(i) Title of the submitted paper, authors names, and affiliations

Title : A Real – time Lipreading LSI for Word Recognition

Authors and affiliations :
Kazuhiro Nakamura
(Center for Information Media Studies, Nagoya University),
Noriaki Murakami
(Department of Information Engineering, Nagoya University),
Kazuyoshi Takagi
(Department of Information Engineering, Nagoya University),
Naofumi Takagi
(Department of Information Engineering, Nagoya University)

(ii) Postal & e – mail addresses and phone & FAX numbers of the contact author

Postal address :
Furo – cho, Chikusa – ku, Nagoya – shi, 464 – 8603 JAPAN
e – mail addresses : nakamura@nuie.nagoya – u.ac.jp,
phone : +81 – 52 – 789 – 2763
FAX : +81 – 52 – 789 – 3798

(iii) Paper category number (from the above list) that best describes the paper

Paper category : Digital Circuits

(iv) Name of the speaker

Name of the speaker : Kazuhiro Nakamura

A Real-time Lipreading LSI for Word Recognition

Kazuhiro Nakamura† Noriaki Murakami‡ Kazuyoshi Takagi‡ Naofumi Takagi‡

† Center for Information Media Studies, Nagoya University
‡ Department of Information Engineering, Nagoya University
Furo-cho, Chikusa-ku, Nagoya-shi, 464-8603 JAPAN
{nakamura,noria,ktakagi,ntakagi}@takagi.nuie.nagoya-u.ac.jp

Abstract— In the paper, we present a real-time lipreading LSI for recognizing spoken words from lip movement. The LSI recognizes up to 8 words based on the Hidden Markov Model (HMM). The LSI accepts the 256 $\times$ 256 8-bit gray-scale images from a camera, and outputs the 3-bit symbol code of words for 43 images (corresponding to 1.53 sec). We present a lipreading algorithms optimized for hardware implementation. We have designed the lipreading LSI and fabricated a 4.9 mm $\times$ 4.9 mm chip using 0.35 μm process via VDEC Rohm. The LSI performs real-time recognition at 40 MHz operation.

1. Introduction

Speech-based interfaces have become one of popular human interfaces, and there have been developed many speech recognition LSIs. Lipreading is one of high-level human interfaces with no-keyboard, and there have been developed many lipreading software systems. Lipreading is also useful for auxiliary measures of speech recognition. From these points, we develop a handy lipreading LSI that works in real-time with small area, less memory consumption and low power dissipation. The LSI recognizes spoken words from lip movement.

In this paper we present a real-time lipreading LSI for recognizing spoken words from lip movement. The LSI recognizes up to 8 words based on the Hidden Markov Model (HMM) which is a well known user-independent recognition method. The LSI accepts the 256 $\times$ 256 8-bit gray-scale images from a camera, and outputs the 3-bit symbol code of words with the highest probability on the HMM for 43 images (corresponding to 1.53 sec). The LSI includes lip tracking module, vector quantization (VQ) module and HMM-based recognition module for lipreading.

We first implemented the lipreading algorithm in C language to check the behavior, and we converted the C programs to match constraints for hardware implementation. In the conversion, floating point operations are changed to fixed-point operations keeping the recognition quality. Based on the C description, we have designed the lipreading LSI using Verilog HDL and fabricated a 4.9 mm $\times$ 4.9 mm chip using 0.35 μm process via VDEC Rohm. The chip performs real-time recognition at 40 MHz operation.

The rest of the paper is organized as follows. Section 2 shows the structure of our HMM-based lipreading system. In Section 3, we present lipreading algorithms optimized for hardware implementation. Section 4 shows hardware design of lipreading LSI. Section 5 describes the implementation and evaluation of the lipreading LSI.

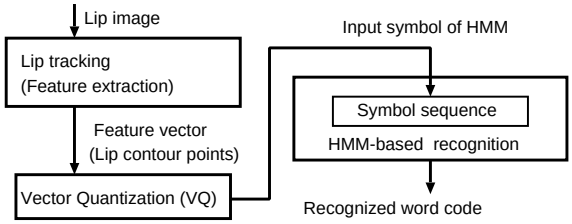

Fig. 1. Structure of Our HMM-based Lipreading System.

2. Lipreading Based on Hidden Markov Model (HMM)

Many lipreading methods have been developed[1, 2, 3, 4]. Within these methods, a recognition method based on Hidden Markov Model (HMM) becomes popular, since the method is effective for user-independent recognition[5]. We adopt the HMM-based method for our hardware design and adopt words as a recognition unit. Words are basic unit of speech and are suitable for a human interface of embedded systems.

