EURASIP Journal on Applied Signal Processing 2002:11, 1213–1227

Audio-Visual Speech Recognition Using MPEG-4 Compliant Visual Features

Petar S. Aleksic
Department of Electrical and Computer Engineering, Northwestern University, 2145 North Sheridan Road, Evanston, IL 60208-3118, USA
Email: apetar@ece.nwu.edu

Jay J. Williams
Department of Electrical and Computer Engineering, Northwestern University, 2145 North Sheridan Road, Evanston, IL 60208-3118, USA
Email: jjw@ece.nwu.edu

Zhilin Wu
Department of Electrical and Computer Engineering, Northwestern University, 2145 North Sheridan Road, Evanston, IL 60208-3118, USA
Email: zlwu@ece.nwu.edu

Aggelos K. Katsaggelos
Department of Electrical and Computer Engineering, Northwestern University, 2145 North Sheridan Road, Evanston, IL 60208-3118, USA
Email: aggk@ece.nwu.edu

Received 3 December 2001 and in revised form 19 May 2002

We describe an audio-visual automatic continuous speech recognition system, which significantly improves speech recognition performance over a wide range of acoustic noise levels, as well as under clean audio conditions. The system utilizes facial animation parameters (FAPs) supported by the MPEG-4 standard for the visual representation of speech. We also describe a robust and automatic algorithm we have developed to extract FAPs from visual data, which does not require hand labeling or extensive training procedures. The principal component analysis (PCA) was performed on the FAPs in order to decrease the dimensionality of the visual feature vectors, and the derived projection weights were used as visual features in the audio-visual automatic speech recognition (ASR) experiments. Both single-stream and multistream hidden Markov models (HMMs) were used to model the ASR system, integrate audio and visual information, and perform a relatively large vocabulary (approximately 1000 words) speech recognition experiments. The experiments performed use clean audio data and audio data corrupted by stationary white Gaussian noise at various SNRs. The proposed system reduces the word error rate (WER) by 20% to 23% relatively to audio-only speech recognition WERs, at various SNRs (0–30 dB) with additive white Gaussian noise, and by 19% relatively to audio-only speech recognition WER under clean audio conditions.

Keywords and phrases: audio-visual speech recognition, facial animation parameters, snake.

1. INTRODUCTION

Human listeners use visual information, such as facial expressions, and lips and tongue movement, in order to improve perception of the uttered audio signal [1]. Impaired hearing individuals, using lipreading, or speechreading, can achieve very good speech perception [1, 2, 3, 4]. The use of visual information in addition to audio, improves speech understanding especially in noisy environments. Visual information, obviously independent of audio noise, is complementary to the audio signal, and as such can improve speech perception even in noise-free environments [5].

In ASR, a very active area of research over the last decades, visual information is ignored. Because of this the performance of the state of the art systems is much worse than that of humans, especially in the presence of audio noise [6, 7]. Hence, in order to achieve noise robust speech recognition, the interest in the area of audio-visual speech recognition (AVSR), also known as automatic lipreading or speechreading [8, 9, 10, 11], has been growing over the last

several years. Improving ASR performance by exploiting the visual information of the speaker's mouth region is the main objective of AVSR. Visual features, usually extracted from the mouth area, thought to be the most useful for ASR are the outer lip contour, the inner lip contour, the teeth and tongue location, and the pixel intensities (texture) of an image of the mouth area. Choosing the visual features that contain the most useful information about the speech is of great importance. The improvement of the ASR performance depends strongly on the accuracy of the visual feature extraction algorithms. There are three main approaches for visual feature extraction from image sequences; image-based, model-based, and combination approaches. In the image-based approach, the transformed mouth image (by, for example PCA, Discrete Wavelet Transform, Discrete Cosine Transform) is used as a visual feature vector. In the model-based approach, the face features important for visual speech perception (mainly lip contours, tongue, and teeth positions) are modeled, and controlled by a small set of parameters which are used as visual features [12]. In the combination approach, features obtained from the previous two methods are combined and used as a new visual feature vector. A number of researchers have developed audio-visual speechreading systems, using image-based [13, 14, 15, 16], model-based [16, 17, 18, 19, 20], or combination approaches [13, 21] to obtain visual features. The reported results show improvement over audio-only speech recognition systems. Most of the systems performed tests using a small vocabulary, while recently results of the audio-visual ASR performance improvement over audio-only ASR performance on a large vocabulary were shown [13, 14].

In this paper, we utilize a model-based feature extraction approach. The use of templates, defined as certain geometric shapes, and their fitting to the visual features can, in some cases, be successfully used as a visual feature extraction algorithm [22, 23]. On the other hand, face features cannot be represented accurately enough with simple geometric shapes, and in order to achieve a close fit of the templates to the real visual features the order of the geometric shapes has to be increased, which increases computational requirements. The active contour method is a relatively new method for extraction of visual features, and is very useful in cases when it is hard to present the shape of an object with a simple template [16, 21]. This method, however, is sensitive to random noise, and to certain salient features, such as lip reflections close to the desired features, which can affect the deformation of the snake.

MPEG-4 is an audio-visual object-based video representation standard supporting facial animation. MPEG-4 facial animation is controlled by the facial definition parameters (FDPs) and FAPs, which describe the face shape and movement, respectively [24, 25]. A synthetic (avatar's) face can be animated with different shapes and expressions by using FDPs and FAPs. The MPEG-4 standard defines 68 FAPs. Transmission of all FAPs at 30 frames per second requires only around 20 kbps (or just a few kbps, if MPEG-4 FAP interpolation is efficiently used [26]), which is much lower than standard video transmission rates. There are many applications that can use the information contained in FDPs and FAPs, such as communicating with hearing impaired people, and customizing ATM machine interfaces (by storing a set of FDPs on a card, or in a database on a local disc). When video conferencing is not feasible, due to bandwidth limitations, audio, FDPs and FAPs could be utilized to transmit a synthesized facial representation of the individual. FAPs contain important visual information that can be used in addition to audio information in ASR [27]. Another application area is automatic transcription of speech to text. FAPs require much less storage space than video, and can be stored together with audio and used to improve automatic transcription of speech to text. To the best of our knowledge no results have been previously reported on the improvement of AVSR performance when FAPs are used as visual features with a relatively large vocabulary audio-visual database of about 1000 words. Reporting on such results is the main objective of this paper.

In this paper, we first describe an automatic and robust method for extracting FAPs, by combining active contour and templates algorithms. We use the Gradient Vector Field (GVF) snake, since it has a large capture area, and parabolas as templates. We next describe the audio-visual ASR systems we developed, utilizing FAPs for the visual representation of speech. The single-stream and multistream HMMs were used to model the audio-visual information integration, in order to improve speech recognition performance over a wide range of audio noise levels. In our experiments, we utilize a relatively large vocabulary (approximately 1000 words) audio-visual database, the Bernstein Lipreading Corpus [28]. The results of the performance improvement in noisy and noise-free speech conditions are reported.

The rest of the paper is organized as follows. In Section 2, the database used is described. In Section 3, the automatic algorithm for visual feature extraction is presented. In Section 4, the method applied on the extracted FAP vectors in order to decrease the dimensionality of the visual feature observation vectors is described. In Section 5, the audio-visual integration methods, namely early integration (single-stream HMMs) and late integration (multistream HMMs), are presented. Finally, in Sections 6 and 7 we describe the audio only, and audio-visual ASR experiments performed, we compare the results, and present conclusions.

2. THE AUDIO-VISUAL DATABASE

This work utilizes speechreading material from the Bernstein Lipreading Corpus [28]. This high quality audio-visual database includes a total of 954 sentences, of which 474 were uttered by a single female talker, and the remaining 480 by a male talker. For each of the sentences, the database contains a speech waveform, a word-level transcription, and a video sequence time synchronized with the speech waveform. The raw visual observations framed the head and shoulders of the speaker against a light blue background. Each utterance began and ended with a period of silence. The vocabulary size is approximately 1,000 words. The average utterance length

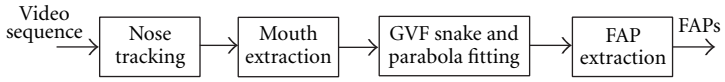
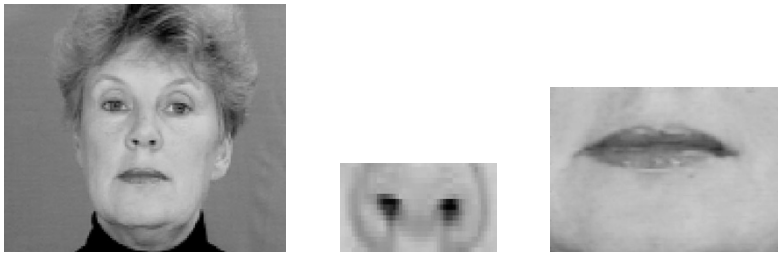


Figure 1: FAP extraction system.



