International Journal of Advanced Intelligence
Volume 2, Number 1, pp.119-144, July, 2010.

AIA International Advanced Information Institute

A New Geometric Distance Method to Remove Pseudo Difference in Shapes

Michihiro Jinnai

Department of Electro-Mechanical Systems Engineering,
Kagawa National College of Technology
355 Chokushi-cho, Takamatsu, 761-8058, Japan
jinnai@t.kagawa-nct.ac.jp

Satoru Tsuge

School of Informatics, Daido University
10-3 Takiharu-cho Minami-ku, Nagoya, 457-8530, Japan
tsuge@daido-it.ac.jp

Shingo Kuroiwa

Department of Information and Image Science, Chiba University
1-33 Yayoi-cho Inage-ku, Chiba, 263-8522, Japan
kuroiwa@faculty.chiba-u.jp

Minoru Fukumi

Faculty of Engineering, University of Tokushima
2-1 Minami-josanjima, Tokushima, 770-8506, Japan
fukumi@is.tokushima-u.ac.jp

Received (January 2010)
Revised (May 2010)

In our previous paper, a new similarity scale called the Geometric Distance was proposed. With the conventional geometric distance algorithm, there are the following three shortcomings. 1. Since standard and input patterns are normalized to have the same area, a pseudo difference in shapes occurs between them. 2. Since "shape variation" is calculated in each combination of the standard and input patterns, the processing overhead increases when the number of standard patterns increases. 3. Since reference patterns are evaluated for each movement position of a normal distribution, the computational memory overhead increases when the number of components of standard and input patterns increases. To counter these shortcomings, a new geometric distance algorithm is proposed. 1. It is derived without normalization of the standard and input patterns, so that the pseudo difference in shapes is removed. 2. It reduces the processing overhead by separating the calculation of "shape variation" into a registration process and a recognition process. 3. It reduces the computational memory overhead by sharing a single reference pattern. Experiments in vowel recognition were carried out using the same voice data as the previous paper. At a mean of 5 dB SNR, the recognition accuracy improved from 78% to 82% over the conventional algorithm.

Keywords: Similarity measures; Distance functions; Pattern matching; Noise robust.

1. Introduction

In pattern recognition, a shape of an input pattern is compared with that of a standard pattern using a similarity scale. Traditionally, for the similarity scale, the Euclidean distance and cosine similarity have been widely used.[1,2] In recent years, various similarity scales have been proposed for comparing two patterns in speech recognition,[2,3,4,5,6,7,8,9,10] pattern classification[11] and image retrieval.[12,13,14,15]

In our previous paper,[16] a new similarity scale called the Geometric Distance was proposed. At that point, we initially developed a mathematical model incorporating the following two characteristics.
<1> The distance metric must show good immunity to noise.
<2> The distance metric must increase monotonically when a difference increases between peaks of the standard and input patterns.
An algorithm based on a one-to-many point mapping was proposed to realize the mathematical model. In the algorithm, the difference in shapes between the standard and input patterns is replaced by the shape change of a normal distribution, and the magnitude of this shape change is numerically evaluated as a variable of the moment ratio ("shape variation") that is derived from the kurtosis.

With the above conventional algorithm, there are the following three shortcomings. 1. Since the standard and input patterns are normalized to have the same area, a pseudo difference in shapes occurs between the standard and input patterns and the recognition performance of geometric distance becomes unpredictable. 2. Since "shape variation" is calculated in each combination of the standard and input patterns if we use multiple standard patterns and a single input pattern, the processing overhead increases when the number of standard patterns increases. 3. Since positive and negative reference patterns are evaluated for each movement position of the normal distribution, the computational memory overhead increases in proportion to the square of the number of components of the standard and input patterns.

In this paper, we propose a new geometric distance algorithm that can realize the above mathematical model and that can also improve the above shortcomings. 1. The new algorithm is derived without normalization of the standard and input patterns, so that the pseudo difference in shapes is removed and the recognition performance of geometric distance becomes stable. 2. The new algorithm reduces the processing overhead during an input pattern recognition process by separating the calculation of "shape variation" into a standard pattern registration process and an input pattern recognition process. 3. The new algorithm reduces the computational memory overhead by sharing a single reference pattern.

The proposed algorithm can be applied widely to pattern recognition such as pattern classification or clustering and image retrieval using the distance between histograms. This paper explains the technique using the same voice data and feature parameters as those of the previous paper.[16] The paper consists of the following sections. Section 2 describes the shortcomings that are found in the conventional

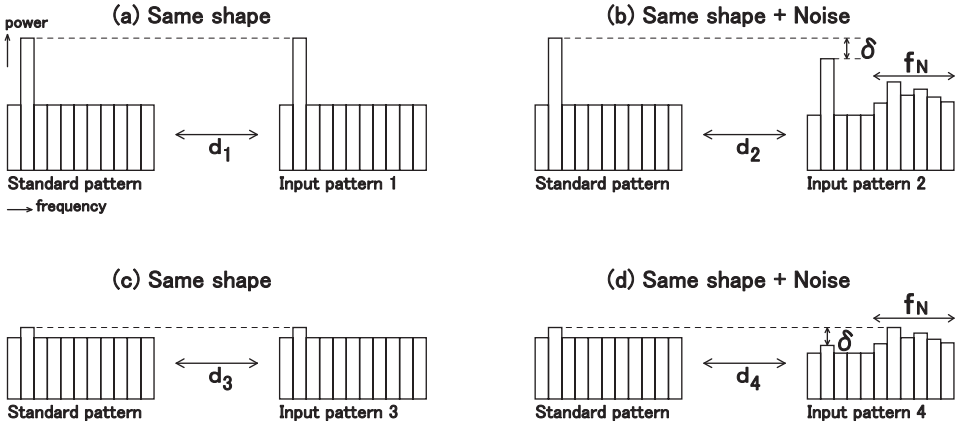

Fig. 1. Pseudo difference in shapes.

algorithm. Section 3 describes the new algorithm, provides the evaluation results of the processing overhead and the computational memory required for the algorithm, describes numerical experiments, and describes that the algorithm performs well. Section 4 describes the speech recognition tests that have been carried out, and describes the stabilized recognition performance. Section 5 provides the conclusions and touches on future work.

2. Conventional Geometric Distance Algorithm

With the conventional algorithm, the standard and input patterns are normalized to have the same area. Then, a difference in shapes between standard and input patterns is replaced by a shape change of a normal distribution. If this method is used, a pseudo difference in shapes may occur between standard and input patterns due to normalization of power spectrum. As an example, Figs. 1(a) and (b) show the standard pattern, input patterns 1 and 2 having the same shape in the power spectrum. In the input pattern 2, however, noise has been added to the power spectrum in frequency band f_N, and the input pattern 2 has been normalized to have the same area as the standard pattern. As a result, a pseudo difference δ in shapes occurs at the peaks of the standard pattern and the input pattern 2 as shown in Fig. 1(b). Figs. 1(c) and (d) show another example of this. However, the input pattern 4 has been normalized to have the same maximum value as the standard pattern. After normalization, a pseudo difference δ in shapes occurs again at the peaks of the standard pattern and the input pattern 4 as shown in Fig. 1(d). Because the pseudo difference in shapes always occurs regardless of the use of any normalization method, it results in an actual shape change of the normal distribution and the recognition performance of geometric distance becomes unpredictable.

Moreover, with the conventional algorithm, we need to calculate the moment ra-

tios (shape variation) in each combination of standard and input patterns if we use multiple standard patterns and a single input pattern. Hence the processing overhead increases when the number of standard patterns increases. If the calculation of pattern recognition is separated into a standard pattern registration process and an input pattern recognition process, then the moment ratios (shape variation) are calculated during the input pattern recognition process. Therefore, the calculation time of the input pattern recognition process increases in proportion to the number of standard patterns.

However, with the conventional algorithm, we need to evaluate positive and negative reference patterns for each movement position of the normal distribution. Therefore, the computational memory overhead increases in proportion to the square of the number of components of the standard and input patterns.

Because of these shortcomings, we propose a new algorithm that we will introduce in the next section.

3. New Geometric Distance Algorithm

In this section, we use the same mathematical model as the conventional algorithm. We propose a new algorithm that can realize the mathematical model and that can also improve the above shortcomings. Specifically, we use a weighting vector that consists of a rate of change of the moment ratio, and create two weighted pattern vectors by performing the product-sum operation using the weighting vector and the standard pattern vector and the product-sum operation using the weighting vector and the input pattern vector. Then, we use the angle between these weighted pattern vectors as a new geometric distance. As a result, we can remove the pseudo difference in shapes and stabilize the recognition performance of the geometric distance. Also, we can reduce the processing overhead during the input pattern recognition process and reduce the computational memory overhead for the positive and negative reference pattern vectors. In the second half of this section, numerical experiments are carried out using some geometric patterns with the "difference" and "wobble", and the proposed algorithm is confirmed to perform well.