Our lipreading system accepts 8-bit gray-scale lip images on 256 $\times$ 256 dot plane. Fig.1 shows the structure of our HMM-based lipreading system, which includes three major tasks: lip tracking, vector quantization (VQ) and HMM-based recognition. In recognition processes, lip contour points are computed by the lip tracking as a feature vector, and then the feature vector is translated into an input symbol of HMMs by the VQ. Finally, the code of recognized word is output after the HMM-based recognition.

3. Lipreading Algorithm Optimized for Hardware Implementation

3.1. Lip Tracking

In general, a lip contour is a boundary between light-color pixels and dark-color pixels in lip image. A lip tracking algorithm based on contrast analysis have been developed[4]. The contrast-based lip tracking algorithm takes a lip image and positions of lip contour edges (left and right) as inputs. These edge positions are used as initial points to approximate the lip contour by polygon.

We modify the algorithm for hardware implementation, and introduce an algorithm for detecting lip contour edges automatically. The algorithm searches both left and right sides of the lip image (56-dot width) with 40 $\times$ 40 dot square as shown in Fig.2(a), and detects the darkest area. The detected area is searched again with 8 $\times$ 8 dot square. And finally, lip edges

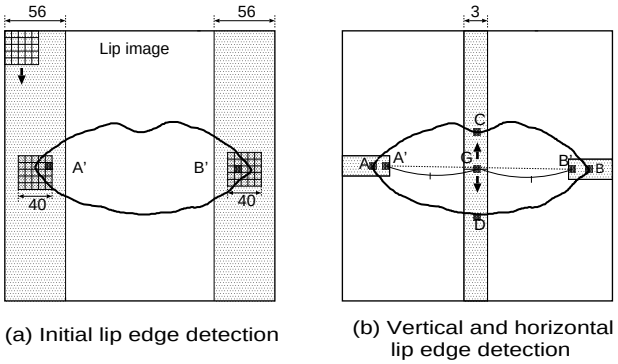

Fig. 2. Vertical and horizontal lip edge search.

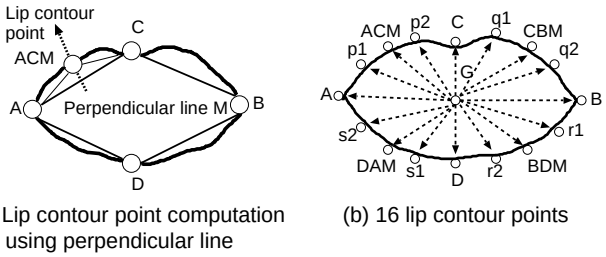

Fig. 3. Lip contour points computation.

A' and B' are detected as described in Fig.2(a). The size of these square is decided from the default distance between the camera and the lip.

After that, the middle point G of line A'B' is computed. And then, the algorithm searches the lip image vertically and horizontally as shown in Fig.2(b). The points with maximum contrast along the search are lip contour edges, because the area around lip contour has higher contrast. The points A, B, C and D are detected by the search as shown in Fig.2(b).

Fig.3(a) shows lip contour points computation with perpendicular line[4]. In the figure, lip contour point ACM is obtained by the contrast-based search along the perpendicular line M which goes through the middle point of line AC. New contour points are computed in a similar way, until 16 lip contour points are obtained. These 16 points are specified with x and y coordinates from the point G as shown in Fig.3(b). In these 32 values, we use 28 values as feature vector, because x of C and D points and y of A and B points are always 0. Note that we use a line with inclination 1 instead of exact perpendicular line to reduce hardware cost for exact computation. Fig.4(b) shows 6 regions (30×50) around the point G. We also compute the mean pixel value for each region, and use the means as feature vector. Fig.5 shows the lip tracking process for computing 16 lip contour points.

3.2. Vector Quantization (VQ)

The 34-dimensional feature vector obtained by lip tracking is translated into one of 256 input symbols to HMMs using a table called VQ codebook. The VQ codebook is a 256×34 table as shown in Fig.6. We choose the code with minimum

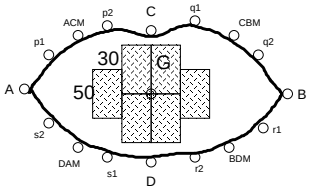

Fig. 4. Mean pixel value for each 6 regions.

