World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:2, No:8, 2008

Real-Time Vision-based Korean Finger Spelling Recognition System

Anjin Park, Sungju Yun, Jungwhan Kim, Seungk Min, and Keechul Jung

Abstract**—**Finger spelling is an art of communicating by signs made with fingers, and has been introduced into sign language to serve as a bridge between the sign language and the verbal language. Previous approaches to finger spelling recognition are classified into two categories: glove-based and vision-based approaches. The glove-based approach is simpler and more accurate recognizing work of hand posture than vision-based, yet the interfaces require the user to wear a cumbersome and carry a load of cables that connected the device to a computer. In contrast, the vision-based approaches provide an attractive alternative to the cumbersome interface, and promise more natural and unobtrusive human-computer interaction. The vision-based approaches generally consist of two steps: hand extraction and recognition, and two steps are processed independently. This paper proposes real-time vision-based Korean finger spelling recognition system by integrating hand extraction into recognition. First, we tentatively detect a hand region using CAMShift algorithm. Then fill factor and aspect ratio estimated by width and height estimated by CAMShift are used to choose candidate from database, which can reduce the number of matching in recognition step. To recognize the finger spelling, we use DTW(dynamic time warping) based on modified chain codes, to be robust to scale and orientation variations. In this procedure, since accurate hand regions, without holes and noises, should be extracted to improve the precision, we use graph cuts algorithm that globally minimize the energy function elegantly expressed by Markov random fields (MRFs). In the experiments, the computational times are less than 130ms, and the times are not related to the number of templates of finger spellings in database, as candidate templates are selected in extraction step.

Keywords**—** CAMShift, DTW, Graph Cuts, MRF.

I. INTRODUCTION

COMMUNICATION is composed of different kinds of methods, such as words, voice of tone, and non-verbal forms. Among these methods, non-verbal forms are more effective in delivering a message. According to the researches, in a conversation, verbal expression forms only 35% of overall communication, with the rest consisting of non-verbal forms of communication, such as facial expression, hand and body gesture in lieu of speech [1]. Gesture is a form of non-verbal communication made from a part of body motion and it commonly derives from face and hand, used instead of or in combination with verbal communication.

In our social community, gesture and sign language play an important role in communication between verbal and non-verbal people. A sign language is a language without sound. It is used to convey the meaning of a speaker's thoughts by a combination of hand shape, orientation and movement of the hand, arms or body and facial expression. Sing language is the main communication resource for members of the deaf and speech impaired community. Thus, sign language and gesture recognition is important in assisting human communication especially for the deaf or speech impaired community to deliver their messages by using sign language.

Finger spelling or known as dactylology is an art of communicating by signs made with fingers. Finger spelling has been introduced into sign language in order to serve as a bridge between the sign language and the verbal language. There are many finger letters in use and adopted as a distinct part of sign language around the world that only uses hand to represent the letters of writing an numeral system.

Many researchers aggressively invent and study effective methods or tools to recognize the sign language and finger spelling, and are generally classified into two categories: glove-based and vision-based approaches [2]. The glove-based approaches to recognize finger spelling are simpler recognizing work of hand posture than vision-based, and it also provides the hands-on and minds-on learning experience to the beginner [2]. However, the glove-based approaches require the user to wear a cumbersome device, and carry a load of cables that connected the device to a computer. Vision-based approaches provide an attractive alternative to the cumbersome interface devices for human-computer interaction, and vision-based recognition of hand posture in particular promises more natural and unobtrusive human-computer interaction [2].

Due to this reason, we focus on the vision-based approach in this paper. Huet and Hancock [3] proposed a method of hand shape recognition using geometric histograms, and Mokhtarian et al. [4] proposed to track hand regions using curvature approaches. Suk and Flusser [5] proposed a method to describe the hand region using moment invariants, and Iivarinen and Visa[6] proposed to represent hand regions using chain code histogram. However, these solutions required solving a time consuming, optimization problem. To tackle this problem, Starner and Pentland [7] and New [8] worked for small gesture sets.