3.1. *Properties of moment ratio*

With the conventional algorithm, the difference in shapes between standard and input patterns is replaced by the shape change of the normal distribution, and the magnitude of this shape change is numerically evaluated as a variable of the moment ratio. If variable u_i is a discrete value, moment ratio A of function $f(u_i)$ can be

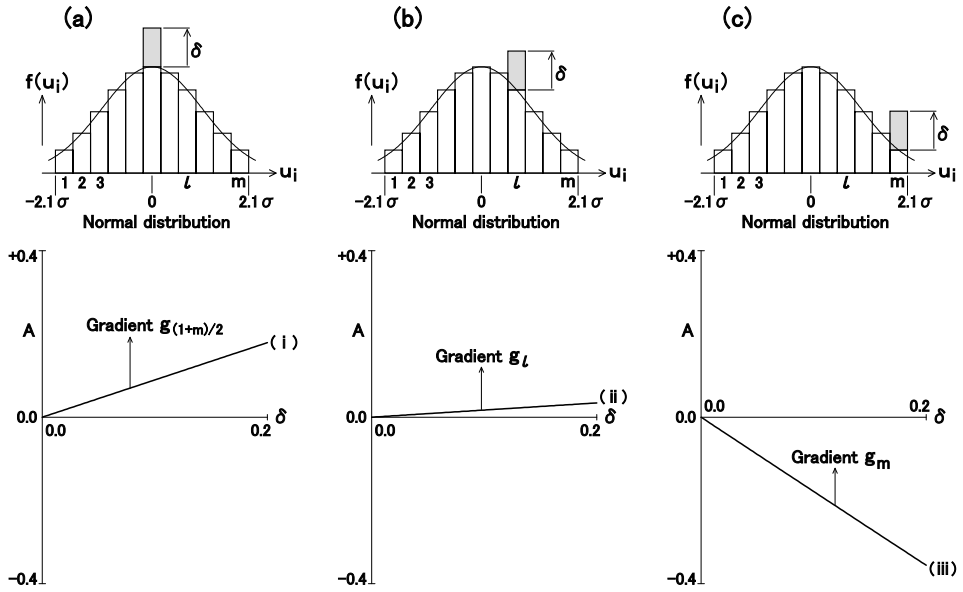

Fig. 2. Change of moment ratio A.

calculated using the following equation.

$$A = \frac{\left\{\sum_i f(u_i)\right\} \cdot \left\{\sum_i (u_i)^4 \cdot f(u_i)\right\}}{\left\{\sum_i (u_i)^2 \cdot f(u_i)\right\}^2} - 3 \tag{1}$$

Then, numerical experiments are carried out to study the relationship between moment ratio A and the increment value δ of bar graphs seen in Figs. 2–4. The upper side of graphs (a)–(c) of Figs. 2–4 shows the bar graphs each having m bars whose height is the same as function value $f(u_i)$ of the normal distribution. Note that $m = 11$ and the bar graphs are created by using the area of $-2.1\sigma \leq u_i \leq 2.1\sigma$ ($\sigma = 1$) of the normal distribution.[16] On bar graphs of Figs. 2(a)–(c), only a single bar increases by value δ in the center, an intermediate position, and an end of the normal distribution. In Figs. 3(a)–(c), two bars of the graph increase by the same value δ. Also, in Figs. 4(a)–(c), only one bar increases by value δ and another bar increases by value 0.2 at the same time. Here, the moment ratio A is calculated using Eq. (1) for the bar graph whose shape is changed as described above. The obtained relationship between values A and δ is shown by graphs (i) to (ix) in the lower side of graphs (a)–(c) of Figs. 2–4.

From graphs (i) to (iii) shown in Figs. 2(a)–(c), it is discovered that $A = 0.0$ if $\delta = 0.0$. Also, the value of A changes approximately linearly when value of δ increases. In Figs. 3(a)–(c), graphs (i)+(ii), (ii)+(iii), and (i)+(iii) are the results

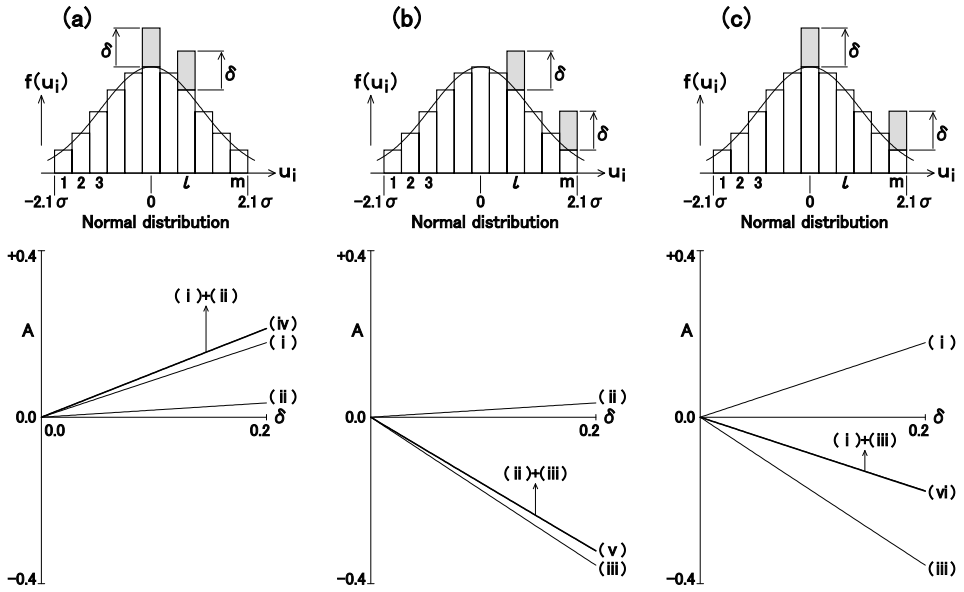

Fig. 3. Change of moment ratio A.

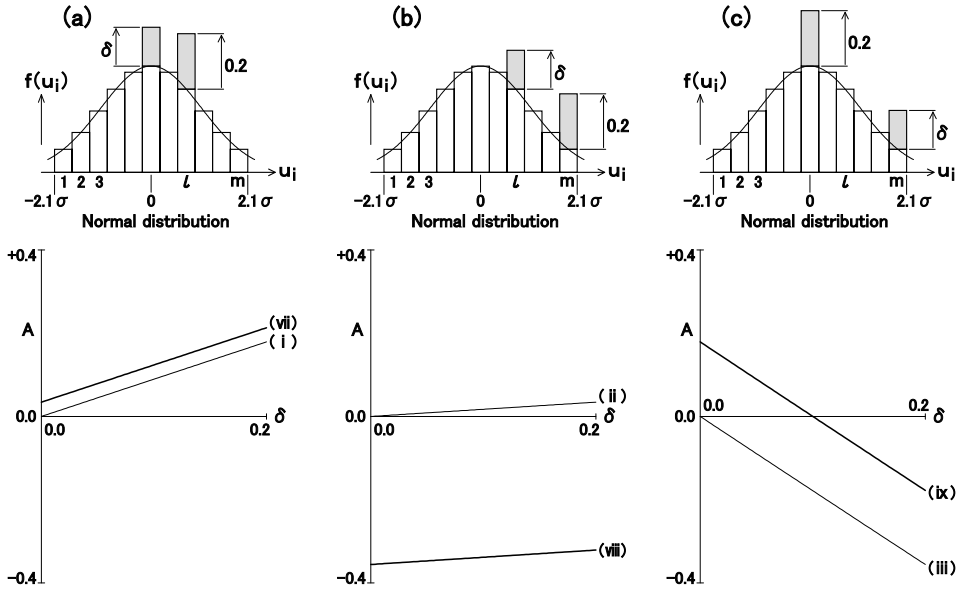

Fig. 4. Change of moment ratio A.

obtained by addition of graphs (i), (ii) and (iii) respectively. From these graphs, it is discovered that graphs (iv), (v) and (vi) are approximated to respective graphs (i)+(ii), (ii)+(iii), and (i)+(iii). Also, from Figs. 4(a)–(c), it is discovered that the

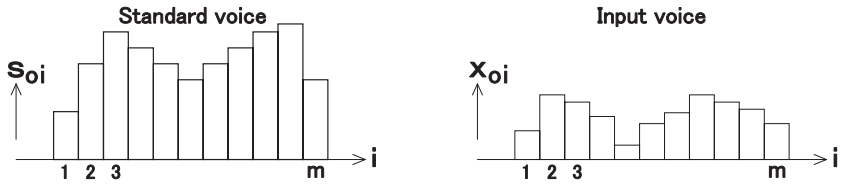

Fig. 5. Power spectra of standard voice and input voice.

gradients of graphs (vii), (viii) and (ix) are equal to those of graphs (i), (ii) and (iii) respectively, and that the intercepts on the vertical axis are equal to the change amounts of moment ratio A if $\delta = 0.2$ on graphs (ii), (iii) and (i) respectively.

From the above description, it is discovered that we can plot approximate graphs (iv) to (ix) using graphs (i) to (iii) if we have already plotted graphs (i) to (iii) using Eq. (1) in advance. In other words, if the rate of change g_i $(i = 1, 2, \cdots, m)$ of moment ratio A is calculated in advance based on the gradients of graphs (i) to (iii), we can determine the product of g_i multiplied by δ_i for each bar graph even when multiple bar graphs change by different values δ_i. Also, we can calculate an approximate value of moment ratio A by summing $g_i \cdot \delta_i$ for all i. This property holds for all values of m and for any variance σ^2 of the normal distribution.

