  $(\Delta x, \Delta y)$ , is given by the following equation:

$$C(i, j) = \sum_{p=1}^n \sum_{q=1}^m \begin{cases} 1, & I(p, q) = i \text{ and } I(p+\Delta x, q+\Delta y) = j \\ 0, & \text{otherwise} \end{cases} \quad (19)$$

A co-occurrence matrix  $C$  is converted to the matrix symmetric about its diagonal  $C_s$  by adding its own transposition:

$$C_s = C + C^T \quad (20)$$

A symmetrical co-occurrence matrix is identical for offsets symmetrical around (0, 0), e.g. (−1, 0) and (1, 0) or (−1, −1) and (1, 1). In the last step the matrix is expressed as a probability by normalizing its elements according to the formula:

$$P(i, j) = \frac{C_s(i, j)}{\sum_{i=0, j=0}^{N-1} C_s(i, j)} \quad (21)$$

As mentioned before, in order to extract feature vectors used by ANN to classify lip gestures, a face region containing user lips is converted from RGB to CIE LUV color space. The space is perceptually uniform which means that a change in color values results in the same change perceived by a human eye over any part of the space. A set of symmetrical, normalized, co-occurrence matrices  $P$  is calculated for three lip image representations: the luminance  $L$  and chrominance  $U$  of the CIE LUV color space and the first vertical derivative of the luminance image calculated with the Sobel operator (Young et al. 1998). Each set contains eight matrices  $P$  calculated for four different directions of offsets; for  $0^\circ$  (0, 1) and (0, 2) offsets are used, for  $45^\circ$ : (1, 1) and (2, 2), for  $90^\circ$ : (1, 0) and (2, 0), for  $135^\circ$ : (1, −1) and (2, −2). A set of symmetrical, normalized, co-occurrence matrices  $P$  is calculated for three lip image representations. Pixel values are further quantized into 25 equally-spaced ranges in each channel of the CIE LUV space separately in order to make real-time feature extraction possible. Therefore each co-occurrence matrix contains  $25 \times 25$  elements.

Five statistical parameters are calculated for every co-occurrence matrix  $P$ . They are as follows: contrast, energy, mean, standard deviation and correlation.

This produces a total number of 120 parameters (i.e. three image representations  $\times$  8 co-occurrence matrices  $\times$  5 parameters) based on co-occurrence matrices contained in the feature vector.

Detailed results of lip gesture classification are shown in Table 3. It is seen that increasing the ANN post-processing threshold  $T$  improves the effectiveness of neutral gesture recognition and worsens results of other three gestures classification. It is assumed that an optimum value of the  $T$  threshold is 0.5, which provides compromise between the effectiveness and the false-positive ratio of real gesture recognition.

**Table 3** Results of lip gesture classification (optimum invariant of lip gesture extraction is used for every test recording)

Gesture	No. of image frames	Effectiveness of the lip gesture classification for different ANN post-processing thresholds (%)			
		$T = 0$	$T = 0.25$	$T = 0.5$	$T = 0.75$
Neutral (no gesture)	6,120	92.9	93.8	94.9	96.1
Mouth opening	6,120	95.4	94.8	92.4	89.2
Forming puckered lips	6,120	92.5	91.8	88.2	83.6
Sticking out the tongue	6,120	94.1	93.2	91.3	85.6
All gestures	24,480	93.7	93.4	91.7	88.6

## 5 Audio modality: speech stretcher

A non-uniform real-time scale speech modification algorithm was designed to improve the perception of speech by people with the hearing resolution deficit (Kupryjanow and Czyzewski 2010). The software employing this algorithm enables to use an ultra-portable computer (e.g. smartphone) as a speech communication interface for people suffering from certain type of central nervous system impairments, which can impede learning.