A. Park is with the Department of Digital Media, Soongsil University, Seoul, 156-743, Korea (e-mail: anjin@ssu.ac.kr).

S. Yun is with the Department of Digital Media, Soongsil University, Seoul, 156-743, Korea (e-mail: yuyu04@ssu.ac.kr).

J. Kim is with the Department of Digital Media, Soongsil University, Seoul, 156-743, Korea (e-mail: kjw598@ssu.ac.kr).

S. Min is with the Department of Digital Media, Soongsil University, Seoul, 156-743, Korea (e-mail: dfmin84@ssu.ac.kr).

K. Jung is with the Department of Digital Media, Soongsil University, Seoul, 156-743, Korea (corresponding author to provide phone: 82-2-828-7260; fax: 82-2-822-3622; e-mail: kcjung@ssu.ac.kr).

Above-mentioned vision-based approaches generally consist of two steps: hand extraction and recognition, and two steps are processed independently. This paper proposes real-time vision-based Korean finger spelling recognition system by integrating hand extraction into recognition. First, we tentatively detect a hand region using CAMShift algorithm, which can also estimate width, height, and orientation of the detected region. Then fill factor and aspect ratio estimated based on width, height and the tentatively detected regions are used to choose candidate from database, which can reduce the number of templates that will be matched in recognition step. To recognize the finger spelling, we use using dynamic time warping (DTW) algorithm based on modified chain code, and the feature used in this paper is invariant to rotation using modified chain codes and is invariant to scale using DTW algorithm. In this procedure, since hand regions should be accurately extracted without holes and noises, we use graph cuts algorithm that globally minimizes the energy function elegantly expressed by Markov random fields (MRFs). In the experiments, the computational times are less than 130ms, and the times are not related to the number of template finger spellings in database, as candidate templates are selected in extraction step.

The remainder of this paper is organized as follows. Section 2 describes how to tentatively detect a hand region (called hand detection) based on CAMShift algorithm, and section 3 describes how to definitively extract the hand region using a graph cuts method. A hand extraction step and some experimental results are presented in section 4, and the final conclusions are given section 5.

II. Hand Detection

A. Hand Detection

A skin-color model is used to detect hand regions, yet this model usually encounters two main problems: what color space to choose, and how exactly to model the skin color distribution. Thus, normalized RGB (Eq. 1) is used as the color space, which is easily obtained from the RGB values, reduces the space dimensionality, and is invariant to ambient light [9].

$$r = \frac{R}{R+G+B}, g = \frac{G}{R+G+B}, \text{and } b = \frac{B}{R+G+B}. \quad (1)$$

Here, only r and g of the normalized RGB are considered. The Bayesian classifier used to model the skin color distribution is expressed as follows:

$$P(skin|\mathbf{c}) = \frac{P(\mathbf{c}|skin)P(skin)}{P(\mathbf{c})},$$

where $P(\mathbf{c})$ is the probability of observing color $\mathbf{c}$, which is not considered, as it always has same value according to the nature of the Bayesian classifier. Based on the assumption that the probability that a pixel is skin and the probability that a pixel is not skin are same, the skin-color distribution is modeled using only $P(\mathbf{c}|skin)$, and a Gaussian mixture model is used as follows:

$$P(\mathbf{c}|skin) = \frac{1}{2\pi|\mathbf{\Sigma}|^{1/2}} e^{-\frac{1}{2}(\mathbf{c}-\boldsymbol{\mu})^{\mathrm{T}}\mathbf{\Sigma}^{-1}(\mathbf{c}-\boldsymbol{\mu})}$$

where $\mathbf{c}$ is the input vector, and $\boldsymbol{\mu}$ and $\mathbf{\Sigma}$ are the mean vector and covariance matrix as parameters of the skin-color distribution, respectively.

B. CAMShift Algorithm

Hand Detection based on the skin-color model has an important drawback that a significant computational time is needed due to the exhaustive scanning of the entire input image. Thus, the following CAMShift algorithm [10] is used for hand detection to avoid a full scanning of the input image (Fig. 1).

```
1. Set up initial location ($mean_x^0, mean_y^0$) and size ($height^0, width^0$) of initial search window $W$.
2. Generate hand probability image (HPI) within $W$ using skin-color model.
3. Derive new location ($mean_x^t, mean_y^t$) and size ($height^t, width^t$) using zeroth, first, and second-order moments of HPI within $W$.
4. Modify $W$ proportional to values derived in step 3.
5. Increment iteration number $t$.
6. Repeat step 2 until mean location moves less than preset threshold ($\varepsilon_x, \varepsilon_y$).
```

