# Dynamic gesture recognition using a model-based temporal self-similarity and its application to *taebo* gesture recognition

**Kyoung-Mi Lee[1] and Hey-Min Won[2]**
[1]Department of Computer Science, Duksung Women's University
Seoul, 132-714, Korea
[e-mail: kmlee@duksung.ac.kr]
[2]Intelligent Multimedia Lab., Duksung Women's University
Seoul, 132-714, Korea
[e-mail: haemini86@naver.com]
*Corresponding author: Kyoung-Mi Lee

*Received May 20, 2013; revised August 7, 2013; revised September 17, 2013; accepted October 23, 2013; published November 29, 2013*

## Abstract

There has been a lot of attention paid recently to analyze dynamic human gestures that vary over time. Most attention to dynamic gestures concerns with spatio-temporal features, as compared to analyzing each frame of gestures separately. For accurate dynamic gesture recognition, motion feature extraction algorithms need to find representative features that uniquely identify time-varying gestures. This paper proposes a new feature-extraction algorithm using temporal self-similarity based on a hierarchical human model. Because a conventional temporal self-similarity method computes a whole movement among the continuous frames, the conventional temporal self-similarity method cannot recognize different gestures with the same amount of movement. The proposed model-based temporal self-similarity method groups body parts of a hierarchical model into several sets and calculates movements for each set. While recognition results can depend on how the sets are made, the best way to find optimal sets is to separate frequently used body parts from less-used body parts. Then, we apply a multiclass support vector machine whose optimization algorithm is based on structural support vector machines. In this paper, the effectiveness of the proposed feature extraction algorithm is demonstrated in an application for *taebo* gesture recognition. We show that the model-based temporal self-similarity method can overcome the shortcomings of the conventional temporal self-similarity method and the recognition results of the model-based method are superior to that of the conventional method.

***Keywords:*** Gesture recognition, feature extraction, dynamic gesture, gesture spotting

This research was supported by the Duksung Women's University Research Grants 2013.

**http://dx.doi.org/10.3837/tiis.2013.11.016**

## 1. Introduction

Human gesture recognition is receiving increased attention from computer vision researchers. This attention is motivated by a wide spectrum of application domains, such as video surveillance, machine control, interactive physical game, and sport video analysis. All these application domains have their own demands, but in general, the gesture recognition methods must be able to detect and recognize various human gestures in real time. Also, as people look different and move differently, the designed methods must be able to handle both variations in performing gestures and various kinds of environments.

Many approaches for human gesture recognition have been proposed in the literature [1,2]. Recently there have been a lot of studies on analyzing dynamic human gestures that vary over time. Most approaches to dynamic gesture recognition are addressed with a variety of machine learning techniques such as Hidden Markov Models (HMM) [6] and Support Vector Machines (SVM) [7]. The HMM-based approach can process data in the time domain, but it requires multi-dimensional gesture data to be converted into discrete one-dimensional data. On the other hand, the SVM-based approach can deal with multi-dimensional data and is easy to optimize.

For dynamic gesture recognition, time-varying gestures have been represented as spatio-temporal features, instead of each frame being analyzed as individual, separate gestures. Bobick and Davis use spatio-temporal templates where a vector value at each point is a function of the motion properties at the corresponding spatial location in an image sequence [3]. Li and Greenspan build a multi-scale gesture model as a set of 3D spatio-temporal surfaces of a time-varying contour [4]. Gorelick *et al*. analyze 2D shapes as silhouettes of a moving torso and protruding limbs and generalize to deal with volumetric space-time action shapes [5]. Junejo *et al*. propose a temporal self-similarity(TSS) method, as a gesture descriptor that captures the structure of temporal similarities and dissimilarities within gesture sequences for view-independent video analysis [7]. However, TSS does not take into account relations among parts of the human body. The TSS method cannot recognize similar gestures performed with different body parts, such as a Jab with a hand and a Side Kick with a leg.

In this paper, we use TSS for dynamic gesture features and modify it by grouping body parts into several sets. Dividing a whole body into several sets allows a recognition system to know whose sets move because it must no longer consider a body as one set. For example, while the modified TSS can recognize whose sets move, TSS can know only how much the body moves. Also, using a human model with relationships among body parts can be a solution in cases where some parts are missing from feature extraction. In Section 2, we describe TSS and the proposed model-based TSS. The proposed model-based TSS ties adjacent parts that are likely to move together in the human model. Then, the proposed TSS is applied to *taebo* gesture recognition using SVM in Section 3. Section 4 presents experimental results of *taebo* gesture recognition using the model-based TSS compared to conventional TSS.