3.2. *Creation of pattern vectors*

Fig. 5 gives an example of the power spectrum of standard and input voices. Note that the power spectrum is generated from the output of filter bank with the m frequency bands (where, m is an odd number). If the i-th power spectrum values (where, $i = 1, 2, \cdots, m$) of standard and input voices are s_{oi} and x_{oi} respectively, we create an original standard pattern vector $\boldsymbol{s_o}$ having s_{oi} components, and an original input pattern vector $\boldsymbol{x_o}$ having x_{oi} components, and represent them as follows. In Eq. (2), the function of "T" means a transposed matrix.

$$\begin{aligned} \boldsymbol{s_o} &= (s_{o1}, s_{o2}, \cdots, s_{oi}, \cdots, s_{om})^T \\ \boldsymbol{x_o} &= (x_{o1}, x_{o2}, \cdots, x_{oi}, \cdots, x_{om})^T \end{aligned} \tag{2}$$

Moreover, the component values s_{oi} and x_{oi} are divided by the summation of s_{oi} and the summation of x_{oi} respectively, and normalized power spectra s_i and x_i have been calculated. Then, we create a standard pattern vector $\boldsymbol{s}$ having s_i components, and an input pattern vector $\boldsymbol{x}$ having x_i components, and represent them as follows.

$$\begin{aligned} \boldsymbol{s} &= (s_1, s_2, \cdots, s_i, \cdots, s_m)^T \\ \boldsymbol{x} &= (x_1, x_2, \cdots, x_i, \cdots, x_m)^T \end{aligned} \tag{3}$$

If we assign constants c_s and c_x to the summation of s_{oi} and the summation of x_{oi} respectively in Eq. (2), we can show the relationship between component values of

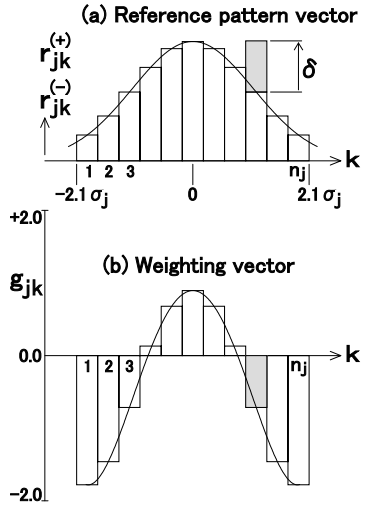

Fig. 6. Creating weighting vector.

Eqs. (2) and (3) as follows.

$$\begin{aligned} s_i &= s_{oi}/c_s \\ x_i &= x_{oi}/c_x \qquad (i = 1, 2, 3, \cdots, m) \end{aligned} \tag{4}$$

Also, the component values s_{oi} and x_{oi} are divided by the maximum value of s_{oi} and the maximum value of x_{oi} respectively, and normalized power spectra s'_i and x'_i have been calculated. Then, we create a standard pattern vector $\boldsymbol{s'}$ having s'_i components, and an input pattern vector $\boldsymbol{x'}$ having x'_i components, and represent them as follows.

$$\begin{aligned} \boldsymbol{s'} &= (s'_1, s'_2, \cdots, s'_i, \cdots, s'_m)^T \\ \boldsymbol{x'} &= (x'_1, x'_2, \cdots, x'_i, \cdots, x'_m)^T \end{aligned} \tag{5}$$

If we assign constants c'_s and c'_x to the maximum value of s_{oi} and the maximum value of x_{oi} respectively in Eq. (2), we can show the relationship between component values of Eqs. (2) and (5) as follows.

$$\begin{aligned} s'_i &= s_{oi}/c'_s \\ x'_i &= x_{oi}/c'_x \qquad (i = 1, 2, 3, \cdots, m) \end{aligned} \tag{6}$$

Eqs. (2),(3) and (5) express the shapes of the power spectra of the standard voice and input voice by the m pieces of component values of the pattern vector respectively. Note that in this paper the width of each bar graph is $1/m$ for power spectrum shown in Fig. 5. The area and the maximum values usually differ between $\boldsymbol{s_o}$ and $\boldsymbol{x_o}$ shown in Fig. 5. Meanwhile, the area of $\boldsymbol{s}$ and $\boldsymbol{x}$ are the same and the maximum values of $\boldsymbol{s'}$ and $\boldsymbol{x'}$ are the same.

3.3. *Creation of weighting vector*

From the conventional algorithm, as shown in Fig. 6(a), we created positive and negative reference pattern vectors $\boldsymbol{r}_j^{(+)}$ and $\boldsymbol{r}_j^{(-)}$ having function values $r_{jk}^{(+)}$ and $r_{jk}^{(-)}$ of the normal distribution as components for each movement position j, and represented them as follows.

$$\begin{aligned} \boldsymbol{r}_j^{(+)} &= (r_{j1}^{(+)}, r_{j2}^{(+)}, \cdots, r_{jk}^{(+)}, \cdots, r_{jn_j}^{(+)})^T \\ \boldsymbol{r}_j^{(-)} &= (r_{j1}^{(-)}, r_{j2}^{(-)}, \cdots, r_{jk}^{(-)}, \cdots, r_{jn_j}^{(-)})^T \\ & \qquad (j = 1, 2, 3, \cdots, m) \end{aligned} \tag{7}$$

Figs. 6(a) and (b) show the rate of change of A (g_{jk}, where a change of δ occurs at the k-th position, $k = 1, 2, \cdots, n_j$) for a normal distribution and a single instance of δ. Note that each bar graph has n_j bars. The rate of change g_{jk} is described by the following equation.

$$\begin{aligned} \mathrm{g}_{jk} = A/\delta \qquad & (k = 1, 2, 3, \cdots, n_j\,) \\ & (j = 1, 2, 3, \cdots, m\,) \end{aligned} \tag{8}$$

The $\mathrm{g}_{j(1+n_j)/2}$, g_{jl} and g_{jn_j} correspond to the gradients of respective graphs shown in the lower side of Figs. 2(a)–(c). Next, in Fig. 6(a), position k of the bar that has increased by value δ is scanned from 1 to n_j, and Eq. (8) is calculated. Fig. 6(b) shows a bar graph of the calculated value g_{jk}, where $\delta = 0.2$. Here, we create a weighting vector $\mathbf{g}_j$ having g_{jk} components, and represent it as follows.

$$\begin{aligned} \mathbf{g}_j &= (\mathrm{g}_{j1}, \mathrm{g}_{j2}, \cdots, \mathrm{g}_{jk}, \cdots, \mathrm{g}_{jn_j})^T \\ & \qquad (j = 1, 2, 3, \cdots, m) \end{aligned} \tag{9}$$

Eq. (9) expresses the rate of change of moment ratio A by the m pieces of component values of the vector. As $\boldsymbol{r}_j^{(+)}$ and $\boldsymbol{r}_j^{(-)}$ are equivalent vectors in the initial state, the weighting vector calculated from $\boldsymbol{r}_j^{(+)}$ and the weighting vector calculated from $\boldsymbol{r}_j^{(-)}$ are equal to each other. Thus, symbols (+) and (–) are omitted in Eq. (9). Also, the curve shown in Fig. 6(b) is the envelope curve of the g_{jk} bar graph, that has been calculated assuming the value n_j is sufficiently large, and it is called "Weighting curve" in this paper. As shown in Figs. 6(a) and (b), the normal curve corresponds to the weighting curve, and the positive and negative reference pattern vectors correspond to the weighting vector.

3.4. *Approximate calculation of moment ratio*

With the conventional algorithm, a difference in shapes between standard pattern vector $\boldsymbol{s}$ and input pattern vector $\boldsymbol{x}$ has been replaced by the shape changes of positive and negative reference pattern vectors $\boldsymbol{r}_j^{(+)}$ and $\boldsymbol{r}_j^{(-)}$ using the following

equation.

$$
\begin{aligned}
&For\ \ i=1,2,3,\cdots,m\,;\\
&\quad when\ \ k=i-j+(1+n_j)/2 \quad (where, 1\leq k\leq n_j)\,;\\
&\quad\ \bullet\ if\ x_i > s_i,\ then\ r_{jk}^{(+)} \longleftarrow r_{jk}^{(+)} + |x_i - s_i|\\
&\quad\ \bullet\ if\ x_i < s_i,\ then\ r_{jk}^{(-)} \longleftarrow r_{jk}^{(-)} + |x_i - s_i| \qquad (10)\\
&\qquad\qquad\qquad\qquad\qquad\qquad (j=1,2,3,\cdots,m)
\end{aligned}
$$

With the conventional algorithm, moment ratios of $\boldsymbol{r}_j^{(+)}$ and $\boldsymbol{r}_j^{(-)}$, whose shapes have changed according to Eq. (10), have been calculated using the following equation.

