Support Vector Machine Based Phoneme Segmentation for Lip Synch Application

Kunyoung Lee* · Hanseok Ko*

ABSTRACT

In this paper, we develop a real time lip-synch system that activates 2-D avatar's lip motion in synch with an incoming speech utterance. To realize the "real time" operation of the system, we contain the processing time by invoking merge and split procedures performing coarse-to-fine phoneme classification. At each stage of phoneme classification, we apply the support vector machine (SVM) to reduce the computational load while retraining the desired accuracy. The coarse-to-fine phoneme classification is accomplished via two stages of feature extraction: first, each speech frame is acoustically analyzed for 3 classes of lip opening using Mel Frequency Cepstral Coefficients (MFCC) as a feature; secondly, each frame is further refined in classification for detailed lip shape using formant information. We implemented the system with 2-D lip animation that shows the effectiveness of the proposed two-stage procedure in accomplishing a real-time lip-synch task. It was observed that the method of using phoneme merging and SVM achieved about twice faster speed in recognition than the method employing the Hidden Markov Model (HMM). A typical latency time per a single frame observed for our method was in the order of 18.22 milliseconds while an HMM method applied under identical conditions resulted about 30.67 milliseconds.

Keywords: SVM, lip-synch

1. Introduction

"Lip-synch" system refers to a system realizing 2-D avatar's lip-animation synchronized in motion with incoming speech utterances. A "real-time" lip-synch system intends to perform the avatar's lip motion synchronized with the incoming speech simultaneously or in real time.

Although many applications can be envisioned for a real-time lip-synch system, developing such a system is not trivial. There are three important but difficult steps in realizing a real-time lip-synch system: (1) real-time continuous phoneme recognition, (2) lip shape refinement and (3) real time avatar model generation. In this paper, we focused the first two issues for modeling and extracting features suitable for real-time phoneme classification.

* Dept. of Electronics and Computer Engineering, Korea University

To achieve real-time continuous phoneme recognition (or classification), we employ the support vector machine (SVM) instead of the more traditional Hidden Markov Model (HMM). Although HMM is widely used in general-purpose speech recognition systems[1][2], our implementation of an HMM has typically shown significant delay time due to its inherent structure with multiple state-models making it not practical in a real-time system. In the proposed real-time lip-synch applications, the desired speech unit for recognition is a phoneme which is usually very short in duration and has no clear indication of a boundary between two neighboring phonemes. The lack of ability to sort out the incoming speech in some predefined boundaries forces the recognition to take place at the frame unit much smaller than the size of phonemes. Pattern matching based on the feature over each frame can be used to accomplish phoneme recognition with the assumption that the signal in one frame is stable so that a unique phoneme can be established over the frame. Among such simple pattern matching methods, we propose to employ SVM for classifying phonemes into 3 classes in the first stage of speech pattern analysis. The first stage classification is based on the MFCC as the feature. The result is then fed into the second stage of the pattern analysis for further refinement in capturing the detailed lip variations from each class by the use of the formant locations. While the vowels and voiced consonants are the main drivers for lip motion, we focus on a real-time system that takes any speech utterance to capture the acoustic models and map them into the lip motioning visemes. In short, the first stage is to generate acoustical classifications capturing the 3 types of baseline lip openings and the second stage refines the 3 acoustical classifications to the specific lip shape by detailing the lip width and height with respect to the baseline openings.

Fig. 1 shows a block diagram of the proposed real-time lip-synch system. It consists of 3 modules: (1) feature extraction, (2) 3-class acoustic classification module, and (3) lip shape refinement module. When a speech utterance is made, formant frequencies for lip shape are determined, a speech feature is extracted and analyzed in frame unit for segmentation and classification. To be more specific, we first check for the energy level of the acoustic signal in every frame to determine speech segment from silence. If the frame is determined to contain a speech signal on the basis of some established energy-based threshold, the system makes a decision as to what its main content is, being either silence or speech. If the outcome of the decision is 'speech', the system gets its formant frequencies for lip shape and it proceeds to classifying the content into 3 phoneme classes using the SVM classifier. The outcome of the classification task is then fed into the lip shape refinement process using the formant frequencies to determine the height and the width of the lips. The formant frequencies are known to reflect the shape of the vocal tract. With the notion that a vocal tract includes lips, we postulate that formants must reflect the information from the lip shape. Our preliminary experiments confirm this postulation in that the first and second formant frequencies are found to be directly correlated with the lip shape. It was found that the first formant is correlated

with the height of lip and the second formant is correlated with width.

This paper is organized as follows. In Section 2, we describe the SVM-based 3-class acoustic mapping scheme by presenting the proposed feature extractions. Several representative experiments were conducted to determine the performance of the method, and the results are presented and discussed in Section 3. Finally, concluding remarks are presented in Section 4.