The block diagram of the proposed algorithm is presented in Fig. 6. The algorithm provides a combination of voice activity detection, vowel detection, rate of speech estimation and time-scale modification algorithms. Signal processing is performed in time frames in the following order:

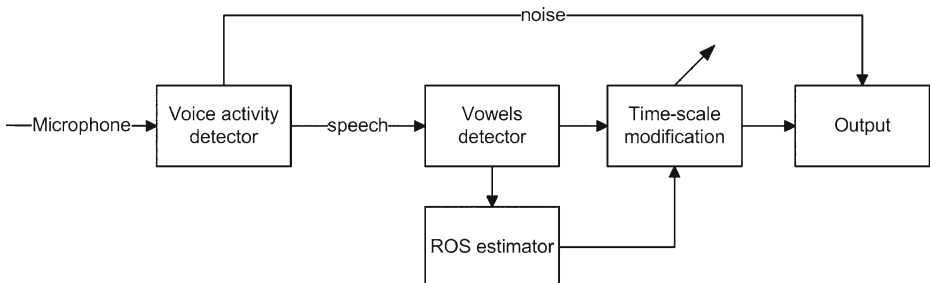
1. voice activity detector examines speech presence,
2. for noisy components the frame synchronization procedure is performed; if the output signal is not synchronized with the input then noise sample frames are not sent to the output,
3. speech sample frames are tested in order to find vowels,
4. information about vowels locations is used by the rate of speech estimator to determine the speech rate,
5. speech frames are stretched up with different stretching factors.

As speech signal is usually unrepeatable and often modeled as a stochastic process, above operations (3)–(5) demand a heuristic approach to computing.

The vowel detection algorithm is based on the assumption that all vowels amplitude spectra are consistent. To quantify this similarity a parameter called *PVD* (peak-valley difference) was used (Moattar et al. 2010). Initially *PVD* was introduced for the robust voice activity detection. It is defined by the following formula (22):

$$PVD(VM, A) = \frac{\sum_{k=0}^{N-1} (A(k) \cdot VM(k))}{\sum_{k=0}^{N-1} VM(k)} - \frac{\sum_{k=0}^{N-1} (A(k) \cdot (1 - VM(k)))}{\sum_{k=0}^{N-1} (1 - VM(k))} \quad (22)$$

where  $PVD(VM, A)$  is the value of peak-valley difference for one frame of the input signal,  $A(k)$  is the value of the  $k$ th spectral line of the input signal magnitude spectrum and  $VM(k)$  is the value of the  $k$ th value in the vowel model vector.



**Fig. 6** Non-uniform real-time scale speech modification algorithm block diagram

The *VM* is created in the training stage on the basis of the average magnitude spectra calculated for the pre-recorded vowels. The model consists of binary values, where 1 is placed in the position of the peak in the average magnitude spectrum and 0 for all other positions. When the magnitude spectrum of the input signal is highly correlated with the vowels spectra, the *PVD* value is high. Therefore, the *PVD* have higher values for the vowels than for consonants or silence parts.

Vowels detection is executed only for speech frames. The algorithm is based on time frames with the duration of 23 ms. Each signal frame is windowed using a triangular window defined as:

$$\omega(n) = \begin{cases} \frac{2n}{L}, & 1 \leq n \leq \frac{L+1}{2} \\ \frac{2(L-n+1)}{L}, & \frac{L}{2} + 1 \leq n \leq L \end{cases} \quad (23)$$

where  $L$  is the size of the window and  $n$  is the sample number. This type of window ensures higher accuracy of vowel detection than other shapes.

Vowel detection requires the initialization step which is performed in parallel to the initialization of the voice activity detection algorithm. In this step the threshold for the *PVD* is calculated as the mean value of first 40 frames of the signal according to the formula:

$$Pth = C \frac{\sum_{n=1}^N PVD(n)}{N} \quad (24)$$

where  $Pth$  is initial value of the threshold,  $PVD(n)$  is the value of peak-valley difference for the  $n$ th signal frame,  $N$  is the number of frames that were used for initial threshold calculation,  $C$  is the correction factor. The correction factor was selected experimentally and was set to 1.1.