## 2. Model-based TSS for Dynamic Gesture Features

This section proposes new dynamic gesture features based on a human model $h^{m=1..M}$, where $M$ is the number of body parts. We introduce the conventional TSS [7] and describe the proposed model-based TSS using $h^{m=1..M}$.

### 2.1 Temporal Self-Similarity

The conventional TSS feature is used to discover particular time-dependent gesture features and data in a matrix format. Junejo *et al.* extracted gesture features using a self-similarity matrix (SSM) in calculating distance among all features by extracting time-frame and storage results [7]. For a sequence of frames $I = \{I_1, I_2, \cdots, I_N\}$, an SSM is computed in a Euclidean distance matrix form of size $N \times N$,

$$d_{ij} = \begin{bmatrix} 0 & d_{12} & \cdots & d_{1N} \\ d_{21} & 0 & \cdots & d_{2N} \\ \cdots & \cdots & \cdots & \cdots \\ d_{N1} & d_{N2} & \cdots & 0 \end{bmatrix} \quad (1)$$

where $N$ is the number of frames. A diagonal of the matrix, $d_{ii}$, refers to the periodicity of gestures ('0') which is a comparison of the frame to itself. With a human or object of $M$ parts $h^{m=1..M}$, a Euclidean distance between the $m$-th parts at any instances $i$ and $j$ can be calculated as the sum of movements:

$$d_{ij} = \frac{1}{M} \sum_{m=1}^{M} \left\| h_i^m - h_j^m \right\|_2 \quad (2)$$

where $h_i^m$ and $h_j^m$ are positions of $m$-th parts at time instances $i$ and $j$. Each structure or pattern of the matrix is dependent on the distance measurement of $d_{ij}$.

(a) right hand forward (b) left hand forward (c) both hands forward

**Fig. 1.** TSS feature pattern examples

**Fig. 1** shows pattern examples by extracting gesture features using TSS after normalizing consecutive gesture data. The patterns are made by gestures of the right and left hands, which moved to the front by 50cm in 10 frames. TSS calculates by estimating the Euclidean distance of all gesture trajectories. In **Fig. 1**, a lighter color means a longer distance and longer movement, while a darker color means a shorter distance and shorter movement. **Figs. 1(a)** and **1(b)** reveal that gesture features would be almost the same if the size of the gesture is equal even though the gestures are created using different body parts. In addition, **Fig. 1(c)** differs from **Figs. 1(a)** and **1(b)**, but also has a similar pattern to **Figs. 1(a)** and **1(b)** if they are scaled with their maximum values. Even though the gestures are created with both hands, however, it is unknown which body part is used to create these gestures. Therefore, the conventional TSS feature has difficulty determining partial gestures because of a lack of information on which body parts are used to create the gestures.

### 2.2 Model-based TSS

In this section, we propose a model-based TSS feature to solve the shortcomings of the conventional TSS feature. The model-based TSS feature is obtained by binding the features of human or object with more than one feature, as follows from **Eq. (2)** :

$$d_{ijp} = \frac{1}{M_p} \sum_{m=1}^{M_p} \left\| h_i^m - h_j^m \right\|_2 \quad (3)$$

where $p$ is the number of sets, and $M_p$ refers to the set of features within the $p$-th set.

The proposed model-based TSS feature has the following properties :

i) $M_1 \cup M_2 \cup ... \cup M_p \subset M$ : a union of feature sets is included in a set of all features.

ii) $M_i \cap M_j \geq \varnothing$ : an intersection of any feature sets may not be empty, meaning a feature can be included in different sets.

iii) If each feature set has the same size of gesture, the model-based TSS has the same pattern.

iv) If there is no movement, the model-based TSS has a homogenous pattern.

v) If a noise occurs, the distance $d_{ijp}$ between the noisy frame and any other frames is equal to the size of the noise. The noised frame is calculated the sum of movements as a long pattern, and the following frames are represented as point patterns.

vi) If the movement of a feature set has a constant velocity, the distance $d_{ijp}$ with following frames is proportional to the distance between $i$ and $j$. The previous patterns of such proportional distances are repeated behind the newly-calculated pattern.

vii) If the movement of a feature set has a cycle, several similar patterns of distance $d_{ijp}$ appear. For example, the following model-base TSS contains periodic flat long patterns.

viii) If the movement of a feature set has constant acceleration, then the distance $d_{ijp}$ becomes greater and the length of the patterns increases.