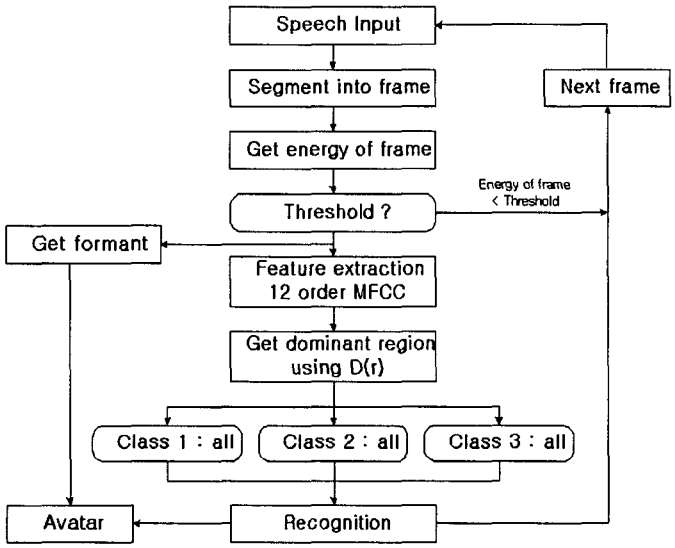

Fig. 1. Real-time lip-synch system block diagram

2. Modeling and extracting feature set for phoneme classification

Fig. 2 shows a block diagram of the proposed phoneme classification algorithm. After detecting speech from input utterance, the input speech is segmented into frame units. The level of energy, then, is analyzed in the frame to determine whether the frame has speech signal or not. If the frame indeed contains a speech signal, we extract 12-th order Mel Frequency Cepstral Coefficients (MFCC) as the feature vector. The classification is then performed over the frames using SVM with the extracted MFCC. The procedure is described in detail as follows.

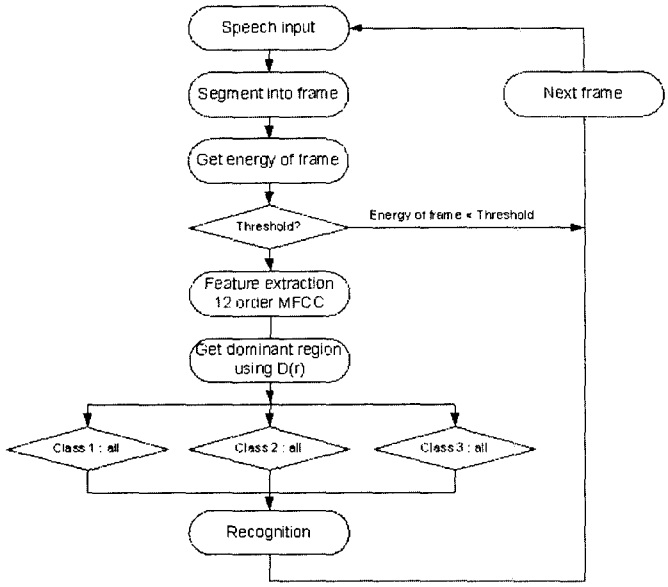

Fig. 2. Block diagram of 3 phoneme classification procedure

Frame of speech: The system performs the classification process at every frame of the continuous speech input signal. To capture long phoneme signals in one frame, we take the frame size at 128 ms (2048 samples at 16 kHz sampling rate) intervals. To avoid missing the short phonemes, however, we take the window of feature at a shifting rate of 8 ms (128 samples at 16 kHz sampling rate). In this way, each frame produces 12 windows whose length is 40 ms (640 samples at 16 kHz sampling rate) each, since Hamming window must contain at least two pitch periods. Fig. 3 shows the structure of one particular frame and next equation is about frame of speech.

$$Frame\ Length = Shift\ term \times (Window\ Numbers - 1) + Window\ Length \tag{2.1}$$