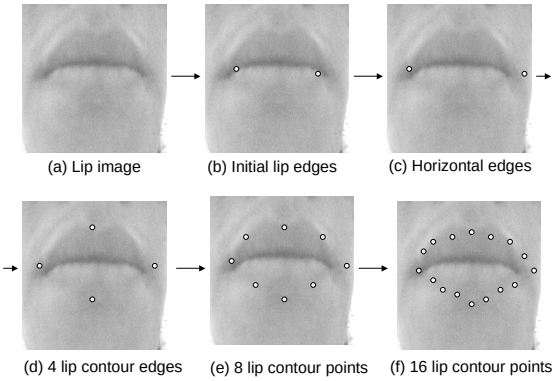

Fig. 5. Lip tracking process.

difference in the representative feature vector. The difference is the sum of the absolute value of the difference in each element.

The VQ codebook is obtained by clustering training data[5]. In the clustering, the centroid of training data is computed, and training data is partitioned into two based on the centroid. The centroid computation and the partitioning are repeated until 256 centroids are obtained. Centroids are used as representative feature vectors in the VQ codebook.

3.3. Hidden Markov Model (HMM)

Words are recognized using Hidden Markov Model (HMM), and one HMM is constructed for each word. We show a definition of an HMM.

Definition 1 An HMM M is a 6-tuple $(S, \Sigma, A, B, \Pi, F)$ where

- S is a finite set of states,
- Σ is an input alphabet,
- $A : S \times S \rightarrow \mathbf{R}$ is a state transition probability function, where $\mathbf{R}$ is the real number set, $\sum_j A(q_i, q_j) = 1$,
- $B : S \times S \times \Sigma \rightarrow \mathbf{R}$ is an output probability function, $\sum_k B(q_i, q_j, k) = 1$,
- $\Pi = \{\pi_{q_i} \mid$ probability that $q_i \in S$ is an initial state $\sum_i \pi_{q_i} = 1\}$ is a set of initial probabilities,
- $F (\subseteq S)$ is a finite set of final states.

In the following, we describe $A(q_i, q_j)$ and $B(q_i, q_j, k)$ as $a_{q_i q_j}$ and $b_{q_i,q_j}(k)$ respectively. The behavior sequences of M with respect to an input sequence $X = x_1 x_2 \ldots x_N$ $(x_i \in \Sigma)$ are sequences whose length is N, and each sequence starts from a state whose initial state probability is not 0 and ends a final state. Let $Q_j = q_0^{(j)} q_1^{(j)} \cdots q_N^{(j)}$ $(q_N^{(j)} \in F)$ be jth sequence. Probability $P(x_1 x_2 \cdots x_N \mid M)$ that X is accepted by M is obtained by the following formula:

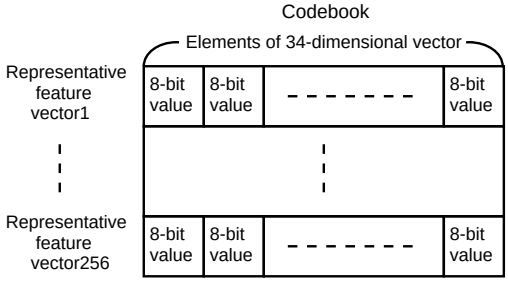

Fig. 6. Structure of the VQ codebook.

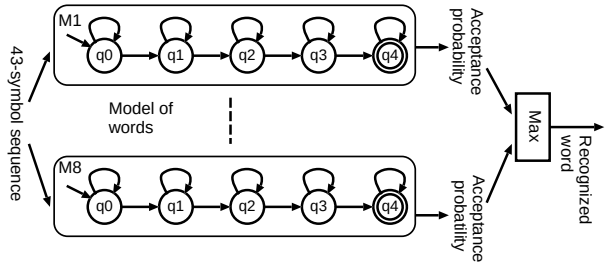

Fig. 7. HMM of words and HMM-based lipreading.

$$P(x_1 x_2 \cdots x_N \mid M) = \sum_{Q_j} \pi_{q_0^{(j)}} \prod_{i=1}^{N} a_{q_{i-1}^{(j)}, q_i^{(j)}} b_{q_{i-1}^{(j)}, q_i^{(j)}}(x_i).$$

3.4. Recognition Algorithm Based on Hidden Markov Model

In the following, 8 HMMs are used to recognize 8 words. We use 5-state left-right HMMs based on [5], which have only one-way state transition as shown in Fig.7. HMMs accept 43 input symbols corresponding to 43 images (corresponding 1.53 sec) and decide the word.

The acceptance probability $P(X|M_h)$ $(h = 1, 2, ..., 8)$ with respect to 43 symbols $X = x_1 x_2 \ldots x_{43}$ is computed for all HMMs. Then the word whose HMM has the greatest probability is recognized.