(a) (b) (c)

Figure 2: (a) Neutral facial expression image; (b) extracted nose template; (c) extracted mouth image.

is approximately 4 seconds. In order to extract visual features from the Bernstein Lipreading Corpus, the video was sampled at a rate of 30 frames/sec (fps) with a spatial resolution of 320×240 pixels, 24 bits per pixel. The luminance information was used in the development of the algorithms and the experiments. Audio is collected at a rate of 16 kHz (16 bits per sample).

3. VISUAL FEATURE EXTRACTION

Figure 1 illustrates the FAP extraction system we have implemented. In order to extract the mouth area from the Bernstein Lipreading Corpus, and extract the FAPs, a neutral facial expression image was chosen among the sampled video images (Figure 2a). A 17×44 image of the nostrils, (shown enlarged in Figure 2b) was extracted from the neutral facial expression image to serve as a template for the template matching algorithm. The nostrils were chosen since they did not deform significantly during articulation. The template matching algorithm, applied on the first frame of each sequence, locates the nostrils by searching a 10×10 pixel area centered at the neutral face nose location, for the best match. Once the location has been identified, a rectangular 90×68 pixel region is extracted enclosing the mouth (shown enlarged in Figure 2c). In the subsequent frames, the search area is constrained to a 3×3 pixel area centered at the nose location in the previous frame. Since the position of the speaker's head in the Bernstein database does not change significantly during articulation, it was not necessary to compensate for scaling or rotation, just for translation.

3.1. Facial animation parameters

The MPEG-4 standard defines 68 FAPs. They are divided into 10 groups, as shown in Figure 3 and Table 1, which describe the movement of the face [24, 25]. These parameters are either high level parameter (group 1), that is, parameters that describe the facial expressions, or low-level parameters (groups 2–10), that is, parameters which describe displacement of the specific single point of the face. FAPs control the key features of the mesh model of a head, shown in Figure 4a (the corresponding texture model is shown in Figure 4b), and animate facial movements and expressions [29]. In this work, only group 8 parameters, which describe the outer lip movements, are considered. Group 8 parameters are expected, together with inner lip (group 2) and tongue (group 6) position parameters, to be the FAPs that contain the most useful information for ASR. The effectiveness of group 8 parameters is demonstrated in this work.

3.2. Gradient vector flow snake

A snake is an elastic curve defined by a set of control points [30, 31], and is used for finding important visual features, such as lines, edges, or contours. The snake parametric representation is given by

$$\mathbf{x}(s) = [x(s), y(s)], \quad s \in [0, 1], \tag{1}$$

where $x(s)$ and $y(s)$ are vertical and horizontal coordinates and s the normalized independent parameter. Snake deformation is controlled by the internal and external snake forces, $E_{\text{int}}(\mathbf{x}(s))$, $E_{\text{ext}}(\mathbf{x}(s))$, respectively. The snake moves through the image minimizing the functional

$$E = \int_0^1 \left(E_{\text{int}}(\mathbf{x}(s)) + E_{\text{ext}}(\mathbf{x}(s))\right) ds. \tag{2}$$

The internal snake force $E_{\text{int}}(\mathbf{x}(s))$ is defined as

$$E_{\text{int}}(\mathbf{x}(s)) = \frac{1}{2}\left(\alpha \cdot |\mathbf{x}'(s)|^2 + \beta \cdot |\mathbf{x}''(s)|^2\right), \tag{3}$$

where α and β are the parameters that control the tension and rigidity of the snake, while $\mathbf{x}'(s)$ and $\mathbf{x}''(s)$ denote the first and second derivatives of $\mathbf{x}(s)$ with respect to s. The external forces, $E_{\text{ext}}(\mathbf{x}(s))$, are derived from the image data. The commonly used external forces are defined either as

$$E_{\text{ext}}(\mathbf{x}(s)) = -|\nabla I|^2, \tag{4}$$

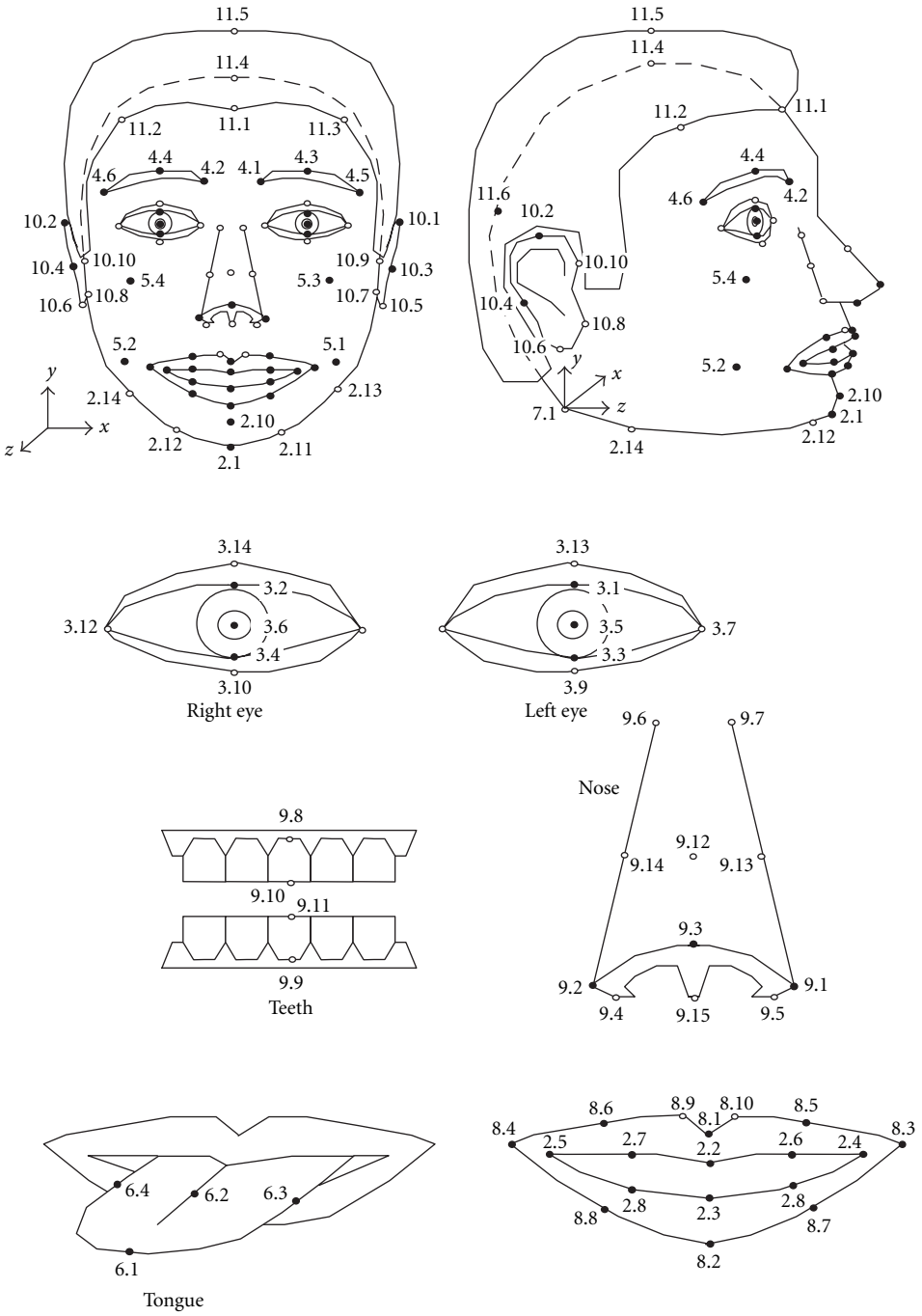


Figure 3: Facial animation parameters (FAPs) [24].

or after blurring the image, as

$$E_{\text{ext}}(\mathbf{x}(s)) = -\left|\nabla\left(G_\sigma * I\right)\right|^2, \tag{5}$$

where ∇ denotes the gradient operator, G_σ denotes a two-dimensional Gaussian function with standard deviation σ, and I denotes the intensity of a gray level image.

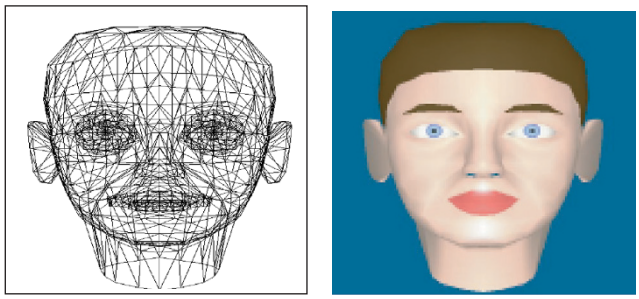
The gradient vector flow field [32, 33], defined as a vector field $\mathbf{v}(x, y) = (u(x, y), v(x, y))$, can be used as an external force. It is computed by minimizing the energy functional

$$\varepsilon = \iint \mu\left(u_x^2 + u_y^2 + v_x^2 + v_y^2\right) + |\nabla f|^2 \cdot |\mathbf{v} - \nabla f|^2 \, dx\, dy, \tag{6}$$

where f denotes an edge map, defined as $f = -E_{\text{ext}}(\mathbf{x}(s))$, and μ is a weighting factor which is determined based on the noise level in the image. The important property of the GVF is that when used as an external force, it increases the capture range of the snake algorithm. The initial snake for the

Table 1: Facial Animation Parameters [24].