The conventional TSS uses a square matrix and its diagonal itself (without difference) is the comparison of the frame, represented by 0, which stands for gesture symmetry. Therefore, we arrange all values in line without gesture symmetry in the matrix (not a square matrix) in the Model-based TSS. In **Fig. 2**, human body parts are grouped into six sets and each cell in left-side tables present a corresponding set of body parts. Each row of the Model-based TSS in the middle shows TSS of the corresponding body set. **Figs. 2(a)** and **2(b)** present the model-based TSS features after normalizing consecutive gesture frames that created by moving right and left hands to the front by 50cm in 10 frames. The features get the information on which body part is used because the patterns that represent movements of the right and left hands are presented on different lines. As shown in **Fig. 2(c)**, in addition, the Model-based TSS displays the gestures easily, even though both hands are used at the same time. Any two body parts which are differentiated from each other should be included in different feature sets.

## 3. *Taebo* Gesture Recognition

We apply the proposed model-based TSS feature to *taebo* gestures recognition and implement a gesture recognition method that employs human detection, gesture spotting, and gesture recognition algorithms, which is summarized in **Fig. 3**.

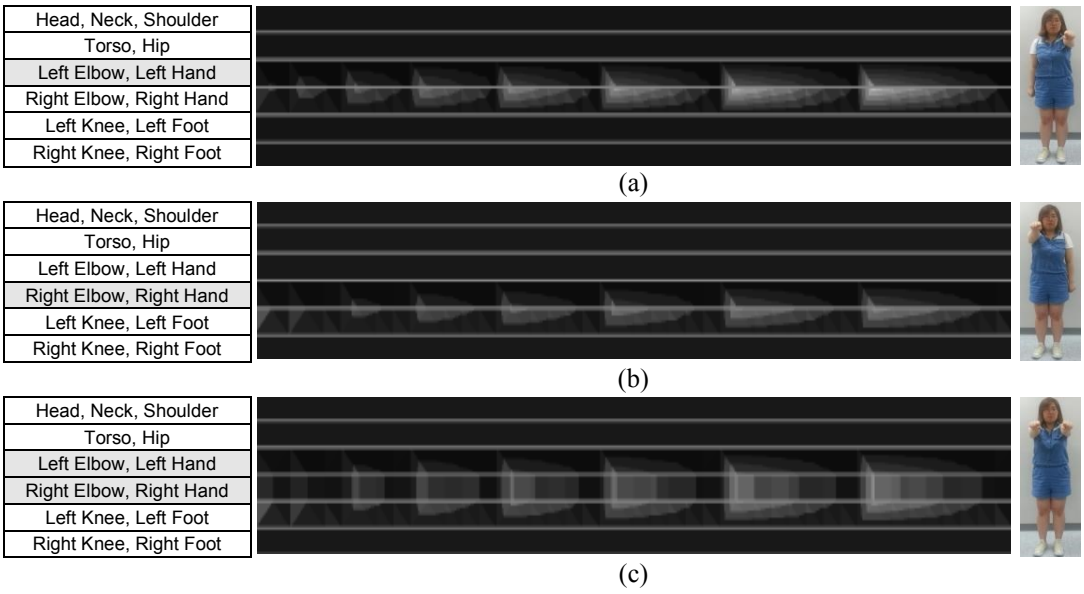



**Fig. 2.** Model-based TSS pattern examples : (a) right hand forward, (b) left hand forward, (c) both hands forward

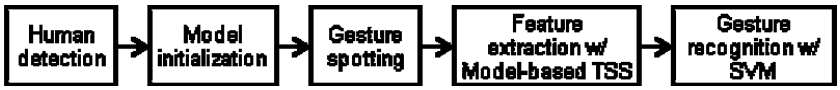



**Fig. 3.** Overview of the proposed gesture recognition method

### 3.1 Human Detection and Model Initialization

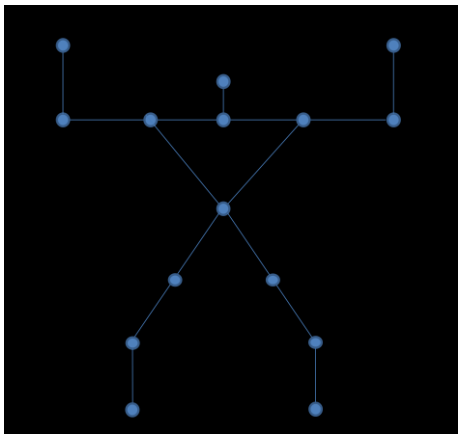