Since hardware cost of a multiplier is expensive, $\log(\max_Q P(X|M_h))$ is used instead of the probability $P(X|M_h)$. $\max_Q P(X|M_h)$ is the probability of a behavior sequence with the highest probability. $\alpha'_{q_i,n}$ is the probability to stay the state q_i after the n-th symbol.

$$\alpha'_{q_i,n} = \begin{cases} \log \pi_{q_i},\ (n = 0,\ \pi_{q_i} \neq 0) \\ -\infty,\ (n = 0,\ \pi_{q_i} = 0) \\ \max\limits_{q_j \in S,\ a_{q_j,q_i} \neq 0} \left(\alpha'_{q_j,n-1} + \log\left(a_{q_j,q_i} \cdot b_{q_j,q_i}(x_n) \right) \right) \end{cases} \cdots (4)$$

$$\log(\max_Q P(X|M_h)) = \max_{q \in F} \alpha'_{q,43}$$

4. Hardware Design of Lipreading LSI

4.1. Architecture Design

Fig.8 shows a block diagram of the lipreading LSI. The lipreading LSI includes a image handling module, a lip tracking module, a VQ module, an HMM-based recognition module as described in Section 3. The VQ codebook and HMM parameters of 8 HMMs are also stored in the external memory (130KB) of word length 8-bit. The input image is stored into the external memory by the image handling module.

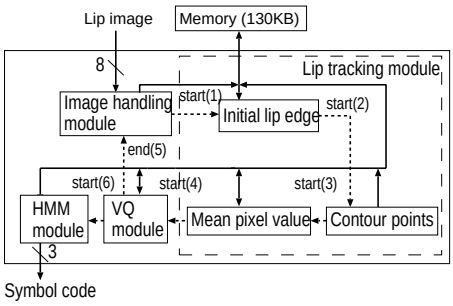

Fig. 8. Block diagram of the lipreading LSI.

```
VQ (C[], CodeBook[][])
 C[0, ..., 33]:  a feature vector;
 CodeBook[0,...,255][0,...,33]:  VQ codebook;
 symbol:  a resulting symbol ;
begin
 min_dist = ∞;
 for (i = 0; i < 256; i = i+1) { /* search */
  tmp_dist = 0.0;
  for (j = 0; j < 34; j = j+1) {/* distance */
   tmp_dist += |C[j] - CodeBook[i][j]|;
  }
  if (tmp_dist < min_dist) {
   min_dist = tmp_dist; symbol = i;
  }
 }
 return symbol;
end
```

Fig. 9. A Vector Quantization algorithm.

4.2. Hardware Module for Lip Tracking

The lip tracking module searches the input lip image stored in external memory, and detects lip edges using 1,160,000 clock cycles (Initial lip edge). Then, the lip tracking module determines 4 lip contour points using 12,000 clok cycles, and computes other 12 contour points using 6,400 clock cycles (Contour points). After computing the mean pixel value (Mean pixel value), a start signal(6) is sent to the VQ module with a 34-dimensional feature vector.

4.3. Hardware Module for Vector Quantization

The VQ module accepts a 34-dimensional feature vector and translates to the 8-bit symbol code. As shown in Fig.6, each element of the feature vector is a 8-bit fixed point integer. The VQ codebook is computed in advance and stored in external memory.

Fig.9 shows a vector quantization algorithm. The algorithm starts from searching *CodeBook*[ ][ ], and if a representative feature vector that is closer to the input feature vector *C*[ ] is found, then the index of the representative feature vector is set to *symbol*. Finally, the index whose representative feature vector is closest to *C*[ ] is obtained and used as an 8-bit symbol. 46,500 clock cycles are used to map a feature vector to an 8-bit symbol.

```
Highest_logP (X[])
 X[n]:  a symbols sequence (n = 1,2,...,43);
 logP:  a probability log P(X|M) ;
 highest_logP:  the highest probability ;
 α[][], tmp_α:  intermediate results ;
 max_α:  maximum value of α[4][43] ;
 HMM_h.A[][]:  HMM parameters of HMM_h ;
 HMM_h.B[][][]:  HMM parameters of HMM_h ;
 HMM_h.code:  symbol code of a word ;
begin
 highest_logP = -∞;  symbol_code = 0 ;
 /* foreach HMM_h */
 for (h = 1; h ≤ 8; h = h+1) {
  /* computing probability */
  max_α = -∞;  α[0][0] = log 1.0;
  for (i = 1; i < 5; i = i+1)
   { α[i][0] = -∞; }
  for (n = 1; n ≤ 43; t = t+1) {
   for (i = 0; i < 5; i = i+1) {
    tmp_α = 0.0;
    for (j = 0; j ≤ i; j = j+1) {
     tmp_α = α[j][n−1] + HMM_h.A[j][i]
           + HMM_h.B[j][i][X[n]];
     if (tmp_α > max_α)
      { max_α = tmp_α; }
    }
    α[i][n] = max_α;
   }
  }
  logP = alpha[4][43]; /*computed probability*/
  if (logP > highest_logP)
   { highest_logP = logP; symbol_code = HMM_h.code; }
 }
end
```

Fig. 10. An algorithm for computing the highest probability.