Fig. 1 CAMShift algorithm for hand detection

In this initial stage of the CAMShift algorithm, the initial location ($mean_x^0, mean_y^0$) and size ($height^0, width^0$) of initial search window are determined. During consecutive iterations, the location, size, and orientation of the hand region are estimated using 2D moments, as follows:

$$M_{pq} = \sum_x \sum_y x^p y^q HPI(x,y),$$

where $HPI(x,y)$ is the hand probability of x and y coordinates. Then the location in the search window is

$$mean_x^t = \frac{M_{10}}{M_{00}}, mean_y^t = \frac{M_{01}}{M_{00}}.$$

The size and orientation can be computed as follows:

$$height^t = \sqrt{2(a+c) + 2\sqrt{b^2 + (a-c)^2}} \text{ and}$$

$$width^t = \sqrt{2(a+c) - 2\sqrt{b^2 + (a-c)^2}},$$

where $a = \frac{M_{20}}{M_{00}} - \left(\frac{M_{10}}{M_{00}}\right)^2$, $b = 2\left(\frac{M_{11}}{M_{00}} - \frac{M_{10}M_{01}}{M_{00}^2}\right)$, and $c = \frac{M_{20}}{M_{00}} - \left(\frac{M_{01}}{M_{00}}\right)^2$. The parameters of the search window are then changed depending on the estimated values. If the mean shift is larger than either the threshold values ε_x (along x-axis) or ε_y (along y-axis), the iteration continues.

Fig. 2 shows the hand detection results when using the skin-color model with CAMShift, where Figs. 2(a,b) are the input images and Figs. 2(c,d) are detected result images that show white color in the hand regions and black color in the non-hand regions. As shown in Fig. 2, when the skin color model is used to extract hand regions in real environments, many noises and holes are occurred, thus the noises should be removed and the holes are filled up to extract accurate a hand region. Therefore, the following section describes how to extract more accurate hand regions using a graph cuts method.

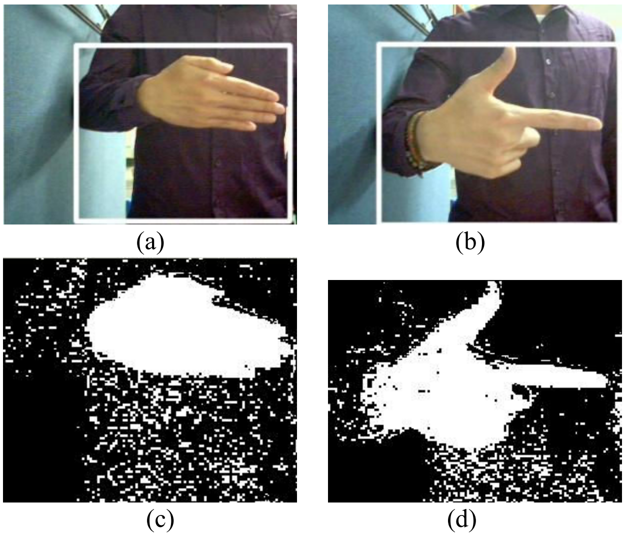


Fig. 2. Results generated by skin-color model: (a,b) input images with rectangles estimated by CAMShift, and (c,d) hand regions detected within rectangles.

III. Hand Extraction

A. Brief Introduction for Segmentation

This paper considers the hand extraction problem as an MRF for labeling problem. A labeling problem is specified in terms of a set of site S and a set of label L. Consider a random field consisting of a set of discrete random variable $\mathbf{F}=\{F_1,F_2,\ldots,F_n\}$ defined on the set S, such that each variable F_s takes one of labels f_s in L. For discrete label set L, the probability that random variable F_s takes the value f_s is denoted $P(f_s)$, and the joint probability is denoted $P(\mathbf{f})$, where $\mathbf{f}=\{f_1,f_2,\ldots,f_n\}$.Here, $\mathbf{f}$ is a configuration of $\mathbf{F}$, corresponding to a realization of the field.