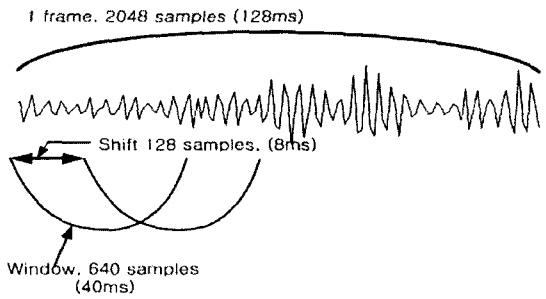

Fig. 3. Shifting window structure of one frame for MFCC feature extraction

Feature vector: We use 12-th order MFCC as feature vector. The delta and acceleration of MFCC are not used since it is assumed that the signal in a frame is stationary; therefore, their influence is expected to be minimal to the classification result. MFCC is known to acoustically model the vocal tract. Consequently, phonemes having similar values would also have similar vocal tract shapes. For the size of the frame chosen here, we can postulate that speech signal has five possible cases in each frame: single consonant, single vowel, two consecutive vowels, a consonant to a vowel and a vowel to a consonant. Since we desire to find a boundary using the change of MFCC value between two neighboring phonemes in one frame, we only consider the cases that are of two consecutive vowels or consonant-to-vowel or vowel-to- consonant in one frame. If there is only one consonant or one vowel in a frame, this segmentation process does not apply. If there are two vowels in one frame, the vocal tract changes the shape and the values of MFCC change in time. Fig. 4 shows an example MFCC values of utterance /oi/, having two vowels in one frame. The changes in MFCC values can be observed in time. Note that the values are stationary in the stable regions of /o/ and /i/ and the values change abruptly at the boundary of the two phonemes.

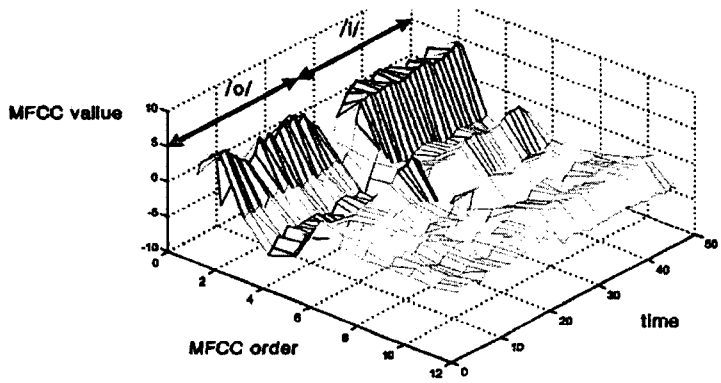

Fig. 4. Abrupt change shown in MFCC of utterance /oi/

If there are consonant-to-vowel or vowel-to-consonant in one frame, the value of MFCC change also. Fig. 5 shows the waveform of /ae/ and /dae/, and Fig. 6 shows the plot of MFCC of these two signals. Note that there is a significant difference at the beginning of signal frame for /dae/ over /ae/.

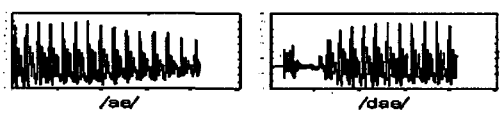

Fig. 5. Signals of /ae/ and /dae/

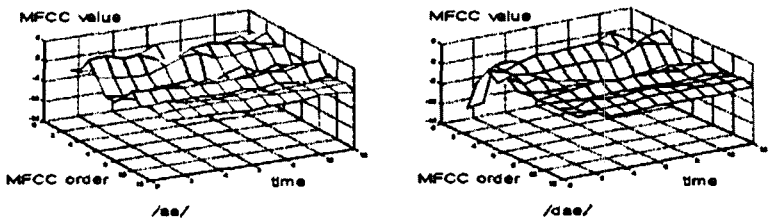

Fig. 6. MFCCs of /ae/ and /dae/

Since we desire to classify what the significant phoneme is in each frame unit, we must search and find the features of an acoustic event that contains a significant boundary signaling the presence of two consecutive vowels or a consonant-to-vowel or a vowel-to-consonant in the frame. To find such a boundary, we look for any acoustic event in a frame at which a major edge or a point of abrupt change occurs between two adjoining phonemes. After locating the edge, we decide which segment of the two sides partitioned by the edge is more dominant in the frame and regard the dominant segment as the frame's feature.

To find such a boundary between two neighboring phonemes, we employ the GLR (General Likelihood Ratio) method [6][7]. The phoneme segmentation method is based on the likelihood ratio of variance of Linear Predictive Coefficient (LPC) excitation. It is a method that constructs the signal into an autoregressive (AR) statistical model and uses test statistics to sequentially detect changes in the parameters of the model, and then estimate the location of the change. A feature vector of a size is extracted in every window and twelve such windows can be found in a frame. This makes the size of the feature vector in one frame.