4.4. Hardware Module for HMM-based Recognition

The HMM module starts recognition when 43 symbols are stored into X in external memory. The X is used as a queue. When a new symbol arrives at the HMM module, the symbol is put into X and the oldest symbol is removed from X. In the recognition process, 8 acceptance probabilities $P(X|M_h)$ $(h = 1,2,...,8)$ are sequentially computed based on the above formula (4). Then, a word whose HMM has maximum acceptance probability in the 8 HMMs is chosen as recognized word, and finally the 3-bit symbol code of the word is output. Fig.10 shows detailed algorithm for computing 8 acceptance probabilities and choosing the words with highest probability. The HMM module takes 14,000 clock cycles for the computation. The HMM module has a 16-bit adder. 16-bit HMM parameters (for functions A and B) are generated by software in advance and stored in external memory.

5. Design and Evaluation of The Lipreading LSI

We have implemented HMM-based lipreading system (including a lip tracking, a VQ and an HMM-based recognition program) in C language in advance of the hardware design. The C program is about 1,300 lines. We also made VQ codebook generator and an HMM parameter generator.

5.1. Design of Real-time Lipreading LSI

We translated the C program into Verilog HDL. The Verilog HDL source is about 9,000 lines. Verilog HDL description of the lipreading chip is synthesized with Synopsys Design Compiler, and the chip layout is performed with Avant! ApolloII. We have fabricated a 4.9 mm $\times$ 4.9 mm chip via VDEC Rohm 0.35 μm process. The picture of the chip is shown in Fig.11. The chip includes about 18,000 cells.

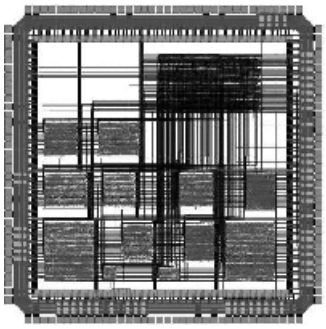

Fig. 11. Layout of the lipreading LSI.

5.2. Evaluation of The Lipreading LSI

We have compared the speech recognition chip and the C program with fixed-point operations executed on a workstation (CPU UltraSPARC III 900 MHz, Main memory 1024 MB). We have measured the elapsed CPU time of the software to recognize 43 images. On average, the elapsed CPU time was 0.091 sec and the elapsed clock cycles were 81.9×10^6. On the other hand, the real-time lipreading chip takes 1.3 sec at 40 MHz operation and 53.4×10^6 clock cycles for 43 images. Hence, the chip is 1.5 times faster than the software if the clock frequency is the same. This is because, the chip includes considering parallel operation.

6. Conclusions

In this paper, we have presented a real-time lipreading LSI. The lipreading LSI recognizes up to 8 words using HMM-based method.

We are now designing a test board for the lipreading LSI. We would like to develop more hardware oriented lipreading and speech recognition algorithms.

Acknowledgments

We would like to express our thanks to members of Takagi Laboratory at Nagoya University for their discussions and comments. The VLSI chip in this study has been fabricated in the chip fabrication program of VLSI Design and Education Center (VDEC), the University of Tokyo with the collaboration by Rohm Corporation and Toppan Printing Corporation.

References

[1] A. Adjounani and C. Benoit, *Audio-vidual Speech Recognition Compared Across Two Architectures*, Eurospeech'95, 1995.

[2] Keren Yu, Xiaoyi Jiang and Horst Bunke, *Sentence Lipreading Using Hidden Markov Model with Integrated Grammar*, International Journal of Pattern Recognition and Artificial Intelligence, vol.15, No.1, 2001.

[3] P. Duchnowski, U. Meier and A. Waibel, *See Me, Hear Me: Integrating Automatic Speech Recognition and Lipreading*, ICSLP'94, 1994.

[4] K. Sato, M. Ishii and M. Nishida, *Lip Feature Extraction and Analysis for Lip Reading*, JCMI'97, 1997, (in Japanese).

[5] Lawrence Rabiner and Biing-Hwang Juang, *Fundamentals of Speech Recognition*, Prentice Hall PTR Prentice-Hall, 1993.