Facial Animation Parameters (FAPs)		
Group	Description	Number of parameters
1	Visemes and expressions	2
2	Jaw, chin, inner lowerlip, cornerlips, midlip	16
3	Eyeballs, pupils, eyelids	12
4	Eyebrow	8
5	Cheeks	4
6	Tongue	5
7	Head rotation	3
8	Outer lip positions	10
9	Nose	4
10	Ears	4



(a) (b)

Figure 4: (a) The MPEG-4 mesh model; (b) the corresponding MPEG-4 texture model [29].

GVF snake algorithm can also cross the boundary and does not require prior knowledge of whether to shrink or expand, since the forces point toward the boundary from both sides of it [32, 33].

The deformation process of the snake is controlled by the iterative search for a local minimum of the energy functional (2). In order to obtain group 8 FAPs from the extracted mouth image, lip tracking, using the GVF snake [32, 33], was performed. An example of a gray level mouth image is shown in Figure 5a, the corresponding GVF in Figure 5b, and the resulting snake position in Figure 5c.

3.3. *Parabola templates*

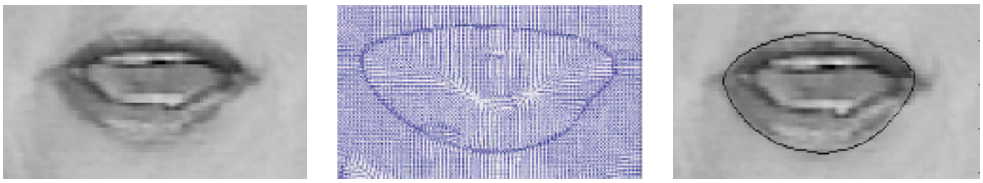
Based on our investigations we concluded that the snake algorithm which uses only the GVF as an external force is sensitive to random noise and certain salient features (such as, lip reflections) around the lips. Therefore, in some cases it does not achieve a close fit of the resulting snake to the outer lips, as will be demonstrated later. In order to improve the lip-tracking performance in these cases, two parabolas are fitted along the upper and lower lip by following the steps described below.

Edge detection is performed on every extracted mouth image using the Canny detector [34] and an edge map image is obtained (an example of a gray level mouth image is shown in Figure 6a, and the corresponding edge map image in Figure 6b). In order to obtain sets of points on the upper and lower lip, the edge map image is scanned from the bottom and the top, column-wise, keeping only the first encountered nonzero pixels (shown in Figure 6c). Parabolas are fitted through each of the obtained sets of points, as shown in Figure 6d.

The noise present in the mouth image and the texture of the area around the mouth in some cases results in unwanted edges in the edge map, which may cause inaccurate fitting of the parabolas to the outer lips. Examples of such unwanted edges are shown in Figures 7d and 7e, which are derived from the images shown in Figures 7a and 7b. Clearly, attempting to fit parabolas to the sets of edges shown in Figures 7d and 7e would result in undesirable outcomes. We resolved these cases by taking the following steps.

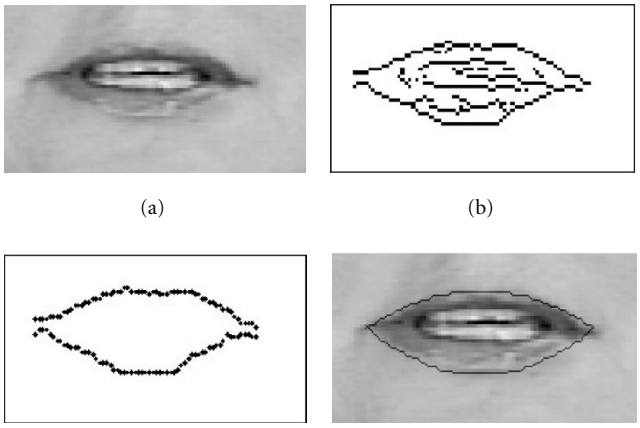
(i) The scan area was constrained by two half ellipses (shown in Figure 7c), to eliminate the pixels in the corners of the mouth image which may not be part of the actual mouth. Since there is a lot of texture inside the mouth area, the edge detection step will result in a large number of edges in that area, as can be seen in Figures 6b, 7d, 7e, and 8b. Therefore, this fact was used to calculate in two steps the medians of the horizontal and vertical coordinates of the edge map pixels and position the scan area. In the first step, medians were calculated based on all edges in the edge map image, while in the second step medians were obtained after outliers whose contribution to the variance is much larger than average were removed. The medians obtained in the second step were used to position the scan area and eliminate the edges outside it (examples are shown in Figures 7f and 7g).

(ii) In order to resolve the cases in which there are unwanted edges inside the scan area (Figure 7g), we calculated in two steps the median (m) and the variance (σ) of the vertical coordinate of the nonzero pixels, inside the scan area, and kept just the ones inside the area of $(m - l \cdot \sigma)$, and $(m + n \cdot \sigma)$



(a) (b) (c)

Figure 5: (a) Extracted mouth image; (b) GVF, external force field; and (c) the final snake position.



(a) (b)

(c) (d)

Figure 6: (a) Extracted mouth image; (b) corresponding edge map image; (c) sets of points on upper and lower lip boundaries; and (d) fitted parabolas.

(the values of l and n were experimentally determined to be equal to 4.75 and 4.5, respectively). The edge map obtained after step (ii) was applied to Figure 7g is shown in Figure 7h.

(iii) In order to eliminate unwanted edges, which are close to the lip contour, we performed the fitting of the parabolas in two steps (examples are shown in Figure 8), by eliminating the points that are the farthest from the parabolas obtained in the first step. Figure 8a shows the gray level mouth image, Figure 8b shows the edge map image obtained after steps (i) and (ii) were taken, Figure 8c shows the sets of points through which parabolas were fitted. The darker parabolas in Figure 8d resulted from the application of the first step of the algorithm, while the lighter parabolas in the same figures resulted from the second step of the algorithm, after the farthest points (indicated as darker points) were eliminated. The fitted parabolas after the second step are shown in Figure 8e. Afterwards the image consisting of the two final parabolas was blurred and the parabola external force, v_{parabola}, was obtained with a use of the gradient operator. It was added to the GVF external force, v_{GVF}, to obtain the final external force, v_{final}, by appropriately weighting the two external forces, that is,

$$v_{\text{final}} = v_{\text{GVF}} + w \cdot v_{\text{parabola}}. \tag{7}$$

The value of $w = 1.5$ proved to provide consistently better results. The final external force was used for running the snake algorithm. For each frame (except for the first frame of each sequence), a resulting snake from the previous frame was expanded and used as an initial snake in order to decrease the number of iterations. Accurate initialization of the snake is not necessary since the GVF has a large capture area and can capture a snake from either side of the outer lips boundary. Both components of the final external force for the snake are obtained from the gradient information. This intensity gradient information, extracted twice in two different ways is, as can be seen in Figure 9, in most cases complementary in part of the boundary. For example, in Figure 9a2 the GVF external force is inaccurate at the bottom of the lip (due to specular reflection, see Figure 9a1), while the parabolas in Figure 9a3 (which were also driven by gradient information) are quite accurate, resulting in a satisfactory overall result in Figure 9a4. A similar example is represented by Figures 9b1, 9b2, 9b3, and 9b4. The reverse is true in Figures 9d1, 9d2, 9d3, and 9d4. This complementarity of the information is further represented by Figures 9c1, 9c2, 9c3, and 9c4 where both components of the external force are in error, but not at the same place. Examples in which both methods give sufficient information are shown in Figure 10.

All FAPs are expressed in terms of facial animation parameter units (FAPU), shown in Figure 11. These units are normalized by important facial feature distances in order to give an accurate and consistent representation. The FAPs from group 8 that describe the outer lip contours are represented with the use of two FAPUs, mouth-width separation and mouth-nose separation. Each of these two distances is normalized to 1024. In the experiments with the Bernstein database, the first frame face in each sequence was used as a neutral face for that sequence. The resulting lip contour of each frame, and the resulting lip contour of the neutral (first) frame were compared in order to generate FAPs, by aligning the current lip contour with the neutral frame reference lip contour, calculating the movement, and normalizing the distance between corresponding locations. As a result we obtained for each frame, at time t, a visual feature vector f_t consisting of 10 outer lip FAPs,