Modeling of feature vector

We can express the feature vector F_w as

$$F_w = \begin{bmatrix} c_{1,1} & c_{1,2} & \cdots & \cdots & c_{1,12} \\ c_{2,1} & \ddots & & & \vdots \\ \vdots & & \ddots & & \vdots \\ \vdots & & & \ddots & \vdots \\ c_{n,1} & \cdots & \cdots & \cdots & c_{n,12} \end{bmatrix} \tag{2.2}$$

where, w is the frame index, i is the window index, and j is the order of MFCC for c_{ij}. We define $\sigma_{i,j,k}$ as the variance of order k of MFCC within time interval $[i,j]$;

$$\sigma_{i,j,k} = \mathrm{var}(c_{l,k}), \qquad l = i, i+1, \cdots, j-1, j \tag{2.3}$$

The variance $\hat{\sigma}_{i,j}$ of MFCC between time interval $[i,j]$ can be determined by the mean of variance $\sigma_{i,j,k}$ in all order of MFCC as

$$\hat{\sigma}_{i,j} = \frac{1}{12}\sum_{k=1}^{12}\sigma_{i,j,k} \tag{2.4}$$

With this variance, we can define the model, Θ^T, and the mean of each order of MFCC within time interval $[i,j]$, θ^T, as

$$\begin{aligned}\Theta^T &= (\theta^T, \hat{\sigma}_{i,j}^T)\\ \theta^T &= \frac{1}{12}\sum_{n=i}^{j}(c_{n,1},\cdots,c_{n,12})\end{aligned} \tag{2.5}$$

Hypothesis and test

We divide one frame into three types of windows (W_3 where there is no significant acoustic event change in the frame, W_1 and W_2 where each window represents a distinctive acoustic event in the frame) and based on three types window we consider two possible hypotheses(H_0, H_1) as indicated by Equation (2.6) and shown in Fig. 7.

$$\begin{aligned}&H_0 : \Theta = \Theta_0 \qquad \text{for } 1 \le k \le n\\ &H_1 : \exists r \text{ such that } \Theta = \Theta_1 \text{ for } 1 \le k \le r,\\ &\qquad\qquad \text{and } \Theta = \Theta_2 \text{ for } r < k \le 12\end{aligned} \tag{2.6}$$

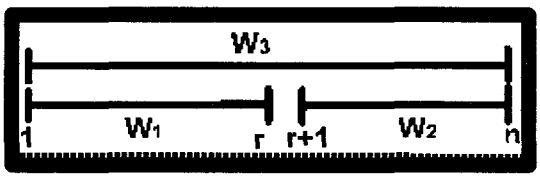

Fig. 7. Three possible windows in one frame.

where n is the size of one frame. The decision is based on the likelihood ratio between these two hypotheses, where the time instant r and the models Θ_i are replaced by their maximum likelihood estimates, so that a change is detected if $D_n \ge \lambda$ where λ is a threshold.

$$D_n : \max_{r}\max_{\Theta_1\Theta_2}\min_{\Theta_0}\log\frac{P(F_w \mid H_1)}{P(F_w \mid H_0)} \tag{2.7}$$

After the inner arguments are satisfied by estimating the θ_i's, D_n can be simplified as:

$$D_n = \max_r D_n(r) \tag{2.8}$$

This means that we now must find the acoustic event where r maximizes D_n. D_n can be written as [6]

$$D_n(r) = n\log\hat{\sigma}_0 - r\log\hat{\sigma}_1 - (n-r)\log\hat{\sigma}_2 \tag{2.9}$$

We can find the boundary of the phoneme by finding r that maximizes the above expression D_n. Since the first term of the right hand side is constant, it can be eliminated for the purpose of finding r as:

$$D_n(r) = r\log\hat{\sigma}_1 + (n-r)\log\hat{\sigma}_2 \tag{2.10}$$

Determining dominant vowel in each speech frame

We now decide if the content of the speech frame is the dominant vowel. There are three possible acoustic events in each speech frame: a combination of two vowels, a combination of a consonant and a vowel, or a combination of a vowel and a consonant. When a frame has a consonant-to-vowel combination or a vowel-to-consonant combination, we must decide which part is a vowel. The reason is that the vowel affects more than consonant for lip synch. To make this determination, we analyze the acoustic properties of speech signal noting the facts that (1) a consonant is shorter than a vowel in general and (2) the variance of the consonant is relatively larger than that of the vowel [8]. With these properties in mind, we choose the part having smaller D_n as the vowel. When a frame has two different vowels as in the 'vowel#1-to-vowel#2' case, we also need to select which of the two is the dominant vowel. In this case, we choose the vowel that has the higher acoustic energy of the two as the dominant one.

After the location of the dominant vowel is determined in a frame, we analyze the feature set of the selected vowel for classification. When the frame segment of the dominant vowel is in the time interval $[i, j]$, its feature set F can be combined by

$$F = \frac{1}{j-i}[\sum_{k=i}^{j} c_{k,1}, \cdots \sum_{k=i}^{j} c_{k,12}] \tag{2.11}$$

Fig 8 shows detection of a boundary in a frame that contains two consecutive vowels using the GLR method.