For every signal frame the *PVD* value is determined and smoothed by calculating the average of the last three values. The signal frame is marked as a vowel when: the value of the smoothed *PVD* is higher than  $Pth$  threshold and it has a local maximum in the *PVD* curve or its value is higher than 70% of the value of the last local maximum. If the value is lower than  $Pth$ , then the decision of the voice activity detector is corrected and the frame is marked as silence. For other situations the frame is assigned to the consonant class.

Rate of Speech (ROS) is a useful parameter in many speech processing systems. For the most part it is used in the automatic speech recognition (ASR) domain. There are a number of options for feature extraction in the ASR, among those many parameters are highly related to ROS. Hence, ROS is used to adjust the HMM model for different speech rates (Zheng et al. 2000).

For real-time unknown input signal, ROS estimation could only be done by the statistical analysis. In this work, ROS was evaluated in terms of the number vowels per second (VPS), since the latter parameter may be considered as a simplified form of another measure, i.e. the number of syllables per second (SPS). Therefore, ROS is defined here as (25):

$$ROS(n) = \frac{N_{\text{vowels}}}{\Delta t} \quad (25)$$

**Table 4** Mean value and standard deviation of ROS calculated for different speech rates

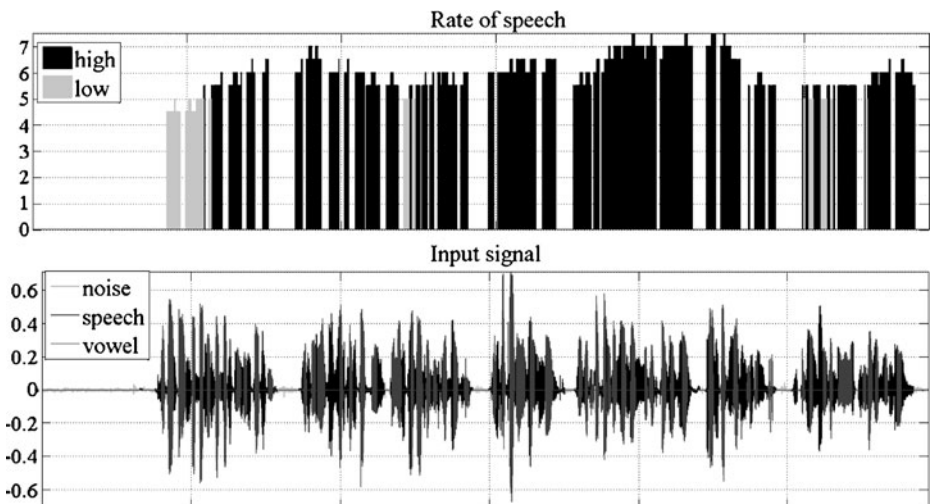
Speech rate	Low	Medium	High
$\mu(\text{ROS})$ (vowels/s)	2.23	2.4	2.56
$\Delta(\text{ROS})$ (vowels/s)	0.6	0.58	0.57

Mean value and standard deviation of ROS calculated for the different speech rates for three persons reading the same phrase with three speech rates: high, medium and low are shown in Table 4.

It can be seen that, because of the high value of the standard deviation (nearly 0.6 for all classes) and as a consequence of the low distance between the neighbor classes, only two classes could be separated linearly using the instantaneous ROS value. On the basis of the statistics, the *ROS* value was set to 2.5 vowel/s. In Fig. 7 waveforms corresponding to the recorded female high rate speech with the estimated speech rate are presented.