$$
\begin{aligned}
A_j^{(+)} &= \frac{\left\{\sum_{k=1}^{n_j} r_{jk}^{(+)}\right\}\cdot\left\{\sum_{k=1}^{n_j}(L_{jk})^4\cdot r_{jk}^{(+)}\right\}}{\left\{\sum_{k=1}^{n_j}(L_{jk})^2\cdot r_{jk}^{(+)}\right\}^2} - 3\\
A_j^{(-)} &= \frac{\left\{\sum_{k=1}^{n_j} r_{jk}^{(-)}\right\}\cdot\left\{\sum_{k=1}^{n_j}(L_{jk})^4\cdot r_{jk}^{(-)}\right\}}{\left\{\sum_{k=1}^{n_j}(L_{jk})^2\cdot r_{jk}^{(-)}\right\}^2} - 3 \qquad (11)\\
&\qquad\qquad\qquad\qquad (j=1,2,3,\cdots,m)
\end{aligned}
$$

In Section 3.1, we determined the product value $g_{jk}\bullet|x_i - s_i|$ using the rate of change g_{jk} of moment ratio A and increment $|x_i - s_i|$, and demonstrated that we can calculate the approximate value of the moment ratio A by summing $g_{jk}\bullet|x_i - s_i|$ for all i. Thus, the values $A_j^{(+)}$ and $A_j^{(-)}$ of Eq. (11) can be calculated approximately using the following equation.

$$
\begin{aligned}
&When\ \ k=i-j+(1+n_j)/2 \quad (where, 1\leq k\leq n_j)\,;\\
&\ \bullet\ for\ \ all\ \ i\ \ where\ \ x_i > s_i\\
&\qquad A_j^{(+)} \approx \sum_{i=1}^{m} g_{jk}\bullet|x_i - s_i|\\
&\ \bullet\ for\ \ all\ \ i\ \ where\ \ x_i < s_i\\
&\qquad A_j^{(-)} \approx \sum_{i=1}^{m} g_{jk}\bullet|x_i - s_i| \qquad (12)\\
&\qquad\qquad\qquad (j=1,2,3,\cdots,m)
\end{aligned}
$$

If value of k does not satisfy $1 \leq k \leq n_j$, we assume $g_{jk} = 0$. Next, we consider the signs and replace $|x_i - s_i|$ by $(x_i - s_i)$, and rewrite Eq. (12) as follows.

$$\begin{aligned} &When\ \ k = i - j + (1 + n_j)/2 \quad (where, 1 \leq k \leq n_j)\,; \\ &\bullet\ for\ \ all\ \ i\ \ where\ \ x_i > s_i \\ &\qquad A_j^{(+)} \approx + \sum_{i=1}^{m} g_{jk} \cdot (x_i - s_i) \\ &\bullet\ for\ \ all\ \ i\ \ where\ \ x_i < s_i \\ &\qquad A_j^{(-)} \approx - \sum_{i=1}^{m} g_{jk} \cdot (x_i - s_i) \qquad (13) \\ &\qquad\qquad\qquad\qquad (j = 1, 2, 3, \cdots, m) \end{aligned}$$

The approximate value of moment ratio can be calculated by product-sum operation using Eq. (13), instead of calculating the moment ratio directly using Eq. (11).

3.5. *Approximate calculation of shape variation*

From the conventional algorithm, the difference in shapes between standard and input patterns has been calculated using the following equation, and it has been defined as "Shape variation D_j".

$$D_j = A_j^{(+)} - A_j^{(-)} \qquad (j = 1, 2, 3, \cdots, m) \qquad (14)$$

Thus, the value D_j of Eq. (14) can be calculated approximately by substituting Eq. (13) into Eq. (14) as follows.

$$\begin{aligned} &When\ \ k = i - j + (1 + n_j)/2 \quad (where, 1 \leq k \leq n_j)\,; \\ &\qquad D_j \approx \sum_{i=1}^{m} g_{jk} \cdot (x_i - s_i) \\ &\qquad\quad = \sum_{i=1}^{m} g_{jk} \cdot x_i - \sum_{i=1}^{m} g_{jk} \cdot s_i \qquad (15) \\ &\qquad\qquad\qquad\qquad (j = 1, 2, 3, \cdots, m) \end{aligned}$$

From Eq. (15), it is discovered that the value D_j can be separated into the product-sum operation using the component value g_{jk} of weighting vector and the component value x_i of input pattern vector, and the product-sum operation using the component value g_{jk} and the component value s_i of standard pattern vector.

3.6. *Creation of weighted pattern vectors*

We assign $s_{g(j)}$ and $x_{g(j)}$ to the two product-sum operations given by Eq. (15) respectively, and represent them as follows.

$$\begin{aligned} &When\ \ k=i-j+(1+n_j)/2 \quad (where, 1\leq k\leq n_j)\,; \\ &\qquad s_{g(j)} = \sum_{i=1}^{m} g_{jk}\cdot s_i \\ &\qquad x_{g(j)} = \sum_{i=1}^{m} g_{jk}\cdot x_i \\ &\qquad\qquad (j=1,2,3,\cdots,m) \end{aligned} \tag{16}$$

Then, we create a weighted standard pattern vector $\boldsymbol{s_g}$ having $s_{g(j)}$ components, and a weighted input pattern vector $\boldsymbol{x_g}$ having $x_{g(j)}$ components, and represent them as follows.

$$\begin{aligned} \boldsymbol{s_g} &= (s_{g(1)}, s_{g(2)}, \cdots, s_{g(j)}, \cdots, s_{g(m)})^T \\ \boldsymbol{x_g} &= (x_{g(1)}, x_{g(2)}, \cdots, x_{g(j)}, \cdots, x_{g(m)})^T \end{aligned} \tag{17}$$

From Eqs. (15) and (16), the value D_j can be represented approximately as follows.

$$D_j \approx x_{g(j)} - s_{g(j)} \qquad (j=1,2,3,\cdots,m) \tag{18}$$

From Eq. (18), it is discovered that the value D_j can be obtained by subtracting the component value $s_{g(j)}$ of weighted standard pattern vector from the component value $x_{g(j)}$ of weighted input pattern vector.

3.7. *Approximate calculation of geometric distance*

Using the conventional algorithm, we have calculated the difference in shapes between standard and input patterns using the following equation and we have defined it as the "Geometric distance d".

$$d = \sqrt{\sum_{j=1}^{m} (D_j)^2} \tag{19}$$

Thus, the value d of Eq. (19) can be calculated approximately by substituting Eq. (18) into Eq. (19) as follows. Note that $\widetilde{d}$ is an approximate value of the geometric distance d.

$$d \approx \sqrt{\sum_{j=1}^{m} (x_{g(j)} - s_{g(j)})^2} = \widetilde{d} \tag{20}$$

As described above, the value $\widetilde{d}$ can be calculated by using Eqs. (3), (9), (16), and (20) sequentially. From Eqs. (16) and (20), we can find that the value $\widetilde{d}$ can be separated into the product-sum operation using the standard pattern vector and the product-sum operation using the input pattern vector.

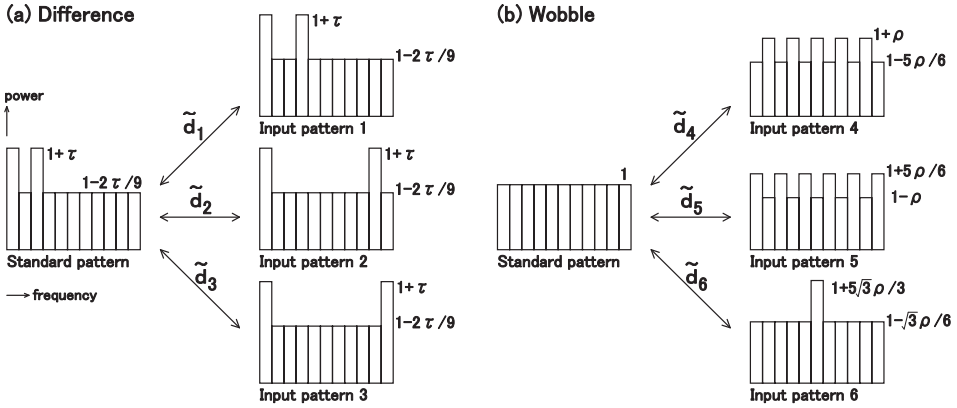

Fig. 7. Typical examples of standard and input patterns.

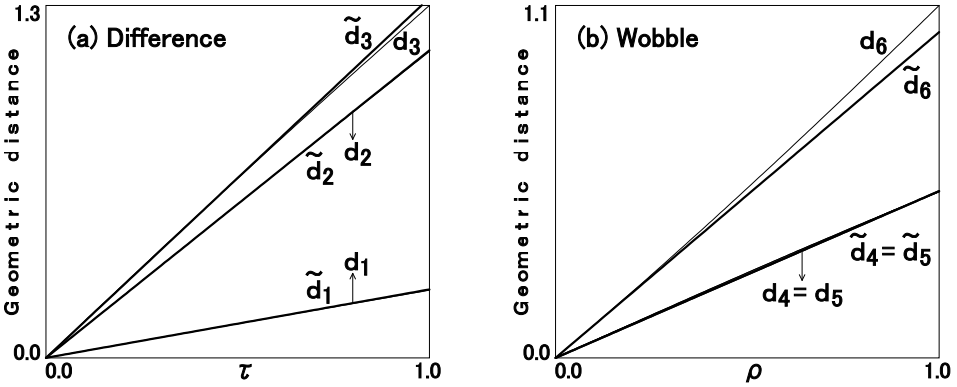

Fig. 8. Calculations in geometric distances d and $\widetilde{d}$.