If each configuration $\mathbf{f}$ is assigned a probability $P(\mathbf{f})$, then the random field defined above is said to be an MRF[11] with respect to a neighborhood $N = \{N_s | s \in S\}$, where Ns is the set of sites neighboring s, if and only if the following two conditions are satisfied: $P(\mathbf{f}) > 0,\ \mathbf{f} \in \mathbf{F}$ (called positivity), where $\mathbf{F}$ is a set of all possible configuration, and $P\left(f_s | f_{S-\{s\}}\right) = P\left(f_s | f_{N_s}\right)$ (called Markovian), where $S - \{s\}$ is the set difference, $f_{S-\{s\}}$ denotes the set of labels at the sites in $S - \{s\}$, and $f_{N_s} = \{f_{s'} | s' \in N_s\}$ stands for the set of labels at the sites neighboring s.

Since the $\mathbf{F}$ is generally not accessible, its configuration $\mathbf{f}$ could be estimated only through an observation *obs*. The conditional probabilities $P(\mathbf{f}|obs)$ is the link between the configuration and the observation. A classical method to estimate the configuration f is to use MAP (Maximum A Posterior) estimation, which aims at maximizing the posterior probability $P(\mathbf{f}|obs)$. This is related to the Bayes rule as follows:

$$P(\mathbf{f}|obs) = \frac{P(obs|\mathbf{f})P(\mathbf{f})}{P(obs)}.$$

Since the problem consists in maximizing the previous equation with respect to f, then $P(obs)$ does not act on it. Then the MAP problem is equivalent to

$$\mathrm{argmax}_{\mathbf{f}\in\mathbf{F}} P(obs|\mathbf{f})P(\mathbf{f}). \qquad (2)$$

In Eq. (2), $P(\mathbf{f})$ is Gibbs distribution, and when pairwise cliques are considered, the $P(\mathbf{f})$ is written as follows:

$$P(\mathrm{f}) = \mathrm{Z}^{-1}\exp\left(-\sum_{\{s,s'\}\in N} V_{s,s'}(f_s, f_{s'})\right).$$

In order to obtain a convenient expression of $P(obs|\mathbf{f})$, we assume that $P(obs|\mathbf{f}) = \prod_{s\in S} P(obs_s|f_s)$, and then this term that links observation and configuration is defined as follows:

$$P(obs|\mathbf{f}) \propto \exp\left(-\sum_{s\in S} D_s(f_s)\right).$$

Therefore, the complete MAR-MRF problem can be rewritten as

$$\mathrm{argmax}_{\mathbf{f}\in\mathbf{F}}\exp\left(-\sum_{s\in S} D_s(f_s) - \sum_{\{s,s'\}\in N} V_{s,s'}(f_s, f_{s'})\right),$$

which is equivalent, in an energy function, to

$$\mathrm{argmin}_{\mathbf{f}\in\mathbf{F}}\exp\left(\sum_{s\in S} D_s(f_s) + \sum_{\{s,s'\}\in N} V_{s,s'}(f_s, f_{s'})\right). \qquad (3)$$

For more information for MAP-MRF, please refer to the paper [11].

B. Energy Function for Hand Extraction

In the context of hand extraction, S corresponds to the set of all image pixels, N is a neighborhood defined on the set, L represents the set of different regions, and the random variable $\mathbf{f}$ denotes the labeling assigned to the pixels in the image. Since every possible assignment of the random variable $\mathbf{F}$ defines segmentation, the image segmentation problem can thus be solved by finding the least energy configuration of the MRF.

In this paper, the notation $D_s(f_s)$ in Eq. 3 is called a data term, which indicates the label-preference of each pixel s and reflects how each pixel fits into the prior information, i.e. observation, given for each label in feature space. In other words, the data term has high costs if the pixel s is fit to a label f_s. The notation $V_{s,s'}(f_s, f_{s'})$ is called a smoothness term that encourages spatial coherence, i.e. piecewise smoothness, by penalizing discontinuities between neighboring pixels s and s' in the image domain. In other words, the smoothness term has high costs if two neighboring pixels are similar. We replace $V_{s,s'}(f_s, f_{s'})$ by $V_{s,s'} \cdot \delta(f_s, f_{s'})$ to make the smoothness term general Potts model, where $\delta(f_s, f_{s'})$ denotes the delta function defined by 1 if $f_s \neq f_{s'}$ and 0 otherwise, and thus this term encourages when two pixels have different labels.