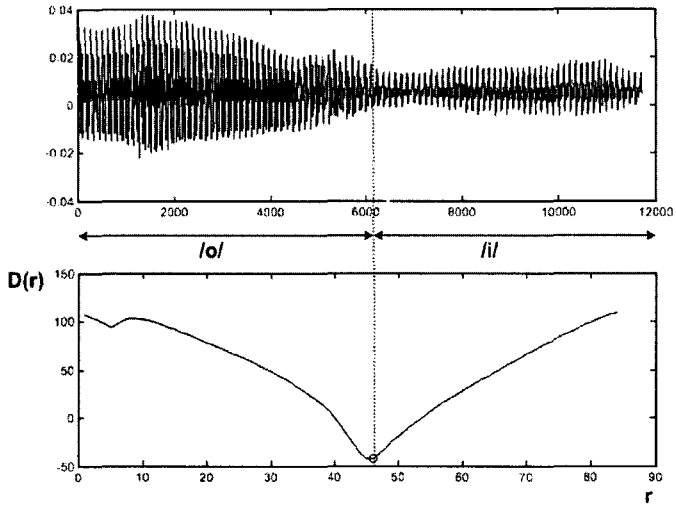

Fig. 8. Detection of boundary in vowel#1-to-vowel#2 speech /oe/

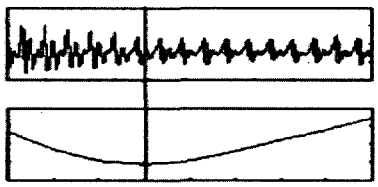

Fig. 9. Detection of boundary in the middle frame of /oe/

Fig. 9 shows the middle frame when we divide the single frame in figure 8 into three frames. The top figure shows the signal waveform in the middle frame capturing /oe/ and the bottom figure shows its corresponding $D(r)$ plot. The bottom figure shows the $D(r)$ plot without labeling the relative intensity between frames to show the minimum $D(r)$ point. In analyzing the signal waveform for determination of multiple vowels, we can observe two particular attributes, (1) the one with higher total energy value is the dominant vowel and (2) a significant difference in the minimum point indicates the presence of two distinctive vowels.

Fig 9 depicts the boundary in the form of a minimum point in the frame. Thus, it is at this point where we segment the frame into two outstanding vowels. Since the total energy value of right part of the frame is higher than that of the left part, the right part is selected as the dominant one (/e/) for this frame. Fig. 10 shows the detection of edge on the speech that consists of a consonant and a vowel. Fig. 11 shows the middle frame when we divide the single

frame in figure 10 into three frames.

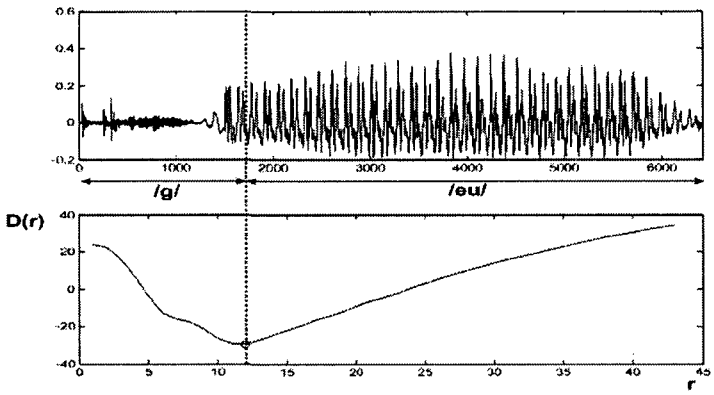

Fig. 10. Detection of boundary in consonant+vowel speech /geu/

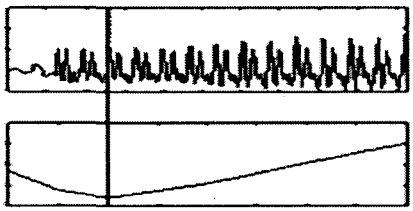

Fig. 11. Detection of boundary in the middle frame for /geu/

In Fig. 11, the top figures show the signal waveform in one frame, each partially capturing /geu/, and the bottom one presents the corresponding $D(r)$ plot of the frame. This figure shows the boundary occurrence between /g/ and /eu/. Since the total energy value of right hand side is higher than that of the left, the right side signal is selected as the dominant one in this frame.

Merging the phonemes : We merge the phonemes into three classes for the purpose of speed and performance in classification. Some key advantages of merging the phonemes into a small number of classes are: the classification error can be reduced; lip shape estimations process becomes more stable; the recognition speed increases. Criteria for merging is based on the Euclidean distance between phonemes and the lip shapes. Euclidean distance between two phonemes (/a/, /ae/) is calculated as follows.

Assume that we obtain one hundred frames from a phoneme pronunciation, with each frame consisting of 12-order feature vectors. Table 1(a) and 1(b) show feature vectors of each /a/ and /ae/ phonemes respectively. $ac_{i,j}$ is MFCC of /a/ phoneme and $aec_{i,j}$ is MFCC of /ae/ phoneme,

where i is the frame number index, and j is the order index of MFCC.