3.8. *Numerical experiments of geometric distance* $\widetilde{d}$

To confirm the approximation accuracy of $\widetilde{d}$ shown in Eq. (20), we performed numerical experiments to calculate the geometric distances[16] d_1 to d_6 and the approximate values $\widetilde{d}_1$ to $\widetilde{d}_6$ of the standard and input patterns shown in Fig. 7. However, we have developed Eqs. (7) and (9) by using values $n_j = 27$ ($\sigma_j = n_j/(4.2m) = 0.58$)[16] that are fixed regardless of movement position j of the normal distribution. Figs. 8(a) and (b) show the results of experiments. We can find that values d_1 to d_6 and $\widetilde{d}_1$ to $\widetilde{d}_6$ are almost identical.

3.9. *Creation of original and weighted pattern vectors*

We assign $s_{og(j)}$ to the product-sum operation using the component value g_{jk} of weighting vector and the component value s_{oi} of original standard pattern vector given by Eq. (2), and assign $x_{og(j)}$ to the product-sum operation using the compo-

nent value g_{jk} and the component value x_{oi} of original input pattern vector, and represent them as follows.

$$\begin{aligned} &When\ \ k=i-j+(1+n_j)/2 \quad (where, 1\leq k\leq n_j)\,; \\ &\qquad s_{og(j)} = \sum_{i=1}^{m} g_{jk}\cdot s_{oi} \\ &\qquad x_{og(j)} = \sum_{i=1}^{m} g_{jk}\cdot x_{oi} \\ &\qquad\qquad (j=1,2,3,\cdots,m) \end{aligned} \tag{21}$$

Then, we create an original and weighted standard pattern vector $\boldsymbol{s_{og}}$ having $s_{og(j)}$ components, and an original and weighted input pattern vector $\boldsymbol{x_{og}}$ having $x_{og(j)}$ components, and represent them as follows:

$$\begin{aligned} \boldsymbol{s_{og}} &= (s_{og(1)}, s_{og(2)}, \cdots, s_{og(j)}, \cdots, s_{og(m)})^T \\ \boldsymbol{x_{og}} &= (x_{og(1)}, x_{og(2)}, \cdots, x_{og(j)}, \cdots, x_{og(m)})^T \end{aligned} \tag{22}$$

Eq. (22) shows the original and weighted pattern vectors that are created without normalization of the power spectrum. Also, we assign $s'_{g(j)}$ to the product-sum operation using g_{jk} and s'_i given by Eq. (5), and assign $x'_{g(j)}$ to the product-sum operation using g_{jk} and x'_i, and represent them as follows.

$$\begin{aligned} &When\ \ k=i-j+(1+n_j)/2 \quad (where, 1\leq k\leq n_j)\,; \\ &\qquad s'_{g(j)} = \sum_{i=1}^{m} g_{jk}\cdot s'_i \\ &\qquad x'_{g(j)} = \sum_{i=1}^{m} g_{jk}\cdot x'_i \\ &\qquad\qquad (j=1,2,3,\cdots,m) \end{aligned} \tag{23}$$

Then, we create a weighted standard pattern vector $\boldsymbol{s'_g}$ having $s'_{g(j)}$ components, and a weighted input pattern vector $\boldsymbol{x'_g}$ having $x'_{g(j)}$ components, and represent them as follows:

$$\begin{aligned} \boldsymbol{s'_g} &= (s'_{g(1)}, s'_{g(2)}, \cdots, s'_{g(j)}, \cdots, s'_{g(m)})^T \\ \boldsymbol{x'_g} &= (x'_{g(1)}, x'_{g(2)}, \cdots, x'_{g(j)}, \cdots, x'_{g(m)})^T \end{aligned} \tag{24}$$

Eq. (24) shows the weighted pattern vectors that are created with normalization of power spectrum using their maximum values.

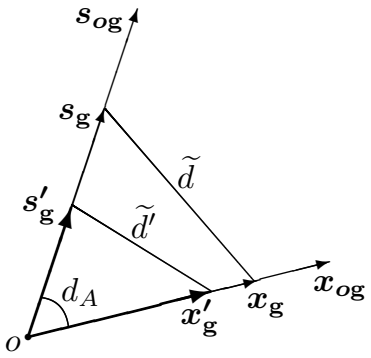

Fig. 9. Relationship among weighted pattern vectors.

3.10. *Relationship among weighted pattern vectors*

Eq. (4) is substituted into Eq. (16), and the following equation is obtained using Eq. (21).

$$\begin{aligned} &When\ \ k=i-j+(1+n_j)/2 \quad (where, 1\le k\le n_j)\,; \\ &\quad s_{g(j)} = \sum_{i=1}^{m} g_{jk}\cdot(s_{oi}/c_s) \\ &\qquad = s_{og(j)}/c_s \\ &\quad x_{g(j)} = \sum_{i=1}^{m} g_{jk}\cdot(x_{oi}/c_x) \\ &\qquad = x_{og(j)}/c_x \qquad\qquad (j=1,2,3,\cdots,m) \end{aligned} \tag{25}$$

Similarly, Eq. (6) is substituted into Eq. (23), and the following equation is obtained using Eq. (21).

$$\begin{aligned} s'_{g(j)} &= s_{og(j)}/c'_s \\ x'_{g(j)} &= x_{og(j)}/c'_x \qquad (j=1,2,3,\cdots,m) \end{aligned} \tag{26}$$

Fig. 9 is a schematic diagram of the m-th dimensional pattern space, and it shows six vectors, those are $\boldsymbol{s_{og}}$ and $\boldsymbol{x_{og}}$ given by Eq. (22), $\boldsymbol{s_g}$ and $\boldsymbol{x_g}$ given by Eq. (17), and $\boldsymbol{s'_g}$ and $\boldsymbol{x'_g}$ given by Eq. (24). Note that all vectors begin at origin o. From Eq. (25), we can understand that $s_{g(j)}$ and $s_{og(j)}$ are proportional with constant $1/c_s$, and that $x_{g(j)}$ and $x_{og(j)}$ are proportional with constant $1/c_x$. Also, from Eq. (26), we can understand that $s'_{g(j)}$ and $s_{og(j)}$ are proportional with constant $1/c'_s$, and that $x'_{g(j)}$ and $x_{og(j)}$ are proportional with constant $1/c'_x$. Therefore, as shown in Fig. 9, vectors $\boldsymbol{s'_g}$, $\boldsymbol{s_g}$ and $\boldsymbol{s_{og}}$ have the same direction. Also, vectors $\boldsymbol{x'_g}$, $\boldsymbol{x_g}$ and $\boldsymbol{x_{og}}$ have the same direction.

3.11. *Derivation of new geometric distance*

From Eq. (20), it is clear that the geometric distance $\widetilde{d}$ can be calculated as the Euclidean distance between the weighted standard pattern vector $\boldsymbol{s_g}$ and the weighted input pattern vector $\boldsymbol{x_g}$. Thus, in Fig. 9, we determine the distance between end points of $\boldsymbol{s_g}$ and $\boldsymbol{x_g}$ as value $\widetilde{d}$. Also, if we use Eq. (5) instead of Eq. (3) to determine the standard and input pattern vectors, the geometric distance $\widetilde{d}'$ can be calculated as the Euclidean distance between $\boldsymbol{s'_g}$ and $\boldsymbol{x'_g}$. Thus, in Fig. 9, we determine the distance between end points of $\boldsymbol{s'_g}$ and $\boldsymbol{x'_g}$ as value $\widetilde{d}'$. From Fig. 9, it is clear that values $\widetilde{d}$ and $\widetilde{d}'$ are changed according to the normalizing method used. To improve on this, we can calculate an angle d_A between $\boldsymbol{s_{og}}$ and $\boldsymbol{x_{og}}$ shown in Fig. 9 by the following equation and we define it as the new "Geometric distance d_A".

$$\cos(d_A) = \frac{\sum_{j=1}^{m} s_{og(j)} \cdot x_{og(j)}}{\sqrt{\sum_{j=1}^{m} (s_{og(j)})^2} \sqrt{\sum_{j=1}^{m} (x_{og(j)})^2}} \tag{27}$$

The geometric distance d_A is not affected by the normalizing method used. If d_A is used, we can expect that the shortcoming of the pseudo difference in shapes between the standard and input patterns due to normalization of power spectrum is improved and the recognition performance becomes stable. Therefore, in order to confirm that d_A matches the mathematical model, we perform numerical experiments in Section 3.14. Also, to confirm the stabilized recognition performance of d_A, we carry out the speech recognition tests in Section 4.