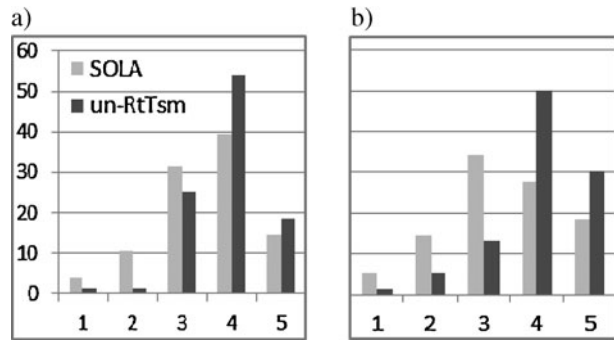
For time-scale modification of speech an algorithm based on the SOLA algorithm (Synchronous Overlap-and-Add) was applied which in its original form uses constant values of the analysis/synthesis frame sizes and the analysis/ synthesis time shift (Pesce 2000). This algorithm also provides good quality of the processed speech (Verhelst and Roelands 1993; Kupryjanow and Czyzewski 2009).

To achieve high quality of the stretched speech, the analysis/synthesis frame size and the analysis time shift should be selected properly, i.e. frame length  $L$  should cover at least one period of the lowest speech component and in the synthesis stage, for all used scaling factors  $\alpha(t)$ , the overlap size should at least be  $L/3$  length. For the designed algorithm  $L$  value was set to 46 ms and the analysis time shift  $S_a$  to 11.5 ms.

**Fig. 7** Speech rate recognition for female voice at a high speech rate



**Fig. 8** Signal quality assessment for different speech rates: **a** low, **b** high (grey bars represent the SOLA algorithm, darker bars represent the proposed heuristic algorithm)



The synthesis time shift  $S_s$  is dependent on the current value of the scaling factor  $\alpha(t)$ . The scaling factor is defined as in (26):

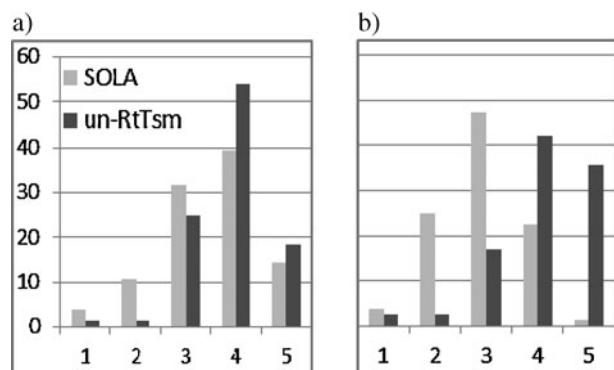
$$\alpha(t) = \frac{S_s}{S_a} \quad (26)$$

Synchronization between two synthesized overlapped frames is obtained by calculating the highest similarity point which is determined by the maximum of the cross-correlation function calculated for the overlapped parts of successive frames.

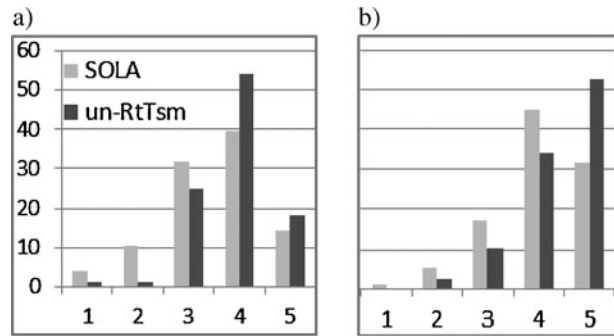
To reduce the duration of the stretched speech and to improve quality of the modified signal, the scaling factor is changed according to different speech content. For both speech rates (low and high) vowels are stretched up with the designed scale factor value ( $\alpha(t) = \alpha_d$ , being the value that is specified for the processing), and noise is not modified ( $\alpha(t) = 1$ ) or removed from the signal dependently on the input/output synchronization state. For the low rate speech consonants are stretched up with the factor lower than  $\alpha_d$  and equal to  $\alpha(t) = 0.8 \cdot \alpha_d$ , and for the high rate speech consonants are not stretched ( $\alpha(t) = 1$ ).