Table 1. Feature vectors of each phoneme

	1-order	2-order		12-order
Frame 1	$ac_{1,1}$	$ac_{1,2}$	.....	$ac_{1,12}$
Frame 2	$ac_{2,1}$	$ac_{2,2}$	.....	$ac_{1,12}$
:			:	
:			:	
:			:	
Frame 100	$ac_{100,1}$	$ac_{100,2}$	.....	$ac_{100,12}$

(a) Feature vectors of /a/ phoneme

	1-order	2-order		12-order
Frame 1	$aec_{1,1}$	$aec_{1,2}$	.....	$aec_{1,12}$
Frame 2	$aec_{2,1}$	$aec_{2,2}$	.....	$aec_{1,12}$
:			:	
:			:	
:			:	
Frame 100	$aec_{100,1}$	$aec_{100,2}$	.....	$aec_{100,12}$

(b) Feature vectors of /ae/ phoneme

First, we compute the mean vector of one hundred frames for each order of feature. We can obtain mean values of each order, as follows.

$$M_j = \frac{1}{100}\sum_{i=1}^{100} ac_{i,j} \quad \text{,where } j \text{ is order index of MFCC.} \tag{2.12}$$

Second, we evaluate Euclidean distance between feature vectors of other phoneme(/ae/) and mean vector of original phoneme (/a/). FD_i is Euclidean distance and i is frame index.

$$FD_i = \sqrt{(M_1 - aec_{i,1})^2 + (M_2 - aec_{i,2})^2 + \cdots + (M_{12} - aec_{i,12})^2} \tag{2.13}$$

Finally, we obtain Euclidean distance value between phonemes (/a/, /ae/), which is the mean of distance values of total one hundred frames.

$$Total\ Euclidean\ Disance = \frac{1}{100}\sum_{i=1}^{100} FD_i \tag{2.14}$$

Table 2 shows the result of Euclidean distances between phonemes. These phonemes are all the single vowels used in Korean (/a/, /ae/, /o/, /u/, /^/, /i/, /eu/).

Table 2. Euclidean distance between phonemes

	/a/	/ae/	/o/	/u/	/^/	/i/	/eu/
/a/	3.65	8.47	10.98	11.32	5.85	9.84	14.41
/ae/	8.44	3.67	13.06	10.84	8.94	9.53	10.56
/o/	10.82	12.88	4.22	8.56	8.18	10.45	12.65
/u/	10.51	10.13	7.91	5.93	8.42	6.8	8.3
/^/	5.99	9.06	8.51	9.49	3.47	9.25	12.49
/i/	9.46	9.1	10.22	7.41	8.67	4.72	9.47
/eu/	13.9	10.05	12.28	8.53	11.8	9.12	5.3

Notice that some phonemes are closer than other phonemes in terms of the Euclidian distance defined earlier. For example, the distance between /a/ and /ae/ is closer than that between /a/ and /eu/. The phonemes of each shaded region show that distance is closer than other phonemes. For more exact merging among phonemes, we determine the other information, namely the shape of lips. Table 3 shows the classification of Korean vowels with respect to the lip shapes(Flat, Round) and tongue positions(Front, Central, Back, High, Mid, Low). For three classes above all we classify vowels with respect to the lip shapes(Flat, Round). Then we can obtain class 2 (Round). And we classify vowels with respect to tongue position(High, Mid, Low). Then we can also obtain class 1 (Mid, Low) and class 3 (High). Finally we suggest that the phonemes can be merged as three classes. Table 4 shows the proposed three classes of vowels.

Table 3. Classification of Korean vowels in the shape of lips and position of tongue

	Front		Central		Back	
	Flat	Round	Flat	Round	Flat	Round
High	/i/		/eu/			/u/
Mid	/ae/		/^/			/o/
Low			/a/			

□ : position of tongue, ▨ : shape of lips

Table 4. Proposed merging of vowels for lip-synch application

	Shape of lips	Position of tongue		Phonemes
		High and low	Front and back	
Class 1	Flat	Middle, low	Front, central	/a/, /ae/
Class 2	Round	High, middle	Back	/o/, /u/
Class 3	Flat	High	Front, central	/eu/, /i/

By merging the phonemes into three classes we can improve speed and accuracy in classification.

In terms of speed, it reduces the computational load in the classification and enables a fast procedure. In this case, as merging seven phonemes into three classes, we obtain reduction of computation time by 4/7. Another advantage of going with only three phoneme classes is the improved accuracy of classification. In general, phoneme classification tends to produce errors due to the presence of multiple phonemes in a single frame. By merging seven phonemes into three classes, we immediately gain more training sets from the same amount of database and also less number of models for classification. The resultant accuracy improvement by the phoneme merging scheme is shown in Section 3.