$$f_t = [\text{FAP8.1.}, \text{FAP8.2.}, \ldots, \text{FAP8.10.}]^T. \tag{8}$$

The above described FAP extraction algorithm was applied

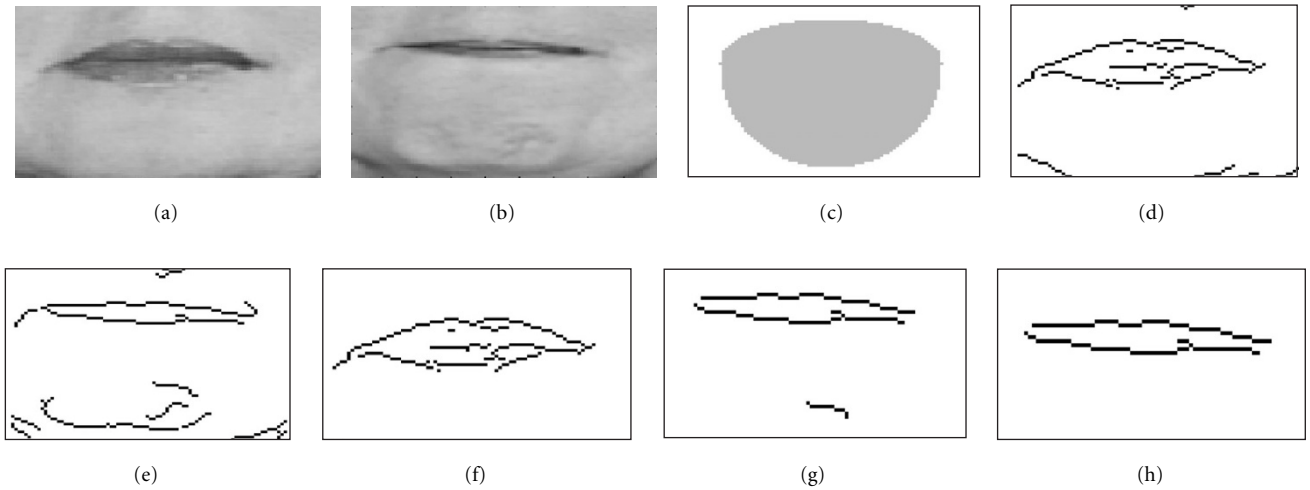


FIGURE 7: (a), (b) Original mouth images; (c) the scan area; (d), (e) edge map images; (f), (g) the edges inside the scan area; (h) the edge map image after the step (ii) was taken.

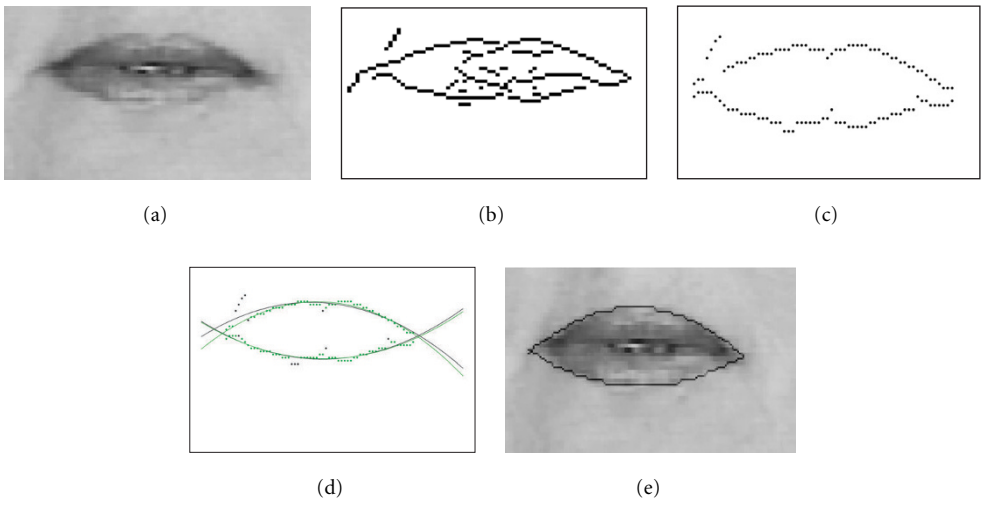


FIGURE 8: (a) The mouth image; (b) the edge map image after steps (i) and (ii) were taken; (c) points used for parabola fitting; (d) two-step parabola fitting, (darker points were not used in the second step), (e) resulting fitted parabolas.

on every video frame of the female speaker in the Bernstein database and the visual feature vectors, with elements FAPs, were obtained.

Through visual evaluation of the FAP extraction results, we observed that the extracted parameters produced a natural movement of the MPEG-4 decoder that synchronized well with the audio (examples of facial animations, obtained by running the MPEG-4 decoder with extracted FAPs, are shown in Figure 12). Therefore, we concluded that the developed algorithm performed very well without any previous use of hand labeling or computationally extensive training. However, the ultimate success in extracting the FAPs needs to be evaluated in terms of the increase in performance of the audio-visual ASR system.

4. VISUAL FEATURE DIMENSIONALITY REDUCTION

In order to decrease the dimensionality of the visual feature vector without losing the relevant speechreading information, PCA, or Discrete Karhunen-Loeve expansion, was performed on the FAP vectors. FAPs obtained from the training data were used to obtain the 10×1 mean FAP visual feature vector $\bar{f}_t$ and the 10×10 covariance matrix. The eigenvalues and eigenvectors of the covariance matrix were computed and sorted according to decreasing values of the eigenvalues. The first K ($K = 1, 2, 6$) eigenvectors corresponding to the largest K eigenvalues were chosen, since they define the projection of the 10-dimensional data onto a K-dimensional space, that best represents the data in the least-squares sense.

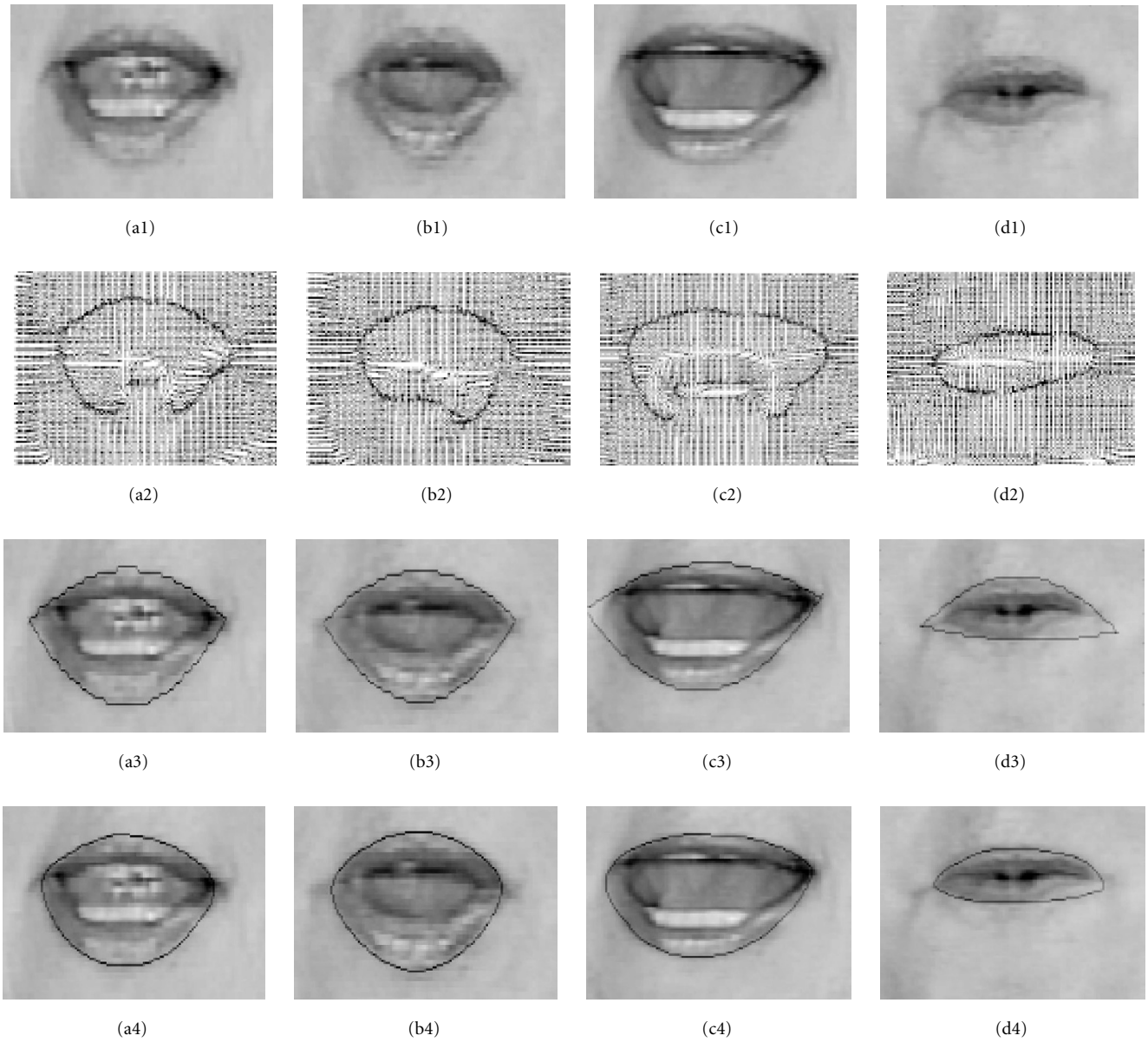


FIGURE 9: (a1)–(d1) Original mouth images; (a2)–(d2) GVFs; (a3)–(d3) fitted parabolas; and (a4)–(d4) final results, for cases when GVF external force does not give enough information (a), (b), and when the parabola templates do not give good results when applied individually (c), (d).