Quality of the engineered speech stretching algorithm was assessed in subjective tests performed by 19 healthy persons (2 women, 17 men) (Kupryjanow and Czyzewski 2010). Each person had to assess quality of speech stretched using the

**Fig. 9** Speech naturalness assessment for different speech rates: **a** low, **b** high



**Fig. 10** Speech intelligibility assessment for different speech rates: **a** low, **b** high



typical SOLA algorithm implementation and the proposed algorithm. Two values of the stretching factors were chosen: 1.9 and 2.1. Four recordings were used during the experiment: two spoken with the low rate, and two with the high rate. Both of them were spoken by a woman and a man. In all recordings the same phrase was uttered.

Three parameters were rated during tests: signal quality (see Fig. 8), speech naturalness (see Fig. 9) and speech intelligibility (see Fig. 10). The assessment was made using the following scale: 1—very poor, 2—poor, 3—medium, 4—good, 5—very good. Test results revealed that for both speech rates, as well as for all parameter evaluated, histograms that represent the proposed algorithm assessment have higher located gravity centers than for the SOLA algorithm.

Recently the proposed algorithm working in real-time has been implemented on a mobile device (the Apple iPhone platform).

## 6 Conclusions

Authors of this paper and their co-workers believe that in the future multimodal interfaces will enable a more natural control of computers with speech, gestures, eye movements and face expression engaging human senses interactively in a much broader way than today. Consequently, future learning systems will engage:

- blending technologies,
- convergence of online and face-to-face education,
- customized pedagogy,
- students as knowledge generators, not just consumers,
- immersive, gaming environment for teaching.

Besides three examples presented in this paper, many more multimodal interfaces are currently under development at our Multimedia Systems Department, including: biofeedback-based brain hemispheric synchronizing man-machine interface (Kaszuba et al. 2010), virtual computer touch pad (Kupryjanow et al. 2010), browser controller employing head movements (Kosikowski et al. 2010), intelligent tablet pen (Ody et al. 2010) and scent emitting computer interface (Kotarski et al. 2011).

**Acknowledgements** Research funded within the project no. POIG.01.03.01-22-017/08, entitled “Elaboration of a series of multimodal interfaces and their implementation to educational, medical, security and industrial applications”. The project is subsidized by the European Regional Development Fund and by the Polish State Budget.