3. Experiments

For evaluating the effectiveness of the proposed phoneme merging procedure in terms of recognition performance and the speed, we conducted several experiments as follows.

Experimental condition : For speech samples, we constructed the hand labeled data from a phonetically balanced 452 word (PBW452) DB.[10] And we obtain the segmented phone wave files using the hand labeled data and experiment using these files. The PBW452 DB consists of 452 words uttered by 38 male and 32 female speakers twice, respectively. The training set consists of 6 male speakers and 6 female speakers and the test set consists of 14 male speakers and 14 female speakers. Table 5 and Table 6 shows phoneme DB of male speakers and female speakers used in the experiment.

Table 5. Construction of phoneme DB of male speakers

Class		Phoneme		Training set		Test set	
Class 1	11280	/a/	4803	4273	1864	7007	2939
		/ae/	6477		2409		4068
Class 2	8328	/o/	3904	3539	1685	4789	2219
		/u/	1676		697		979
		/^/	2748		1157		1591
Class 3	5932	/eu/	1331	2640	683	3292	648
		/i/	4601		1957		2644

Table 6. Construction of phoneme DB of female speakers

Class		Phoneme		Training set		Test set	
Class 1	17847	/a/	8793	4884	2171	12963	6622
		/ae/	9054		2713		6341
Class 2	12804	/o/	5544	3820	1609	8984	3935
		/u/	2710		868		1842
		/^/	4550		1343		3207
Class 3	8552	/eu/	1994	2507	541	6045	1453
		/i/	6558		1966		4592

Classification performance of phoneme merging : To evaluate the performance of the proposed algorithm, we compared the classification result of our proposed algorithm with that of the HMM based recognition method. HTK (Hidden Markov Model Tool Kit)[2] was used for the HMM-based test and SVM^{light} (SVM tool kit)[9] for the SVM based test. The classification phoneme set consists of the 7 Korean vowels. In the HMM based test, we repeated the test with different numbers of states and 8 mixtures. In the SVM-based test, we used RBF kernel. When RBF kernel is Gaussian, it is as follows,

$$k(x,y)=\exp\{-g\|x-y\|^2\},\ x,y\in R^N,\ g\in R \tag{3.1}$$

where g is constant, x is test vector and y is trained support vector. The constant g and the number of selected support vectors are as in Table 7. Then these values were chosen for the

best results based on trial and error. In this experiment, we began by testing the SVM for recognizing all seven vowels without merging them using the 1-vs-all method: namely (/a/-vs-/ae, o, u, ^, eu, i/), (/ae/-vs-/a, o, u, ^, eu, i/), etc.

Table 7. Constant value and the number of support vectors in experiment

	Male speakers Experiment		Female speakers Experiment	
	Constant (g)	Support Vectors	Constant (g)	Support Vectors
/a/	0.001	1324	0.003	881
/ae/	0.006	722	0.0005	2494
/o/	0.002	1619	0.015	1327
/u/	0.03	931	0.007	1338
/^/	0.02	1282	0.004	1216
/eu/	0.006	577	0.018	649
/i/	0.004	496	0.009	652

Table 8 and Table 9 show the results of the experiment. The results show that in terms of accuracy SVM-based recognition is similar to HMM-based recognition.

Table 8. Result of phoneme classification experiment about male speakers

	HMM			SVM
	1 state	2 state	3 state	RBF
/a/	80.20	88.16	83.63	93.47
/ae/	77.06	81.05	82.94	88.4
/o/	74.27	73.91	76.39	87.07
/u/	67.52	72.83	70.68	56.28
/^/	83.97	81.40	86.05	79.89
/eu/	71.76	81.02	81.94	61.42
/i/	95.61	95.65	95.95	95.39
Average(%)	78.63	82.00	82.51	80.27

Table 9. Result of phoneme classification experiment about female speakers

	HMM			SVM
	1 state	2 state	3 state	RBF
/a/	95.55	97.90	96.63	99.18
/ae/	75.59	77.56	76.22	81.44
/o/	77.46	81.35	80.08	87.98
/u/	50.65	60.75	62.32	50.33
/^/	89.87	90.05	91.71	79.58
/eu/	69.99	71.99	69.99	65.45
/i/	95.01	95.49	96.04	96.10
Average(%)	79.16	82.16	81.86	80.01

To verify that the merging of phonemes improves the classification performance, we conducted representative classification performance experiments with the merged classes. The experimental conditions are set identically as those described above. In the SVM-based test, constant of Gaussian RBF kernel in Eq.(3.1) and the numbers of selected support vectors are as Table 10. These values were chosen for the best results based on trial and error. Again we classified as 1-vs-all method, namely (/class 1/ - vs - /class 2, class 3/), (/class 2/ - vs - /class 1, class 3/), (/class 3/ - vs - /class 1, class 2/) and executed recognition test.