The costs of the data term are defined as follows:

$$\begin{cases} D_s(f_s = hand) = P(\mathbf{c}|skin), \\ D_s(f_s = non - hand) = const. \end{cases}$$

where $const$ is threshold value to discriminate hand pixels, and $P(\mathbf{c}|skin)$ denotes likelihood that the pixel is hand ones or not, which described in section II.

The costs of the smoothness term are assigned for discontinuity-preserving between neighboring pixels, and we use general Potts model. Therefore, $V_{s,s'}$ is defined as follows:

$$V_{s,s'} = dis(s, s')^{-1}\exp(-\beta \cdot \|s - s'\|^2), \qquad (4)$$

where $\|s - s'\|^2$ denotes dissimilarities between two pixels s

and s' and $dis(\cdot)$ is the Euclidean distance between neighboring pixels in image domain. When the constant $\beta = 0$, the smoothness term is simply the well-known Ising prior, encouraging smoothness everywhere. However, it has been shown that it is more effective to set $\beta > 0$, as this relaxes the tendency to smoothness in regions of high contrast. The constant β is chosen to be

$$\beta = (\langle \|s - s'\|^2 \rangle)^{-1},$$

where $\langle\cdot\rangle$ denotes expectation over an image sample. This choice of β that ensures that the exponential term in Eq. 4 switches appropriately between high and low constants [12].

To minimize the energy function, we use a graph cut method, as this showed better performance among well-known energy function minimization algorithms [13] that have been proven, such as simulated annealing, iterated conditional modes, and loopy belief propagation. The procedure for energy minimization using the graph cuts method comprises of building a graph in which each cut defines one of all configurations, and the cost of a cut is equal to the energy of its corresponding configuration [14]. For the graph cuts method, a graph $G = \langle \nu, \varepsilon \rangle$ is first constructed with vertices corresponding to the pixels. Two distinguish vertices, *source* (*Src*) and *sink* (*Sin*) called terminals, to represent two labels, plus each vertex has two additional edges {*s*,*Src*} and {*s*,*Sin*}. Therefore, the set of vertices ν and edges ε are as follows:

$$\nu = S \cup \{Src, Sin\} \text{ and } \varepsilon = N \bigcup_{s \in S} \{\{s, Src\}, \{s, Sin\}\}.$$

In the set of edges, *N* is called *n-links* (neighboring links) and {*s*,*Src*} and {*s*,*Sin*} are called *t-links* (terminal links).

The weights of the graph are set for both the t-links and n-links, where t-links connecting each terminal and each vertex correspond to the data term that indicates the label-preferences of each pixel and n-links connecting between neighboring vertices correspond to the smoothness term that indicates continuities between neighboring pixels.

Note that objects segmentation can be solved by finding the least energy configuration of the MRF among every possible assignment of the random variables **F**, and thus minimizing the energy function defined in Eq. 3 is considered as finding the cut with the minimum cost among all the cuts, because the costs of two terms are assigned in weights of the graph. Therefore, after the graph *G* is completely defined, specific labels are then assigned to two disjointed sets connected by *Src* and *Sin* by means of finding the minimum cost cut in the graph [14]. The graph cuts method finds the cut with the minimum cost among all the cuts, and the minimum cost cut problem can be solved by finding the maximum cost cut problem from *Src* and *Sin* based on the theorem of Ford and Fulkerson [15]. Consequently, since the maximum flow in a graph can assist with energy minimization for the labeling problem, the proposed method uses the maximum flow for global minimization of the energy function. Fig. 3 shows result of final hand extraction using the graph cut methods using Fig. 2, and the final width and height are used to choose the candidate from the whole database.

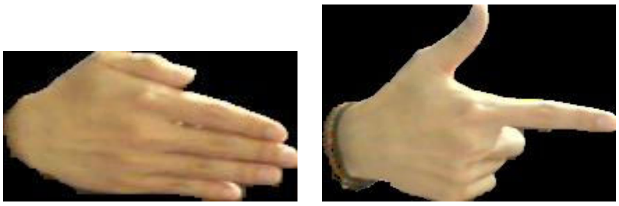


Fig. 3. Result images of final hand extraction using graph cuts method.