**Open Access** This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

## References

- Bradski, G., & Kaehler, A. (2008). *Learning OpenCV: Computer vision with the OpenCV library*. Paris: O'Reilly.
- Clausi, D. A. (2002). An analysis of co-occurrence texture statistics as a function of grey-level quantization. *Canadian Journal of Remote Sensing*, 28(1), 45–62.
- Dalka, P., & Czyzewski, A. (2010a). Controlling computer by lip gestures employing neural network. In *Proc. 7th international conference on rough sets and current trends in computing (RSCTC 2010)* (pp. 80–89). Warsaw, Poland, 28–30 June 2010.
- Dalka, P., & Czyzewski, A. (2010b). Human-computer interface based on visual lip movement and gesture recognition. *International Journal of Computer Science and Applications*, 7(3), 124–139.
- Haralick, R. M., Shanmugam, K., & Dinstein, I. (1973). Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, 3(6), 610–621.
- Hu, M. K. (1962). Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, 8(2), 179–187.
- Kalman, R. R. (1960). A new approach to linear filtering and prediction problems. *Transaction of the ASME—Journal of Basic Engineering*, 82(1), 35–45.
- Kaszuba, K., Kopaczewski, K., Ody, P., & Kostek, B. (2010). Biofeedback-based brain hemispheric synchronizing employing man-machine interface. In G. A. Tsihrintzis, et al. (Eds.), *Intelligent interactive multimedia systems and services, Springer Verlag. Proc. KES 2010, the 3rd international symposium on intelligent and interactive multimedia: Systems and services* (pp. 59–69). Baltimore, USA, 28–30 July 2010.
- Kotarski, M., Smulko, J., Czyzewski, A., & Melkonyan, S. (2011). Fluctuation-enhanced scent sensing using a single gas sensor. *Sensors & Actuators: B. Chemical*, 157 85–91.
- Kosikowski, L., Dalka, P., & Czyzewski, A. (2010). Multimedia browser controlled by head movements. In *37 conf. and exhibition on computer graphics and interactive techniques. SIGGRAPH*, Los Angeles, USA, 25–29 July 2010.
- Kupryjanow, A., & Czyzewski, A. (2009). Time-scale modification of speech signals for supporting hearing impaired schoolchildren. In *Proc. of international conference NTAV/SPA, new trends in audio and video, signal processing: Algorithms, architectures, arrangements and applications* (pp. 159–162). Poznań, 24–25 September 2009.
- Kupryjanow, A., & Czyzewski, A. (2010). Real-time speech-rate modification experiments. In *Audio Engineering Society convention, preprint no. 8052*, London, GB, 22–25 May 2010.
- Kupryjanow, A., Kunka, B., & Kostek, B. (2010). UPDRS tests for diagnosis of Parkinson's Disease employing virtual-touchpad. In *4th international workshop on management and interaction with multimodal information content—MIMIC '10*, Bilbao, Spain, 30 August–3 September 2010.
- Lech, M., & Kostek, B. (2010a). Fuzzy rule-based dynamic gesture recognition employing camera & multimedia projector. In *Advances in intelligent and soft computing: Multimedia & network information systems*. New York: Springer.
- Lech, M., & Kostek, B. (2010b). Gesture-based computer control system applied to the interactive whiteboard. In *2nd international conference on information technology ICIT'2010 Gdansk* (pp. 75–78). 28–30 June 2010.
- Leung, S., Wang, S., & Lau, W. (2004). Lip image segmentation using fuzzy clustering incorporating an elliptic shape function. *IEEE Transactions on Image Processing*, 13(1), 51–62.
- Maes, P., & Mistry, P. (2009). Unveiling the “Sixth Sense”, game-changing wearable tech. In *TED 2009*. Long Beach, CA, USA.
- Mistry, P., & Maes, P. (2009). SixthSense—A wearable gestural interface. In *SIGGRAPH Asia 2009, emerging technologies*. Yokohama, Japan.

- Moattar, M., Homayounpour, M., & Kalantari, N. (2010). A new approach for robust realtime voice activity detection using spectral pattern. In *ICASSP conference*, 14–19 March.
- Moran, L. E. L., & Pinto, R. E. (2007). Automatic extraction of the lips shape via statistical lips modelling and chromatic feature. In *Electronics, robotics and automotive mechanics conference, CERMA* (pp. 241–246).
- Ody, P., Czyzewski, A., Grabkowska, A., & Grabkowski, M. (2010). Smart Pen—New multimodal computer control tool for graphomotorical therapy. *Intelligent Decision Technologies Journal*, 4(3), 197–209.
- Pesce, F. (2000). Realtime-stretching of speech signals. In *Proceedings of the COST G-6 conference on digital audio effects (DAFX-00)*, Verona, Italy, 7–9 December.
- Riedmiller, M., & Braun, H. (1993). A direct adaptive method for faster backpropagation learning: The RPROP algorithm. In *Proc. ICNN*.
- Sugeno, M. (1985). *Industrial applications of fuzzy control*. Amsterdam: Elsevier.
- Verhelst, W., & Roelands, M. (1993). An overlap-add technique based on waveform similarity (WSOLA) for high quality time-scale modification of speech. In *ICASSP-93* (Vol. 2). Minneapolis, USA, 27–30 April.
- Xu, R., & Yi, D. (2008). A computer vision based whiteboard capture system. In *WACV 2008, IEEE workshop on applications of computer vision* (pp. 1–6).
- Young, I., Gerbrands, J., & Vliet, L. (1998). *Fundamentals of image processing*. Delft: Delft University of Technology.
- Zheng, J., Franco, H., Weng, F., Sankar, A., & Bratt, H. (2000). Word-level rate-of-speech modeling using rate-specific phones and pronunciations. In *Proc. IEEE int. conf. acoust. speech signal process, Istanbul* (Vol. 3, pp. 1775–1778).