The extracted FAP visual feature vectors, of dimension 10, which describe the outer lip shape, were then projected onto the eigenspace defined by the first K eigenvectors,

$$f_t = \bar{f}_t + E \cdot o_t^v, \tag{9}$$

where,

$$E = [e_1, e_2, \ldots, e_k] \tag{10}$$

is the matrix of K eigenvectors, $e_1, e_2, \ldots, e_K$, which correspond to the K largest eigenvalues, and o_t^v the $K \times 1$ vector, that denotes the projection weights for each eigenvector in E. The projection weights, o_t^v, are used as visual features from this point on. In our experiments the models with different number of eigenvectors were used to describe the outer lip shape. The first 6 eigenvectors describe 99.6% of the total variance described by all 10 eigenvector, the first 2 describe 93%, and the first eigenvector only describes 81% of the total variance. The mean outer lips shape and the outer lips shape described by the variation of the projection weights o_t^v (by ± 2 standard deviations) for the first 4 eigenvectors are shown in Figure 13. They are obtained by running the MPEG-4 decoder for the visual feature vectors (9). As can be clearly seen in Figure 13, the first eigenvector mostly describes the

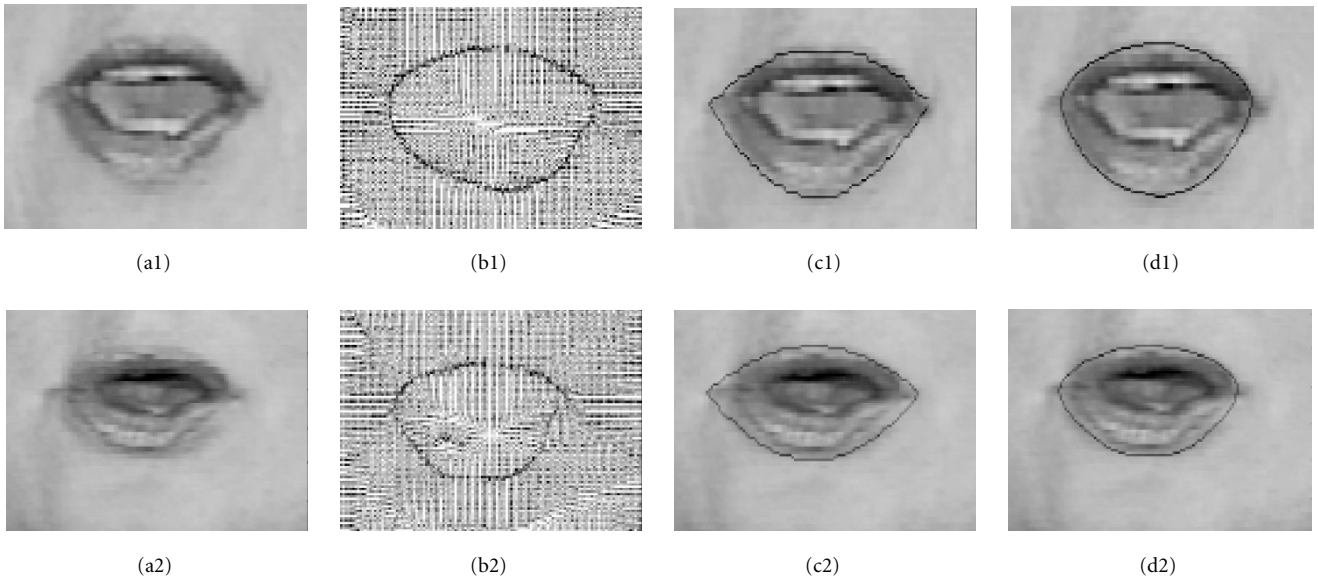


Figure 10: (a1), (a2) Original images; (b1), (b2) GVF external force; (c1), (c2) fitted parabolas; and (d1), (d2) final results, for the cases when both methods perform well.

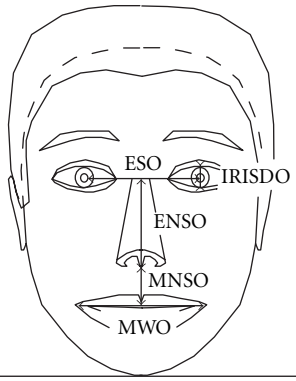
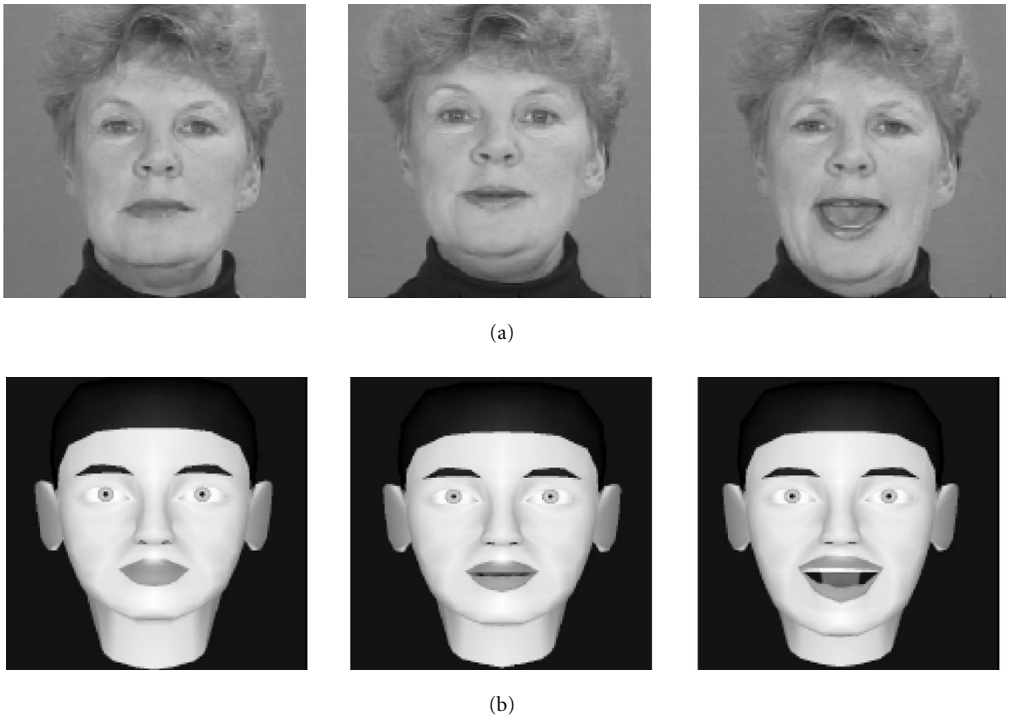
	Description	FAPU value
	IRIS diameter (by definition it is equal to the distance between upper and lower eyelid) in neutral face	IRISD = IRISD0/1024
	Eye separation	ES = ES0/1024
	Eye-nose separation	ENS = ENS0/1024
	Mouth-nose separation	MNS = MNS0/1024
	Mouth-width separation	MW = MW0/1024
	Angular unit	AU = 10^{-5} rad

Figure 11: Facial animation parameter units (FAPU) [24].

position of the lower lip, the second eigenvector mostly describes the position of the upper lip, while the third and the forth eigenvector describe the asymmetries in the lip shape. It is shown in the following paragraphs that considerable ASR improvements for both noisy and clean speech can be achieved even when using only one-dimensional ($K = 1$) visual feature vectors.

5. AUDIO-VISUAL INTEGRATION

The audio and visual speech information can be combined using several techniques. One such technique (early integration) is to just concatenate audio and visual features, form bigger feature vectors, and then train a single-stream HMM system on the concatenated vectors. Audio and visual feature streams should be synchronized before the concatenation is performed. Although using concatenated audio-visual features can improve ASR performance over audio-only ASR, it does not allow for the modeling of the reliability of the audio and visual streams. It therefore cannot take advantage of the information that might be available about the acoustic noise in the environment, which affects audio features, or the visual noise, which affects the quality of the visual features. A second approach is to model audio and visual features as separate feature streams (late integration), by use of multistream HMMs. Their log-likelihoods can be combined using the weights that capture the reliability of each particular stream. The recombination of log-likelihoods can be performed at different levels of integration, such as state, phone, word, and utterance [13].



(a)

(b)

Figure 12: (a) Original images; (b) MPEG-4 facial animations run by extracted FAPs.

The two streams, audio and visual, are combined in the approach we developed as shown in Figure 14. The mel-frequency cepstral coefficients (MFCC), widely used in speech processing, were used as audio features, while the projections of the FAPs onto the eigenspace defined by (9), for different values of the dimensionality of the eigenvector space K, were used as visual features. Since MFCCs were obtained at a rate of 90 Hz, while FAPs at a rate of 30 Hz, FAPs were interpolated in order to obtain synchronized data.