IV. Experimental Results

A. Hand Recognition

Hand recognition step retrieves the image stored in database corresponding to the image of an extracted hand region. In the proposed method, the recognition is performed in two steps. First, candidates are first chosen from the whole features stored in database, and then final recognition is performed within the candidates. We use fill factor (left term in Eq. 5) and aspect ratio (right term in Eq. 5) are used to choose the candidates.

$$\text{tempMatching} = \xi \times \frac{\#\text{ of detected hand pixels}}{(\text{width} \times \text{height})\text{ estimated by CAMShift}} + (1 - \xi) \times \frac{\text{height}}{\text{width} \times \text{height}}. \quad (5)$$

where the symbol ξ specifies the relative importance of two terms, and we used 0.3 as a value of ξ.

To represent the shape of the hand region extracted at the previous step, we use a chain code of the boundary of the region. The chain code represents regions using the direction of the boundary at each edge, and the directions are quantized into one of eight.

The chain code can not only preserve lossless boundaries but also permit compact storage, but can have different descriptions according to rotation and a scale of objects to be recognized. To make rotation-invariant descriptions, we determine a starting point and use differential chain code based on the determined starting point. The starting point is chosen by using a closest boundary pixel from a centroid of the region estimated by 2D moment as follows:

$$StaringPoint = \underset{x_{bi} \in X_b, y_{bi} \in Y_b}{\operatorname{argmax}} \sqrt{(x_{bi} - mean_x)^2 + \left(y_{bi} - mean_y\right)^2}$$

where X_b and Y_b are sets of boundary pixels, x_{bi} and y_{bi} are i^{th} boundary pixels, and *x* and *y* are coordinates.

In real environments, almost all input images have different length of boundaries of hand regions. Therefore, we need algorithm compensating the different length between two features to measure similarity, and use DTW algorithm based on dynamic programming. If input features and reference features are $A=\{a_1,a_2,\ldots,a_I\}$ and $B=\{b_1,b_2,\ldots,b_J\}$, respectively, similarity between two features is objectin by using Eq. 6, and different similarity by means of the length of input features is solved by using a normalization (Eq. 7).

$$\gamma(i,j) = \min\{\gamma(i,j-1) + d(a_i, b_j), \gamma(i-1,j-1) + 2 \times d(a_i, b_j), \gamma(i-1,j) + d(a_i, b_j)\} \quad (6)$$

$$DTW(A,B) = \frac{\gamma(I,J)}{I+J} \quad (7)$$

where $d(a_i,b_j)$ is distance between two elements, and is calculated by Euclidean distance.

B. Database

The Korean language is a language that can inscribe the most pronunciation existing in the world. The basic components of the Korean language are composed of 14 consonants and 10 vowels. There are more than 24 different postures for Korean finger spelling. In the proposed system, we only use the most basic 14 Korean letter postures from Fig. 4.

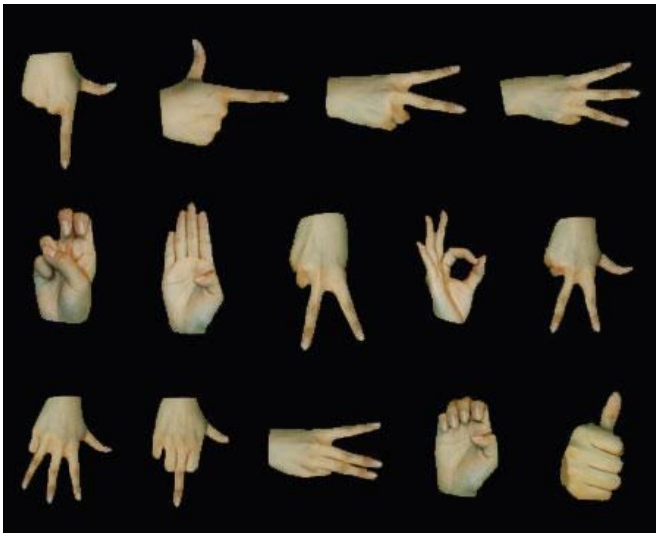


Fig. 4. The most basic 14 Korean letter postures.