Table 10. Constant value and the number of support vectors in experiment

	Male speakers Experiment		Female speakers Experiment	
	Constant (g)	Support Vectors	Constant (g)	Support Vectors
Class 1	0.01	1129	0.003	2218
Class 2	0.01	1123	0.018	1244
Class 3	0.01	872	0.015	1261

Table 11 and Table 12 show the results of the experiment. Again they indicate that the SVM-based classifier achieves similar performance to HMM-based classifier. However, we also noted that the classification performance shown in Table 11 and Table 12 is significantly better than the one shown in Table 8 and Table 9. These results indicate that the classification error can indeed be reduced by merging the phonemes with similar lip openings.

Table 11. Phoneme merging experiment with male speakers

	HMM			SVM
	1 state	2 state	3 state	RBF
Class 1	87.38	86.16	85.51	89.07
Class 2	93.09	94.24	92.32	92.23
Class 3	91.49	92.47	93.10	90.49
Average(%)	90.65	90.96	90.31	90.60

Table 12. Phoneme merging experiment with female speakers

	HMM			SVM
	1 state	2 state	3 state	RBF
Class 1	89.14	88.18	88.19	91.49
Class 2	87.27	89.63	89.66	90.45
Class 3	92.77	93.40	94.41	89.63
Average(%)	89.73	90.40	90.75	90.52

4. Conclusions

For real-time lip-synch, main issues are as follows : better classification performance, faster classification speed and more smooth lip-shape motion. We investigated an SVM-based classification/merging algorithm for achieving such a real-time lip-synch system. The representative experiments have shown that the method is more accurate and also has faster recognition speed than other HMM-based systems. In addition, we confirmed that the refinement of lip shape based on the formant frequencies is useful for lip-synch systems. Namely we proposed that the height and width of lip shapes are linearly dependent on the first and second formant frequencies, and using this relationship we developed a lip-refinement procedure to make smoother lip motion. We applied these algorithms successfully in a real time lip-synch system using the Microsoft Visual Studio.

Acknowledgements

This research was supported by Korea University's internal funding program.

References

[1] Rabiner, L., Biing-Hwang Juang. 1993. Fundamentals of speech recognition. Prentice Hall, pp. 321-322.

[2] Young, S. 2001. The HTK(Hidden Markov Model Tool Kit) book, Cambridge University Engineering Department, pp. 2-13.

[3] Clarkson, P., Moreno, P. J. 1999. "On the use of support vector machines for phonetic classification," in Proc. Int. Conf. Acoust., Speech, Signal Processing. 2, 585-588.

[4] Hiroshi Shimodaira, Ken-ichi Noma, Mitsuru Nakai, Shigeki Sagayama, 2001. "Support Vector Machine with Dynamic Time-Alignment Kernel for Speech Recognition", Eurospeech 2001, Scandinavia.

[5] Golowich, Steven E. & Don X. Sun, 1998. "A Support Vector/Hidden Markov Model Approach to Phoneme Recognition", ASA Proceedings of the Statistical Computing Section, pp. 125-130.

[6] Regine Andre-Obrecht, 1998. "A New Statistical Approach for the Automatic Segmentation of Continuous Speech Signals", IEEE Transactions on Acoustic, Speech, and Signal Processing, Vol. 36, No. 1, January.

[7] A Von Brandt, 1983. "Detecting and estimating parameters jumps using ladder algorithms and likelihood ratio test", in Proc, ICASSP, Boston, MA, pp. 1017-1020.

[8] Ming-Tzaw Lin, Ching-Kuen Lee, & Chin-Yi Lin. 1999. "Consonant/Vowel segmentation for Mandarin syllable recognition", Computer Speech and Language, Vol. 13, pp. 207-222.

[9] "SVM^{light} website", http://svmlight.joachims.org/

[10] Kim, B., Kim, J., Kim, S., Lee, Y. 1997. "A Study on the Design and the Construction of a Korean Speech DB for Common Use," Journal of Acoust. Soc. of Korea, Vol.16, No.4, pp.35-41.

received: 2004. 4. 30
accepted: 2004. 6. 15

▲ Kunyoung Lee
Dept. of Electronics and Computer Engineering, Korea University
5ka-1, Anam-dong, Sungbuk-ku, Seoul 136-701, Korea
Tel: +82-2-3290-3239

▲ Hanseok Ko
Dept. of Electronics and Computer Engineering, Korea University
5ka-1, Anam-dong, Sungbuk-ku, Seoul 136-701, Korea
Tel: +82-2-3290-3239