5.1. The single-stream HMM

In this approach the audio-visual feature observation vector (o_t) is formed by appending the visual observations vector (o_t^v) to the audio observations vector (o_t^a), that is

$$o_t = \left[o_t^{a^T} \quad o_t^{v^T} \right]^T. \tag{11}$$

The newly obtained joint features were used to train a single-stream HMM model, with state emission probabilities given by

$$b_j(o_t) = \sum_{m=1}^{M} c_{jm} N(o_t; \mu_{jm}, \Sigma_{jm}). \tag{12}$$

In (12) subscript j denotes a state of a context-dependent model, M denotes the number of mixtures, c_{jm} denotes the weight of the mth mixture component, and N is a multivariate Gaussian with mean vector μ_{jm} and diagonal covariance matrix Σ_{jm}. The sum of mixture weights c_{jm} is equal to 1.

5.2. The multistream HMM

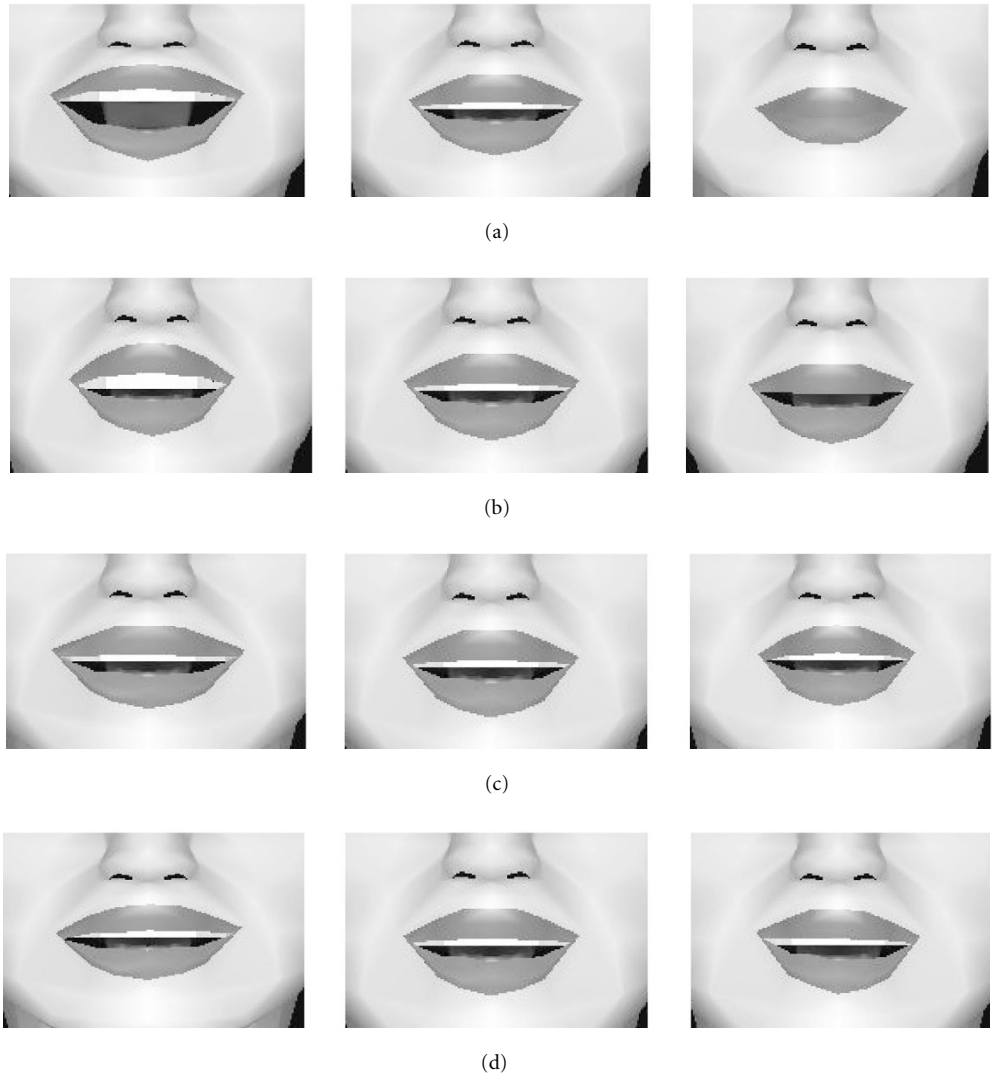
We also used a multistream HMM, and a state log-likelihood recombination method, to perform audio-visual integration, since it allows for easy modeling of the reliability of the audio and visual streams. Two data streams were used to separately model audio and visual information, and audio and visual stream log-likelihoods were combined at the state level to recognize audio-visual speech [35, 36]. The state emission probability of the multistream HMM, $b_j(o_t)$, is given by

$$b_j(o_t) = \prod_{s \in \{a,v\}} \left[\sum_{m=1}^{M_s} c_{jsm} N(o_t^s; \mu_{jsm}, \Sigma_{jsm}) \right]^{\gamma_s}. \tag{13}$$

In (13) subscript j denotes context-dependent state, M_s denotes the number of mixtures in a stream, c_{jsm} denotes the weight of the mth mixture of the stream s, and N is a multivariate Gaussian with mean vector μ_{jsm} and diagonal covariance matrix Σ_{jsm}. The nonnegative stream weights are denoted by γ_s, and they depend on the modality s. We assumed that stream weights satisfy,

$$\gamma_a + \gamma_v = 1. \tag{14}$$

In order to perform multistream HMM training, stream weights must be chosen *a-priori*. They were roughly optimized by minimizing the word error rate (WER) on a development data set.



(a)

(b)

(c)

(d)

Figure 13: The middle column shows the mean lip shape, while the left and right columns show the lip shapes obtained by the variation of the projection weights corresponding to the first (a), second (b), third (c), and fourth (d) eigenvector by 2 standard deviations (st. dev.).

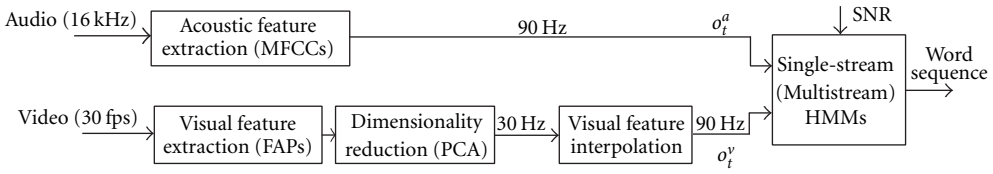


Figure 14: Audio-visual system for ASR.

6. SPEECH RECOGNITION EXPERIMENTS

The baseline speech recognition system was developed using the HTK toolkit version 2.2 [35]. A training procedure similar to the one described in the HTK reference manual [35, Chapter 3] was used. All the experiments performed used only the part of the Bernstein database with the female talker. We modeled 39 phonemes, silence "sil" and short pause "sp." All HMMs were left-right, with 3 states, except the "sp" model which had only one state. Context dependent phoneme models (triphones) were used since they account for variations of the observed features caused by coarticulation effects. They were used as speech units, and their state emission probabilities were modeled by Gaussian mixture densities. Decision-tree based clustering was performed in order to tie states of context-dependent phone models,

Table 2: Parameters that describe the dependence between audio stream weights and SNR for different visual features dimensionality K.

Visual features dimensionality	a	b
$K = 1$	0.0059	0.5089
$K = 2$	0.0064	0.5011
$K = 6$	0.0062	0.5224

and therefore perform reliable estimation of models parameters. The Baum-Welch re-estimation algorithm was used for training the HMM models. Iterative mixture splitting and retraining were performed to obtain the final 9-mixture component context dependent models, which were used for testing. Approximately 80% of the data was used for training, 18% for testing, and 2% as a development set for obtaining roughly optimized stream weights, word insertion penalty and the grammar scale factor. A bi-gram language model was created based on the transcriptions of the training data set. Recognition was performed using the Viterbi decoding algorithm, with the bi-gram language model. The same training and testing procedure was used for both audio-only and audio-visual automatic speech recognition experiments. To test the algorithm over a wide range of SNRs, white Gaussian noise was added to the audio signals. The experiments for SNRs from 0 dB to 30 dB were performed. Both audio-visual integration methods described above, simple concatenation and single-stream HMMs, and multistream HMMs were applied. Separate training and testing procedures were performed for different number of principal components ($K = 1, 2, 6$) used as visual feature vectors. The experiments with the clean original speech, using both audio-visual integration methods, were also performed.

In all experiments that used multistream HMMs, the stream weights were estimated, by roughly minimizing the WER on the development data set, which was exposed to the same noise as the testing data set. The experiments for multiple weights from 0.45 to 0.95 with a step of 0.025 were run, and then the weights, for which the best results were obtained, were chosen. The estimation of the stream weights was performed for each different dimensionality of the visual data, K. The stream weights obtained were roughly linearly related to the SNR according to

$$\begin{aligned} \gamma_a &= a \cdot \mathrm{SNR}(\,\mathrm{dB}) + b, \\ \gamma_v &= 1 - \gamma_a, \end{aligned} \tag{15}$$

where the parameters a and b depend on the dimensionality K, and their values are shown in Table 2.