C. Analysis

All experiments were carried out on a 2.66 GHz Pentium 4 CPU, and the image size used in the experiments is 240 × 320.

Fig. 5 shows the accuracy of the recognition rate for basic 14 letters, which is tested on beginners who are learning the Korean finger spelling system for the first time, and the average rates is 82.5%. In the database, there are some cases that have the similarity and complicate hand spelling gesture. For example, the Korean letters of ‘ㄹ’ and ‘ㄷ’ had similar features, which is extracted from boundaries using the chain code. This causes the recognition accuracy of ‘ㄹ’ is lower than other letters.

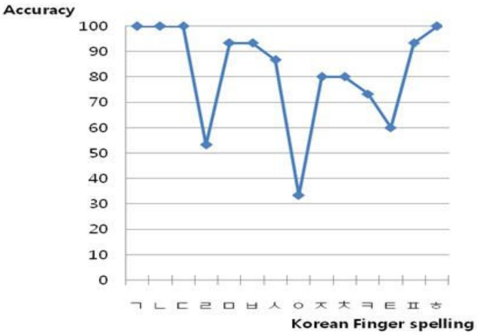


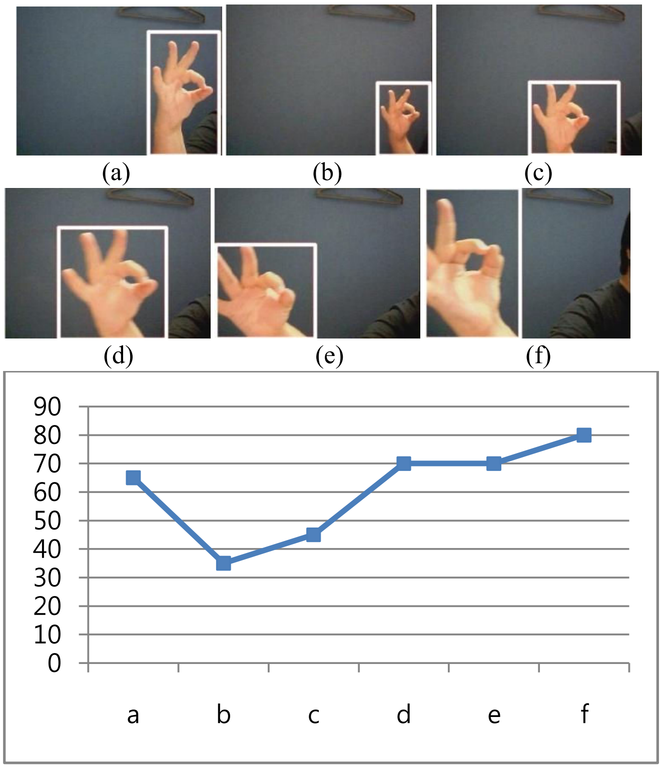
Fig. 5. Accuracy of proposed method.

Table 1 shows the computational times of the proposed method. In our database, each template has three features to be robust in rotation-variation. Therefore, if all features, i.e. 42 features, are compared with three features generated by and input image, the matching step spent about 300ms. However, as shown in Table 1, the proposed method spent about 90ms, as we compared the input features with a few features, i.e. 9 features in present experiments, filled out in extraction step.

Table 1. Computational time of proposed method

	Computational time (msec.)
CAMShift (Tentative extraction)	25
Graph Cuts (Definitive extraction)	40
DTW (Matching)	60
Total	125

The performance of the CAMShift and graph cuts method, hand extraction step, was also evaluated relative to the size of the extracted hand region. In Fig. 6, when the extract hand was far from the camera, the performance of hand extraction step was fast, 35~55ms. However, when the hand was close to the camera, the hand extraction step had a long computational time, 75~100ms. Therefore, average computational time of extraction step was 65ms. Fig. 7 shows application of finger tip recognition developed using the proposed method.