**Fig. 4.** Human KINECT model

To detect a human from a background in a frame, we used a KINECT camera with an infrared sensor at 30 frames per second. The KINECT camera use the human skeleton model $h_i$, **Fig. 4,** which consists of 15 body parts. If a user strikes a pose called "PSI," which looks like lifting a weight by placing his/her feet about as wide as the shoulders, the user is segmented from the background and the calibration is carried out with the human model by obtaining a 3D position of each skeleton part as $x$, $y$, and $z$ dimensional values.

**Fig. 5** shows 9 *taebo* gestures: Front Kick, Side Kick, Knee Kick, Cross Punch, Jab,

Uppercut, Stand, Step 1, and Step 2. Most of these gestures have large movements of one limb through a straight line in one direction. For example, the Front Kick, Side Kick, and Knee Kick all stretch out one leg. Also, the Cross Punch, Jab, and Uppercut all stretch out one arm. As a consequence, we need algorithms to identify and recognize gestures that have similar movements but use different limbs.

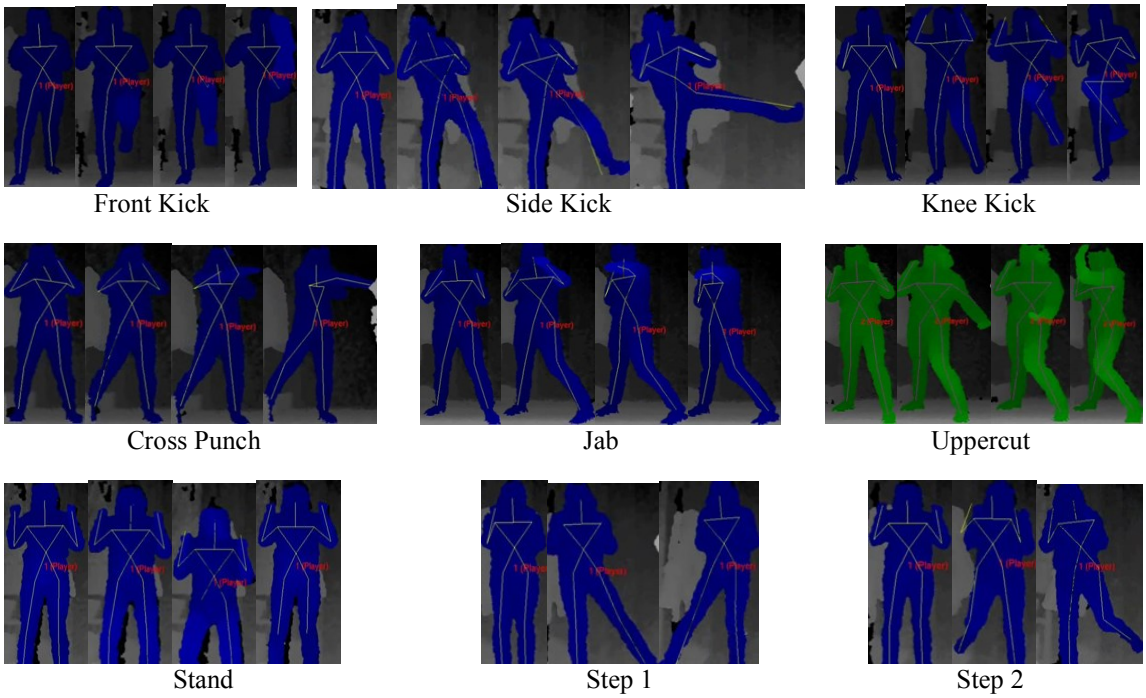



**Fig. 5.** *Taebo* gestures using a human model

### 3.2 Gesture Spotting

For recognition of time-varying gestures, it is necessary to get a meaningful gesture only from input video sequences. This can present a segmentation(spotting) problem that detects a dynamic gesture boundary in finding when a gesture starts and when it ends in a continuous body trajectory [8]. For more efficient gesture recognition, it is an important task to find candidate gesture boundaries. In general, the following three properties are available to find candidates for the cut [9]:

- a frame that has an abnormal velocity,
- a frame that has a static gesture, and
- a frame that has a severe curvature.