6.1. Audio-only speech recognition experiments

Twelve MFCCs and signal energy with first and second order derivatives were used as audio features. HMMs were trained using the procedures described in the previous section. Training and testing were performed on audio corrupted by stationary Gaussian white noise over a wide range of SNRs (0–30 dB), and on clean audio. All audio-only results were obtained using HMMs trained in matched conditions, by corrupting the training data used by the same level of stationary Gaussian white noise, as used for corrupting the testing data. This approach was used in order to accurately measure the influence of visual information on ASR performance, since it results in better performance than systems trained on unmatched data, which use noise compensation techniques [36]. Speech recognition results obtained are summarized in Figure 15. All the results are reported based on the NIST scoring standard [35]. It can be observed that the recognition performance is severely affected by additive noise.

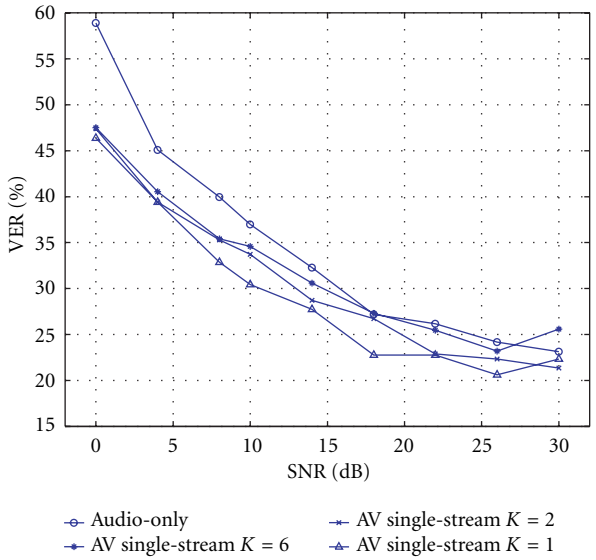


Figure 15: Audio-only and audio-visual single-stream system WERs vs. SNR.

6.2. Audio-visual speech recognition experiments

Projections of the ten FAPs, describing the outer lip movement, onto the eigenspace defined by (10), and their first and second order derivatives were used as visual features. The first and second order derivatives were included, since the dynamics of the movement of the lips contains important speechreading information. The single-stream and the multistream HMMs were trained using the procedure described in previous sections.

6.2.1 Experiments with single-stream HMMs

A single-stream HMM, with a concatenated vector of 12 MFCCs, signal energy, K visual parameters, and their first and second derivatives was trained. Training and testing were performed on the audio corrupted by stationary Gaussian white noise over a wide range of SNRs (0–30 dB). Experiments were performed for different values of the dimensionality, K ($K = 1, 2, 6$), of the visual feature vectors. The results obtained are shown in Figure 15 together with audio-only HMM system results.

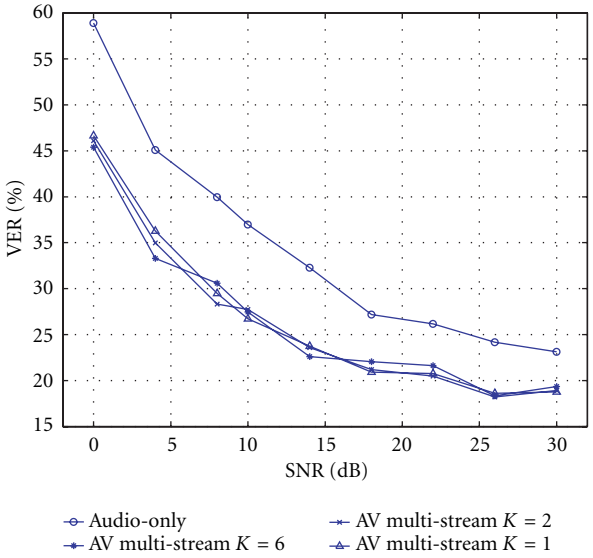


Figure 16: Audio-only and audio-visual multistream system WERs vs. SNR.

It can be seen from Figure 15 that the audio-visual system performed much better than the audio-only system for almost all SNRs tested. It is also important to point out that performance of the systems that use 6-dimensional visual features, was better than the performance of the audio-only ASR system only for SNRs smaller than 30 dB, while for the SNR of 30 dB, they performed worse than the audio-only ASR system, due to the higher dimensionality of the visual feature vectors resulting in inadequate modeling. On the other hand, due to the facts that the first 2 eigenvectors of the covariance matrix describe 93% of the total variance described along all 10 eigenvector, and that the first eigenvector only describes 81% of the total variance, systems that used 2- and 1-dimensional visual features performed better than the audio-only system for all values of SNRs, for which tests were run.

6.2.2 Experiments with multistream HMMs

The multistream HMMs experiments were performed for different values of the dimensionality of the visual feature vectors, K ($K = 1, 2, 6$). The stream weights, a and b in (15), that were obtained by minimizing the WER on the development set are shown in Table 2. The results obtained over a wide range of SNRs are shown in Figure 16, where they are compared with audio-only ASR results.

As can be clearly seen, the proposed multistream HMM audio-visual ASR system performs considerably better than audio-only ASR for all dimensionalities of the visual feature vectors, and for all values of SNR. At the same time the multistream HMM system outperforms the single-stream system for all values of K and for all values of SNR. The results confirm the belief that modeling the reliability of the audio and visual streams improves the ASR performance over the single-stream HMM performance. The relative reduction of WER achieved, compared to the audio-only WER, ranges from 20% for the noisy audio with SNR of 30 dB to 23% for SNR of 0 dB. It is important to point out that the considerable performance improvement was achieved even in the case when only one-dimensional visual features were used. Clearly, the fact that the system has similar performance for different sizes of visual features ($K = 1, 2, 6$) is due to the trade-off between the number of parameters that have to be estimated and the amount of the visual speechreading information contained in the visual features. Results may be further improved by better adjusting the audio and visual stream weights [37].

Table 3: Audio-only and audio-visual system performance under clean audio conditions.

Audio-only and audio-visual system performance under clean audio conditions		WER [%]	Audio-stream weight
Audio-only system		22.19	
Audio-visual system (Single-stream method)	$K = 1$	21.48	
	$K = 2$	21.34	
	$K = 6$	24.47	
Audio-visual system (Multistream method)	$K = 1$	18.21	0.75
	$K = 2$	18.07	0.7
	$K = 6$	18.16	0.85

6.2.3 Experiments with clean speech

Experiments were also performed with clean speech by using both audio-visual integration methods and visual feature vectors of different dimensionalities. The results obtained are shown in Table 3. Improved ASR performance in clean speech was achieved when the single-stream HMMs system (except for $K = 6$) was used, and considerable improvement in the ASR performance was achieved when the multistream HMMs were used (for all values of K). The maximum relative reduction of WER over the audio-only ASR system WER in clean speech was 19% and was achieved when visual feature vectors of dimension 2 ($K = 2$) were used, and when the audio stream weight was equal to 0.7 ($\gamma_a = 0.7$).

7. CONCLUSIONS

We described a robust and automatic FAP extraction system that we have implemented, using GVF snake and parabolic templates. Our method does not require any previous use of hand labeling or computationally extensive training.

We described two audio-visual integration systems that we used in our experiments, the single-stream and multistream systems, and tested their performance on a relatively large audio-visual database, over a wide range of noise levels, and for different values of the dimensionality of the visual feature vectors. The performance of the single-stream HMM system was higher when only 1- or 2-dimensional visual features were used due to better modeling of the system resulting from the lower dimensionality of the visual feature

vectors. Although we used only K-dimensional visual feature vectors ($K = 1, 2, 6$) and no information about the mouth area intensities, we still obtained considerable improvement in ASR performance for all noise levels tested and for all dimensionalities K when the multistream approach was used. The training and testing of the audio-visual system under clean audio conditions was also performed, and considerable improvement in performance over audio-only system performance was obtained. The multistream method achieves a significant WER relative reduction of 19% compared to audio-only system. The improvement in ASR performance that can be obtained by exploiting the visual speech information contained in group 8 FAPs was determined. Therefore, the usefulness of the FAPs in the audio-visual MPEG-4 related applications, like automatic transcription of speech to text, was shown.

As a second step, we plan to extract all the remaining FAPs that contain important speechreading information (FAPs describing inner lip shape and tongue position), determine how much speechreading information they contain, and keep only the FAPs with significant and independent speechreading information [26], in order to lower the visual feature dimensionality and improve the audio-visual ASR performance. We also plan to perform experiments on multispeaker large vocabulary databases, like [13], when they become publicly available.

REFERENCES

[1] Q. Summerfield, "Lipreading and audio-visual speech perception," *Philos. Trans. Roy. Soc. London Ser. B*, vol. 335, pp. 71–78, 1992.

[2] K. W. Grant and L. D. Braida, "Evaluating the articulation index for auditory visual input," *Journal of the Acoustical Society of America*, vol. 89, no. 6, pp. 2952–2960, 1991.

[3] J. J. Williams, *Speech-to-video conversion for individuals with impaired hearing*, Ph.D. thesis, Northwestern University, Evanston, Ill, USA, June 2000.