3.12. *Sharing weighting vector*

In Eq. (7), we have created the m pieces of positive and negative reference pattern vectors (normal curves). Fig. 10(a) gives an example of three normal curves among these curves. Note that the center axis of the normal curve is drawn in component position j. In Eq. (9), we have created the m weighting vectors (weighting curves) from Eq. (7) as shown in Fig. 6. The weighting curves created from every normal curve in Fig. 10(a) are shown in Fig. 10(b). This paper uses a fixed bar width of each graph for both standard and input patterns even when the variance value of the normal distribution has changed. In which case, as shown in Fig. 10(b), the maximum and minimum values of those weighting curves are the same respectively, and those weighting curves match when expanded or compressed in the direction of the horizontal axis. Thus, we thought to reduce the computational memory overhead by sharing a single weighting vector instead of m vectors. Fig. 10(c) shows the weighting curve that has been created from the normal curve of variance $\sigma^2 = 1$. Fig. 10(c) also shows a bar graph having the same height as the function value of weighting curve. Here, the right half of the weighting curve is used to create

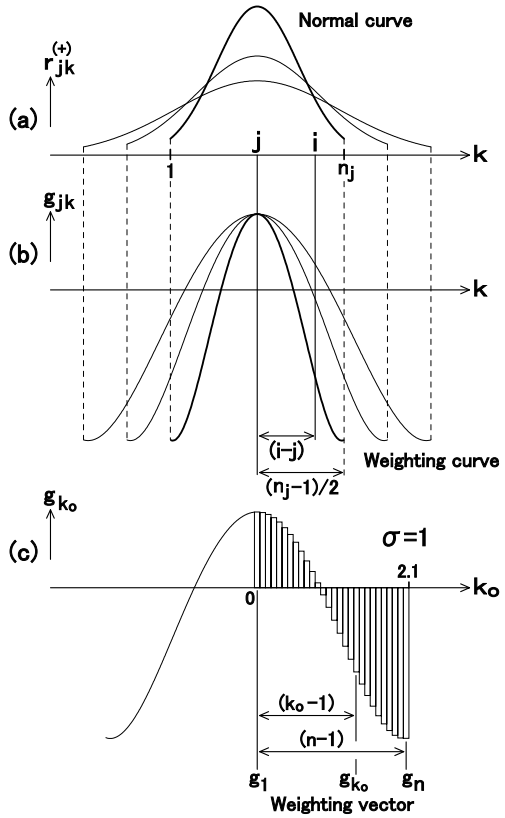

Fig. 10. Sharing weighting vector.

a bar graph for reducing the computational memory overhead. And we create a weighting vector $\mathbf{g}$ having g_{k_0} (where, $k_0 = 1, 2, \cdots, n$) components whose values are the same as the height of bar graph, and represent it as follows.

$$\mathbf{g} = (g_1, g_2, \cdots, g_{k_0}, \cdots, g_n, 0, \cdots, 0)^T \tag{28}$$

However, we assume that value n is sufficiently large when compared with the number of components n_j of Eq. (9). Also, if $n < k_0$, we insert an appropriate number of values $g_{k_0} = 0$. Eq. (28) is the weighting vector that represents Eq. (9), and Eq. (28) consists of both n components expressing the shape of weighting curve and an appropriate number of component values 0.

As shown by the thick-line weighting curve of Fig. 10(b), the difference between component numbers i and j is $(i - j)$ for the weighting vector $\mathbf{g}_j$ given by Eq. (9). The difference between the component number at the center and the component number at the rightmost end position is $(n_j - 1)/2$. On the other hand, as shown in Fig. 10(c), the difference between component numbers k_0 and 1 is $(k_0 - 1)$ and the difference between component numbers n and 1 is $(n - 1)$ for the weighting vector $\mathbf{g}$ given by Eq. (28). As described above, each weighting curve of Fig. 10(b)

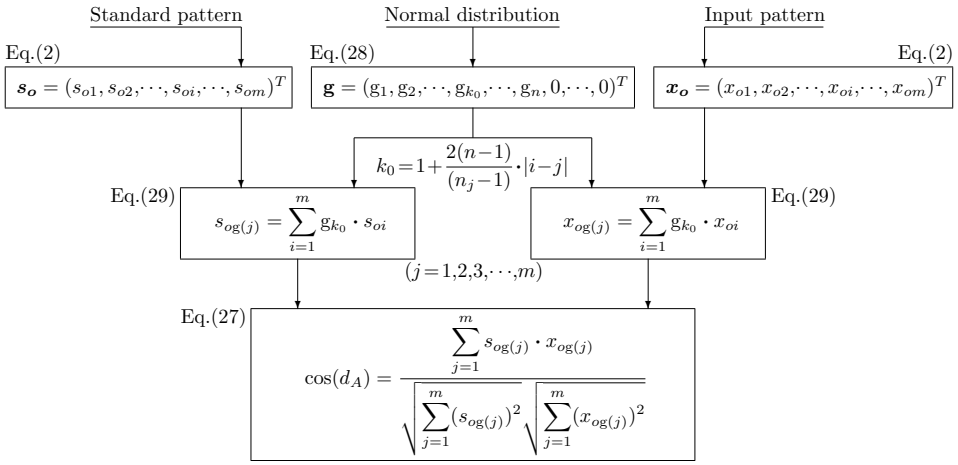

Fig. 11. Flowchart for calculating geometric distance d_A.

can be obtained by expanding or compressing the weighting curve of Fig. 10(c) in the direction of the horizontal axis. Therefore, if the component number i of Figs. 10(a) and (b) corresponds to k_0 of Fig. 10(c), the ratio of $(i-j)$ to $(n_j-1)/2$ is equal to the ratio of (k_0-1) to $(n-1)$. $2(i-j)/(n_j-1) = (k_0-1)/(n-1)$ is satisfied. If we consider that the weighting curve is bilaterally symmetric, we can calculate value k_0 using equation $k_0 = 1 + 2|i-j| \cdot (n-1)/(n_j-1)$. Note that k_0 is rounded to an integer value. If value n is sufficiently large, we can reduce the rounding error. In this way, the values $s_{og(j)}$ and $x_{og(j)}$ can be calculated by using the following equation instead of Eq. (21).

$$
\begin{aligned}
When\ \ k_0 &= 1 + \frac{2(n-1)}{(n_j-1)} \cdot |i-j|\,; \\
s_{og(j)} &= \sum_{i=1}^{m} g_{k_0} \cdot s_{oi} \\
x_{og(j)} &= \sum_{i=1}^{m} g_{k_0} \cdot x_{oi} \qquad (29) \\
&\qquad\qquad (j = 1, 2, 3, \cdots, m)
\end{aligned}
$$

Note that the component number k_1 of Eq. (21) corresponds to k_0 of Fig. 10(c) or Eq. (29). Using Eq. (29), we can calculate both $s_{og(j)}$ and $x_{og(j)}$ by simply creating a single $\mathbf{g}$ instead of creating $\mathbf{g}_j$ for each movement position j of the normal distribution. In this manner, the computational memory of $\mathbf{g}$ is fixed to the value n in Eq. (28). While in Eq. (9), the memory of $\mathbf{g}_j$ increased in proportion to the square of the value m (in proportion to the value $n_j \times m$). This paper assumes that $n = 2101$. As described above, we can reduce the computational memory overhead by sharing a single weighting vector.

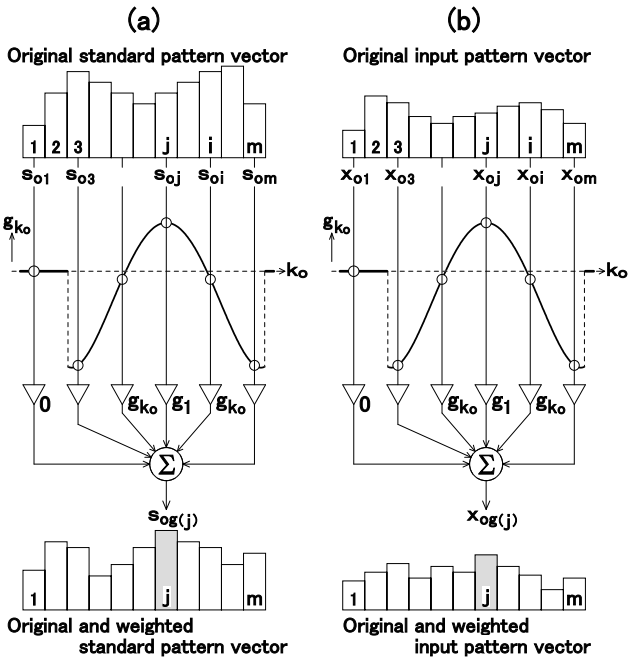

Fig. 12. Diagram for calculating product-sum value.

3.13. *Procedure for calculating geometric distance* d_A

Fig. 11 shows a flowchart for calculating the new geometric distance d_A. From Fig. 11, it is clear that we can calculate the value $s_{og(j)}$ in advance during the standard pattern registration process. Moreover, Figs. 12(a) and (b) show the flow of product-sum operations given by Eq. (29). Note that the curve in the figure is the weighting curve shown in Fig. 10(c), and symbol $\bigtriangledown$ is a multiplier and symbol Σ is an adder. In Fig. 12(a), by using multiplier $\bigtriangledown$, we calculate the product $g_{k_0} \cdot s_{oi}$ using the component value g_{k_0} of weighting vector and the component value s_{oi} of original standard pattern vector. By using adder Σ, we calculate the product-sum by addition of the product $g_{k_0} \cdot s_{oi}$ for i (where, $i = 1, 2, \cdots, m$), and use it as the component value $s_{og(j)}$ of original and weighted standard pattern vector. Similarly, in Fig. 12(b), we calculate the original and weighted input pattern vector by the product-sum operation using the original input pattern vector and the weighting vector. From Figs. 12(a) and (b), it is discovered that the values $s_{og(j)}$ and $x_{og(j)}$ are calculated from s_{oi} and x_{oi}, respectively, by weighting of the weighting curve.