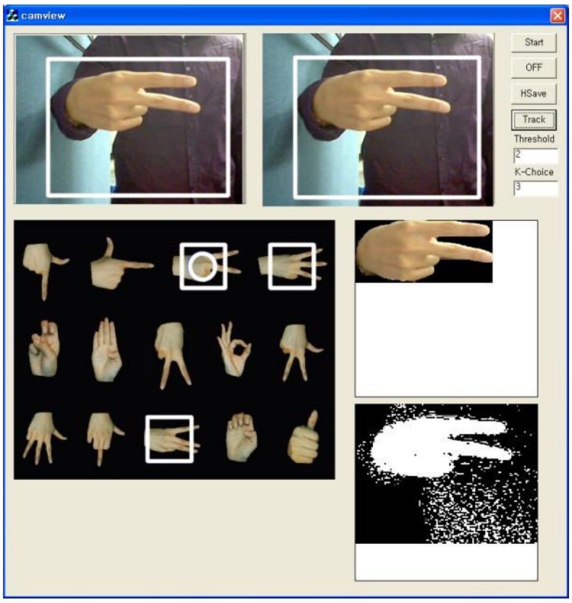


(a) (b) (c)

(d) (e) (f)

(g)

Fig. 6. Performance of hand extraction for image sequence: (a-f) input images, and (g) computational times.



Threshold 2

K-Choice 3

Fig. 7. Proposed system.

V. Conclusion

This paper proposed real-time vision-based Korean finger spelling recognition system, and we integrated the hand extraction into recognition. This was achieved by filling out candidate from the whole database at extraction step. Moreover, the feature used in this paper was invariant to rotation using modified chain codes, and was invariant to scale using DTW algorithm. As a result, the proposed methods can reduce the computational times, not related to the number of templates in database, and showed good accuracy in any invariance, i.e. rotation and scale.

Acknowledgment

This research was supported by the MKE(The Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) support program.

References

[1] R. A. Bolt, "The Integrated Multi-Modal Interface," Institute of Electronics, Information & Communication Engineers, Vol. J77-D, No. 11, pp. 2017-2025, 1987.

[2] V.I. Pavlovic, R. Sharma, T.S. Huang, "Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 7, pp. 677-695, 1997.

[3] B. Huet and E.R. Hancock, "Relational Histograms for Shape Indexing," Proceedings of International Conference on Computer Vision, pp. 563-569, 1998.

[4] F. Mokhtarian, S. Abbasi, and J. Kittler, "Efficient and Robust Retrieval by Shape Content Through Curvature Scale Space," Proceedings of International Workshop on Image Databases and MultiMedia Search, pp. 35-42, 1996.

[5] T. Suk and M.D. Flusser, "Combined Blur and Affine Moment Invariants and Their Use in Pattern Recognition," Pattern Recognition, Vol. 36, pp. 2895-2907, 2003.

[6] T. Starner and A. Pentland, "Visual Recognition of American Sign Language using hidden Markov Models," Proceedings of International Workshop on Automatic Face and Gesture Recognition, pp. 189-194, 1995.

[7] J. Iivarinen and A. Visa, "Shape Recognition of Irregular Objects," Proceedings of International Conference on Intelligent Robots and Computer Vision XV, pp. 25-32, 1996.

[8] Joshua R. New, "A Method for Hand Gesture Recognition," Proceedings of ACM Chapter Fall Conference, 2002.

[9] S.L. Phung, A. Bouzerdoun, and D. Chai, "Skin Segmentation using Color Pixel Classification: Analysis and Comparision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 1, pp. 148-154, 2005.

[10] G.R. Bradski and V. Pisarevsky, "Intel's Computer Vision Library: Application in Calibration, Stereo, Segmentation, Tracking, Gesture, Face and Object Recognition," in Proceedings of IEEE Conference of Computer Vision and Pattern Recognition, vol. 2, pp. 796-797, 2000.

[11] S.Z. Li, *Markov Random Field Modeling in Computer Vision*, Springer, 2001.

[12] Y. Boykov and G. Funka-Lea, "Graph Cuts and Efficient N-D Image Segmentation," *International Journal of Computer Vision*, vol. 70, no. 2, pp. 109-131, 2000.

[13] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother, "A Comparative Study of Energy Minimization Methods for Markov Random Fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, issue 6, pp. 1068-1080, 2008.

[14] Y. Boykov, O. Veksler, and R. Zabih, "Fast Approximate Energy Minimization via Graph Cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1222-1239, 2001.

[15] L. Ford and D. Fulkerson, *Flows in Networks*, Princeton University Press, 1962.