[4] J. J. Williams and A. K. Katsaggelos, "An HMM-based speech-to-video synthesizer," *IEEE Trans. on Neural Networks*, vol. 13, no. 4, pp. 900–915, 2002, Special Issue on Intelligent Multimedia.

[5] Q. Summerfield, "Some preliminaries to a comprehensive account of audio-visual speech perception," in *Hearing by Eye: The Psychology of Lip-Reading*, B. Dodd and R. Campbell, Eds., pp. 97–113, Lawrence Erlbaum Associates, Hillside, Minn, USA, 1987.

[6] R. Lippman, "Speech recognition by machines and humans," *Speech Communication*, vol. 22, no. 1, pp. 1–15, 1997.

[7] Y. Gong, "Speech recognition in noisy environments: A survey," *Speech Communication*, vol. 16, no. 3, pp. 261–291, 1995.

[8] E. Petajan, "Automatic lipreading to enhance speech recognition," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 40–47, San Francisco, Calif, USA, 1985.

[9] E. Petajan, B. Bischoff, D. Bodoff, and N. M. Brooke, "An improved automatic lipreading system to enhance speech recognition," in *CHI-88*, pp. 19–25, ACM, Washington, DC, USA, 1988.

[10] A. J. Goldschen, O. N. Garcia, and E. Petajan, "Continuous optical automatic speech recognition by lipreading," in *Proc. IEEE 28th Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, Calif, USA, October 1994.

[11] D. G. Stork and M. E. Hennecke, Eds., *Speechreading by Man and Machine*, Springer-Verlag, New York, NY, USA, 1996.

[12] E. Petajan and H. P. Graf, "Robust face feature analysis for automatic speechreading and character animation," in *Proc. 2nd Int. Conf. Automatic Face and Gesture Recognition*, pp. 357–362, Killington, Vt, USA, 1996.

[13] C. Neti, G. Potamianos, J. Luettin, et al., "Audio-visual speech recognition," Tech. Rep., Johns Hopkins University, Baltimore, Md, USA, October 2000.

[14] G. Potamianos, J. Luettin, and C. Neti, "Hierarchical discriminant features for audio-visual LVCSR," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, vol. 1, pp. 165–168, Salt Lake City, Utah, USA, 2001.

[15] G. Potamianos and H. P. Graph, "Discriminative training of HMM stream exponents for audio-visual speech recognition," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, vol. 6, pp. 3733–3736, Seattle, Wash, USA, 1998.

[16] C. Bregler and Y. Conig, ""Eigenlips" for robust speech recognition," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, pp. 669–672, Adelaide, Australia, 1994.

[17] J. Luettin and N. A. Thacker, "Speechreading using probabilistic models," *Computer Vision and Image Understanding*, vol. 65, no. 2, pp. 163–178, 1997.

[18] S. Dupont and J. Luettin, "Audio-visual speech modeling for continuous speech recognition," *IEEE Trans. Multimedia*, vol. 2, no. 3, pp. 141–151, 2000.

[19] C. Neti, G. Iyengar, G. Potamianos, A. Senior, and B. Maison, "Perceptual interfaces for information interaction: Joint processing of audio and visual information for human-computer interaction," in *Proc. International Conference on Spoken Language Processing*, vol. III, pp. 11–14, Beijing, China, October 2000.

[20] R. Kaucic, B. Dalton, and A. Blake, "Real-time lip tracking for audio-visual speech recognition applications," in *Proc. European Conference on Computer Vision*, vol. 2, pp. 376–387, Cambridge, UK, 1996.

[21] G. Chiou and J.-N. Hwang, "Lipreading from color video," *IEEE Trans. Image Processing*, vol. 6, no. 8, pp. 1192–1195, 1997.

[22] M. T. Chan, Y. Zhang, and T. S. Huang, "Real-time lip tracking and bimodal continuous speech recognition," in *Proc. IEEE 2nd Workshop on Multimedia Signal Processing*, pp. 65–70, Redondo Beach, Los Angeles, Calif, USA, December 1998.

[23] A. L. Yuille, P. W. Hallinan, and D. S. Cohen, "Feature extraction from faces using deformable templates," *International Journal of Computer Vision*, vol. 8, no. 2, pp. 99–111, 1992.

[24] Text for ISO/IEC FDIS 14496-2 Visual, ISO/IEC JTC1/SC29/WG11 N2502, November 1998.

[25] Text for ISO/IEC FDIS 14496-1 Systems, ISO/IEC JTC1/SC29/WG11 N2502, November 1998.

[26] F. Lavagetto and R. Pockaj, "An efficient use of MPEG-4 FAP interpolation for facial animation at 70 bits/frame," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 11, no. 10, pp. 1085–1097, 2001.

[27] E. Petajan, "Approaches to visual speech processing based on the MPEG-4 face animation standard," in *Proc. IEEE International Conference on Multimedia and Expo(I)*, pp. 575–587, New York, NY, USA, 30 July–2 August 2000.

[28] L. Bernstein and S. Eberhardt, "Johns Hopkins lipreading corpus I-II," Tech. Rep., Johns Hopkins University, Baltimore, Md, USA, 1986.

[29] G. Abrantes, *FACE-Facial Animation System, version 3.3.1*, Instituto Superior Técnico, (c), Lisbon, Portugal, 1997–1998.

[30] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988.

[31] L. D. Cohen and I. Cohen, "Finite element methods for active contour models and balloons for 2-D and 3-D images," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 15, no. 11, pp. 1131–1147, 1993.

[32] C. Xu and J. L. Prince, "Gradient vector flow: A new external force for snakes," in *Proc. IEEE International Conf. on Computer Vision and Pattern Recognition*, pp. 66–71, Puerto Rico, June 1997.

[33] C. Xu and J. L. Prince, "Snakes, shapes, and gradient vector flow," *IEEE Trans. Image Processing*, vol. 7, no. 3, pp. 359–369, 1998.

[34] R. Jain, R. Kasturi, and B. G. Schunck, *Machine Vision*, McGraw-Hill, New York, NY, USA, 1995.

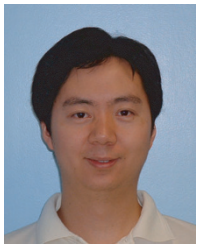
[35] S. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev, and P. Woodland, *The HTK Book*, Entropic, Cambridge, UK, 1999.

[36] L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1993.

[37] H. Glotin, D. Vergyri, C. Neti, G. Potamianos, and J. Luettin, "Weighting schemes for audio-visual fusion in speech recognition," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, vol. 1, pp. 165–168, Salt Lake City, Utah, USA, 2001.

Petar S. Aleksic received his B.S. degree in electrical engineering at the Belgrade University, Yugoslavia, in 1999. He received his M.S. degree in electrical engineering at Northwestern University in 2001. Currently, he is pursuing his Ph.D. degree in electrical engineering at Northwestern University. He has been a member of the Image and Video Processing Lab (IVPL) at Northwestern University since 1999. His current research interests include audio-visual speech recognition, multimedia communications, computer vision, and pattern recognition.

Jay J. Williams received the B.S. degree in electrical engineering at Howard University, Washington, DC, in 1993. He received his M.S. and Ph.D. degrees, both in Electrical Engineering, at Northwestern University, Evanston, IL, in 1996 and 2000, respectively. He is now with Ingenient Technologies, Inc., Chicago, IL. His current research interests include multimedia processing, image and video signal processing, computer vision, and audio-visual interaction.



Zhilin Wu received his B.S. degree in electrical engineering at Peking University, Beijing, China, in 1994 and M.S. degree in electrical engineering from Tsinghua University, Beijing, China, in 1997. He is currently a Ph.D. student in the Image and Video Processing Lab (IVPL) at Northwestern University. His current research interests include lip tracking, MPEG-4 facial animation, speech driven animation and audio-visual signal processing.

Aggelos K. Katsaggelos received the Diploma degree in electrical and mechanical engineering from Aristotelian University of Thessaloniki, Greece, in 1979 and the M.S. and Ph.D. degrees, both in electrical engineering from the Georgia Institute of Technology, in 1981 and 1985, respectively. In 1985 he joined the Department of ECE at Northwestern University, where he is currently professor, holding the Ameritech Chair of Information Technology. He is also the Director of the Motorola Center for Telecommunications, a member of the Associate Staff, Department of Medicine, at Evanston Hospital. He is editor-in-chief of the *IEEE Signal Processing Magazine*, a member of the Publication Board of the IEEE Signal Processing Society, the IEEE TAB Magazine Committee, the Publication Board of the *IEEE Proceedings*, and various Technical Comittees. Dr. Katsaggelos is the editor of "Digital Image Restoration" (Springer-Verlag, Heidelberg, 1991), co-author of "Rate-Distortion Based Video Compression" (Kluwer Academic Publishers, 1997), and co-editor of "Recovery Techniques for Image and Video Compression and Transmission," (Kluwer Academic Publishers, 1998). He is the coinventor of eight international patents, a Fellow of the IEEE (1998), and the recipient of the IEEE Third Millennium Medal (2000), the IEEE Signal Processing Society Meritorious Service Award (2001), and an IEEE Signal Processing Society Best Paper Award (2001).