Figs. 13(a) and (b) show a comparison between calculation amounts of the conventional algorithm and the new algorithm during the input pattern recognition process. From the conventional algorithm, if we calculate the geometric distances d between N standard patterns and a single input pattern, we need to calculate Eqs. (10), (11), (14) and (19) sequentially in each combination of standard and input patterns during the input pattern recognition process. With the new algorithm,

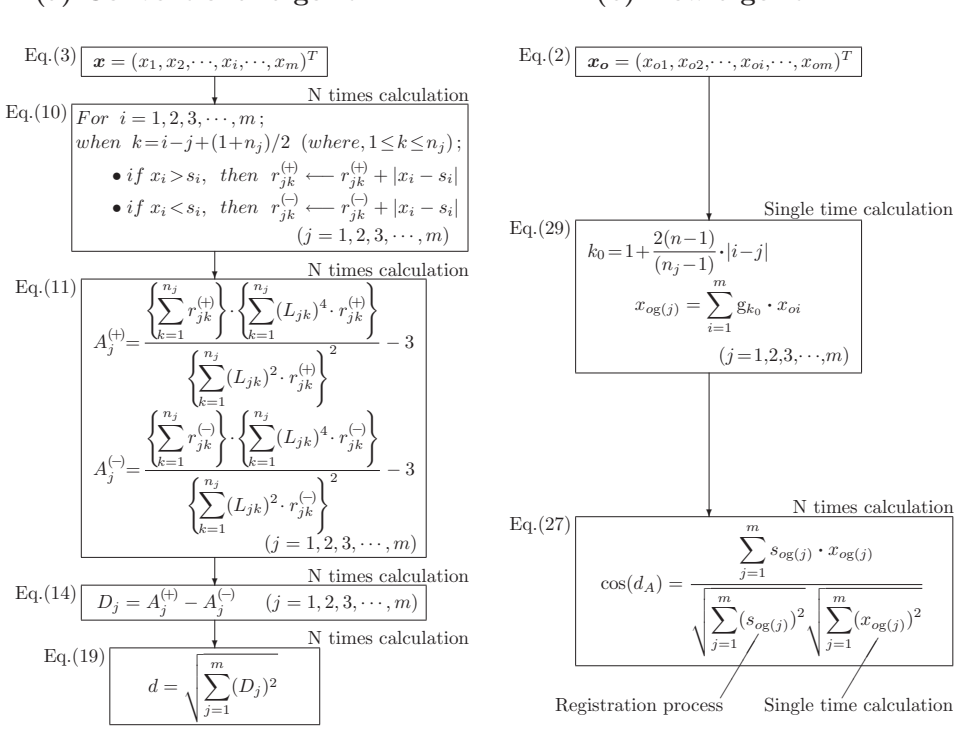

Fig. 13. Flowcharts for conventional algorithm and new algorithm during input pattern recognition process.

we can obtain the N parts of d_A values by performing a single time calculation of $x_{og(j)}$ and an N times of cosine similarity calculation during the input pattern recognition process. From Figs. 13(a) and (b), it is discovered that we can reduce the processing overhead during the input pattern recognition process.

3.14. *Numerical experiments of geometric distance d_A*

To confirm that the algorithm of geometric distance d_A matches the mathematical model that we have assumed in Section 1, we performed numerical experiments to calculate the geometric distances d_A of the standard and input patterns shown in Fig. 7. Note that we used the same n_j value as Section 3.8. Also, note that we read geometric distances $\widetilde{d}_1$ to $\widetilde{d}_6$ in Fig. 7 as new geometric distances d_{A1} to d_{A6} respectively.

Figs. 14(a) and (b) show the results of experiments. From the figures, we can find that $d_{A4} < d_{A5}$ in Fig. 14(b) although $\widetilde{d}_4 = \widetilde{d}_5$ in Fig. 8(b). Here, $m = 11$ for the standard and input patterns shown in Fig. 7. From the experiments, we found that the larger value was switched between d_{A4} and d_{A5} when value m increased. Also, the two graphs became close to position d_{A4} shown in Fig. 14(b). However, the

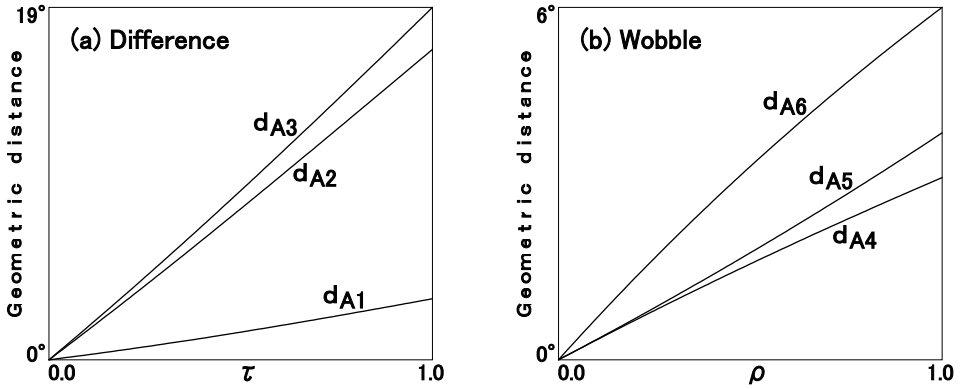

Fig. 14. Calculation in geometric distance d_A.

difference between d_{A4} and d_{A5} is small because we use $m = 23$ in the experiments of vowel recognition performed in the next section. From the numerical experiments described above, we can verify that the algorithm of geometric distance d_A matches the characteristics $<1>$ and $<2>$ of the mathematical model.

4. Experiments of Vowel Recognition

To confirm that the geometric distance d_A removes the pseudo difference in shapes and the recognition performance becomes stable, we have performed the speech recognition experiments using the geometric distance d_A and actual voices. We used the same Japanese speech and feature parameters as those used in the experiments with the conventional geometric distance algorithm.[16] Similar to the speech recognition experiments of the conventional algorithm, we performed the experiments in the following two stages.

(Stage 1) First, we optimized the variance of the normal distribution using the "vowel in the continuous speech" that is different from the voice data for the evaluation experiments.

(Stage 2) Next, we performed the evaluation experiments for the "clean vowel" and the "vowel with noise" by using the optimized normal distribution.

4.1. *Variance optimization of normal distribution*

Similar to the speech recognition experiments of the conventional algorithm, we determine the optimum value of the variance σ^2 of the normal distribution (the optimum value of ω)[16] using the "vowel in the continuous speech". This is equivalent to determining the optimum value of the positive and negative reference pattern vectors given by Eq. (7) and to determining the optimum value of the weighting vector $\mathbf{g}_j$ given by Eq. (9). And we convert $\mathbf{g}_j$ into $\mathbf{g}$ as shown in Figs. 10(b) and (c) and reduce the computational memory overhead. The value ω is incremented by 0.2 from 3.0 to 23.0, and the recognition accuracy of the "vowel in the continuous

Table 1. Vowel recognition accuracy with new geometric distance d_A. ($\omega = 11.0$)

	Babble	Car	Exhibition	Subway	Mean
Clean					99.97%
SNR 20 dB	99.93%	99.88%	99.22%	99.49%	99.63%
SNR 10 dB	98.80%	98.80%	88.77%	93.36%	94.93%
SNR 5 dB	92.34%	88.10%	67.96%	80.03%	82.11%

speech" is calculated. Fig. 15 shows the relationship between the value ω and the recognition accuracy obtained by the above process. From Fig. 15, it is discovered that the recognition accuracy becomes maximum if $\omega = 11.0$. Thus, we determine $\omega = 11.0$ as the optimum value and use it in the following evaluation experiments.

4.2. *Evaluation experiments and their results*

We have performed the evaluation experiments for the "clean vowel" and the "vowel with noise" using the value $\omega = 11.0$ determined in the previous section. Table 1 shows the result of vowel recognition using the new geometric distance d_A. From Table 1, it is learned that the recognition accuracy with d_A is equalized regardless of noise type[17] when compared with the conventional geometric distance d.[16] In particular, the recognition accuracy of "Exhibition5dB" has improved from 61.42% to 67.96%. Also, "mean" of 5 dB SNR has improved from 78.04% to 82.11%. Thus we confirm that the geometric distance d_A removes the pseudo difference in shapes and the recognition performance becomes stable.

4.3. *Verification of optimum value*

Table 1 shows the result of recognition accuracy using the optimum value $\omega = 11.0$ that we have determined from Fig. 15. Here, in order to verify that the value $\omega = 11.0$ is truly the optimum value, the value ω is incremented by 0.2 from 3.0 to 23.0 and the recognition accuracy of the "clean vowel" and the "vowel with noise" is calculated. Figs. 16 and 17 show the calculated relationship between the value ω and the recognition accuracy for the input patterns of the "clean vowel" and the "vowel with 5 dB noise", respectively. From Figs. 16 and 17, we can find that the recognition accuracy is almost maximum in the value $\omega = 11.0$.

5. Conclusions and Future Work

We have used the weighting vector that consists of the rate of change of the moment ratio, and created two weighted pattern vectors by performing the product-sum operation using the weighting vector and the standard pattern vector and the product-sum operation using the weighting vector and the input pattern vector. Then, we have proposed a new algorithm that uses the angle between these weighted pattern

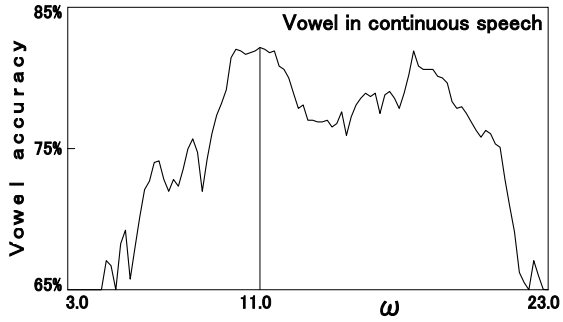

Fig. 15. Vowel recognition accuracy and optimum value ω.

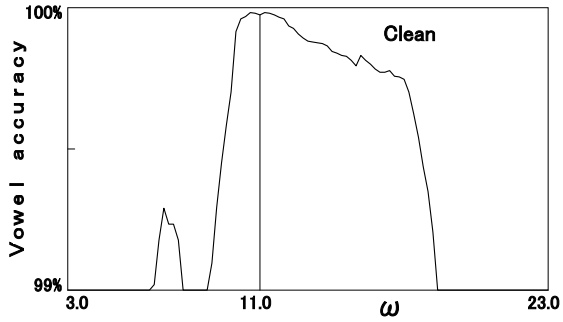

Fig. 16. Vowel recognition accuracy with new geometric distance d_A.

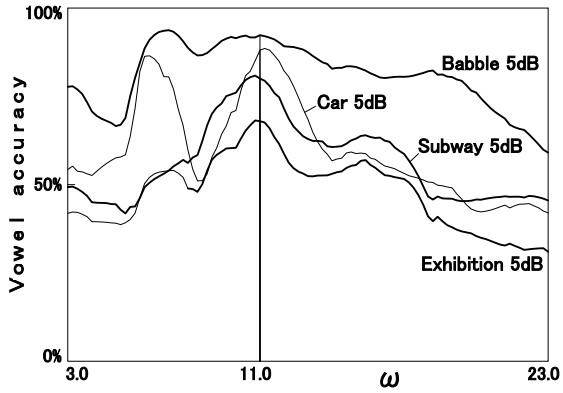

Fig. 17. Vowel recognition accuracy with new geometric distance d_A.

vectors as the geometric distance. At this time, we have evaluated the processing overhead and the computational memory required for the new algorithm. Also, we have performed the vowel recognition experiments, and confirmed that the recognition performance becomes stable.

Finally, we describe future work. For vowel recognition, we used the same ω

value ($\omega = 11.0$) for the optimum value regardless of the first, second and third formants when detecting the amount of "difference" between the formants. We will optimize the variance of the normal distribution for each formant and improve the recognition accuracy in future studies. Also, this paper uses the geometric distance for one-dimensional patterns. We extend the concept of Geometric Distance so that it can be applied to two-dimensional patterns such as voice prints and images.

Acknowledgments

This research has been partially supported by New Energy and Industrial Technology Development Organization (NEDO) of the Japanese Government under Grant No. 10HC7011, by Queensland Parks and Wildlife Service of the Australian Government under Coxen's Fig-parrot Recovery Plan, and funded by Mitsubishi Heavy Industries, Ltd. of Japan and Tokyo Gas Co., Ltd. of Japan, by West Nippon Expressway Engineering Shikoku Company Limited of Japan.

References

1. R.O. Duda, P.E. Hart and D.G. Stork. *Pattern Classification, second ed.*, Wiley, NewYork, 2000.
2. K.K. Paliwal. *Effect of preemphasis on vowel recognition performance*, Speech Communication, **3**, pp.101-106, 1984.
3. L.R. Rabiner and B.H. Juang. *Fundamentals of Speech Recognition*, Prentice Hall, Englewood Cliffs, New Jersey, 1993.
4. F. Itakura and S. Saito. *An analysis-synthesis telephony based on maximum likelihood method*, Proc. 6th Int. Congr. Acoustics, C-5-5, 1968.
5. F. Itakura. *Minimum prediction residual principle applied to speech recognition*, IEEE Trans. Acoust., Speech and Signal Processing, **23**, pp.67-72, 1975.
6. S. Furui. *Digital Speech Processing, Synthesis, and Recognition (Electrical and Computer Engineering)*, Marcel Dekker, Inc., NewYork, 1989.
7. K. Shikano and M. Sugiyama. *Evaluation of LPC spectral matching measures for spoken word recognition*, Trans. IECE, 565-D, **5**, pp.535-541 1982.
8. D. Klatt. *Prediction of perceived phonetic distance from critical band spectra: A first step*, Proc. ICASSP 82, **2**, pp.1278-1281, 1982.
9. D. Mansour and B.H. Juang. *A family of distortion measures based upon projection operation for robust speech recognition*, IEEE Trans. Acoustics, Speech and Signal Processing, ASSP-37, **11**, pp.1659-1671, 1989.
10. N. Nocerino, F.K. Soong, L.R. Rabiner and D.H. Klatt. *Comparative study of several distortion measures for speech recognition*, Speech Communication, **4**, pp.317-331, 1985.
11. S.-H. Cha and S.N. Srihari. *On measuring the distance between histograms*, Pattern Recognition, **35**, pp.1355-1370, 2002.
12. J.-K. Kamarainen, V. Kyrki, J. Ilonen and H. Kälviäinen. *Improving similarity measures of histograms using smoothing projections*, Pattern Recognition Lett., **24**, pp.2009-2019, 2003.
13. F.-D. Jou, K.-C. Fan and Y.-L. Chang. *Efficient matching of large-size histograms*, Pattern Recognition Lett., **25**, pp.277-286, 2004.
14. F. Serratosa and A. Sanfeliu. *Signatures versus histograms: Definitions, distances and algorithms*, Pattern Recognition, **39**, pp.921-934, 2006.
15. V.V. Strelkov. *A new similarity measure for histogram comparison and its application in time series analysis*, Pattern Recognition Lett., **29**, pp.1768-1774, 2008.
16. M. Jinnai, S. Tsuge, S. Kuroiwa, F. Ren and M. Fukumi. *New similarity scale to measure the difference in like patterns with noise*, IJAI, Volume 1, Number 1, pp.59-88, November, 2009.

17. H.G. Hirsch and D. Pearce. *The AURORA experimental framework for the performance evaluation of speech recognition systems under noisy conditions*, ISCA ITRW ASR2000, 2000.

Michihiro Jinnai (Member)

He received the B.S. degree in seismology from Kyoto University, Japan, the M.E. and Ph.D. degrees in speech recognition from Kobe University, Japan, in 1976, 1980, and 1983, respectively. He is currently a professor with the Department of Electro-Mechanical Systems Engineering, Kagawa National College of Technology, Japan. His research interests include similarity scale and pattern matching. He has been developing the application software with geometric distance. It is used for detecting bird call, bat call, and whale call in Australia.

Satoru Tsuge

Satoru Tsuge received his B.E., M.E., and Dr. Eng. degrees from the University of Tokushima, Tokushima in 1996, 1998, and 2001, respectively. From 1997 to 1999, he was an intern researcher at ATR Interpreting Telecommunications Research Laboratories, Kyoto. From 2000 to 2009, he was a lecturer at the University of Tokushima. Since 2010, he has been with Daido University, Japan, where he is currently an Associate Professor of Department of Information Systems. His current research interests include speech recognition, speaker recognition, and information retrieval. He is a member of IPSJ and ASJ.

Shingo Kuroiwa

He received the B.E., M.E. and D.E. degrees in electrocommunications from the University of Electro Communications, Tokyo, Japan, in 1986, 1988, and 2000, respectively. From 1988 to 2001 he was a researcher at the KDD R & D Laboratories. From 2001 to 2007, he was an Associate Professor of Institute of Technology and Science at the University of Tokushima, Japan. Since 2007, he has been with Chiba University, Japan, where he is currently a Professor of Graduate School of Advanced Integration Science. His current research interests include speech recognition, speaker recognition, natural language processing, and information retrieval. He is a member of the IEICE, IPSJ, and ASJ.

Minoru Fukumi

Minoru Fukumi received the B.E. and M.E. degrees from the University of Tokushima, in 1984 and 1987, respectively, and the doctor degree from Kyoto University in 1996. Since 1987, he has been with the Department of Information Science and Intelligent Systems, University of Tokushima. In 2005, he became a Professor in the same department. He received the best paper award from the SICE in 1995 and best paper awards from some international conferences. His research interests include neural networks, evolutionary algorithms, image processing and human sensing. He is a member of the IEEE, SICE, IEEJ, IPSJ and IEICE.