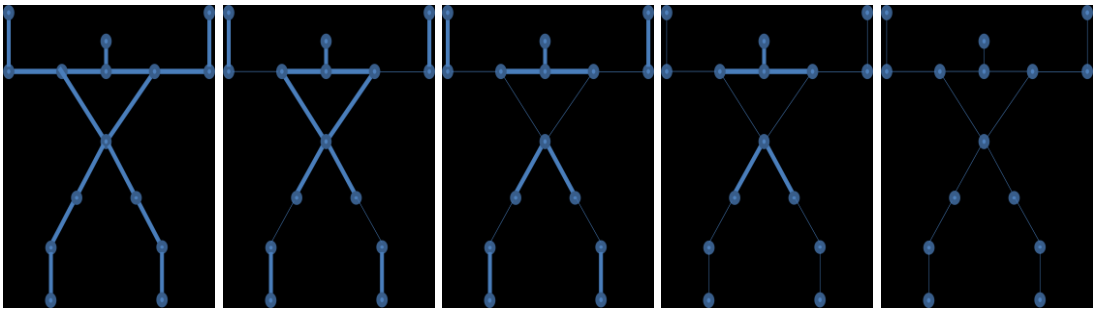
Also, a frame can be selected if it has a common gesture pattern, which is often observed at the start and end of the gesture.

In this paper, we define an individual gesture by stopping for a short time before and after each gesture (i.e., one gesture is performed after another but a pause exists between the two gestures). At the end of a gesture such as punch, stand, or step, the person pauses for a moment. For kick gestures, however, it is difficult for a person to stay balanced at the end of the gesture. Hence, such gestures end by putting down his leg to the ground. Because a gesture increases variation at the start of the gesture and suddenly decreases at the end, we define a candidate frame as one that has a sudden variation in size, velocity, and curvature of movement.

After gesture spotting, we can get a set of meaningful frames to configure the dynamic gesture. However, whenever the same gesture is performed, the number of spotted frames and the number of features will vary. It is not appropriate if a recognition system needs the same number of features. Thus, we normalize the number of frames to make the same number of frames, $n$, per gesture. The algorithm for normalizing the number of frames includes the first frame of spotted frames, and then the following normalized $j$-th frames can be made by interpolation between two adjacent spotted frames.

### 3.3 Gesture Feature Extraction

Once meaningful gestures are segmented, gesture features can be extracted. The extraction of gesture features is a process of extracting common gesture properties. In addition, it is a process of recognizing gestures based on distinctive features, instead of a great number of complicated data. In this paper, we propose a model-based TSS feature that ties body parts of a human model into several sets based on their hierarchical relations. The hierarchical relations, which are configured in tree form with parent-child relationships, have advantages of easy extension and easy updating. In such a hierarchical human-body model, hands and feet can be considered as lowest-level nodes which are farthest from the center of the human body. As they become farther from the center node, the degree of freedom of the human body increases. With *taebo* gestures, there are a lot of movements in the hands and feet, while a face and neck has relatively little movements. Therefore, the face and neck are bound, and we group a set of body parts by focusing on the rest of the parts (especially leaf nodes).



(a) (b) (c) (d) (e)

**Fig. 6.** Human model examples which group 15 body parts into (a) 1, (b) 5, (c) 6, (d) 10, and (e) 15 sets

**Fig. 6(a)** is a model that groups body parts as a single set, which obtained results similar to conventional TSS. **Fig. 6(b)** is a model that groups body parts into five sets. This model groups arms and elbows and knees and feet, which are commonly used in *taebo*, in four sets (arm right and left, leg right and left), and groups central parts(head, neck, torso, shoulders, and hip), which are used relatively little, in a single set. **Fig. 6(c)** is a model that groups the parts into six sets. This is similar to **Fig. 6(b)**, without separating torso and hips from the head, neck, and shoulders. Because the central parts are a large portion of the human body, **Fig. 6(c)** is divided into upper parts and lower parts. The model shown in **Fig. 6(d)** divides four limbs into each part. **Fig. 6(e)** is a model that uses 15 parts as they are, and movement of each part is measured.

**Fig. 7** presents model-based TSS feature patterns of a *taebo* gesture 'Knee Kick' using grouped human models from **Fig. 6**. The gesture 'Knee Kick' has a large movement of one foot with a small amount of movement for hands and elbows. While the pattern of a single set, **Fig. 7(b)**, cannot identify which parts are moved. **In Fig. 7(f),** the pattern of 15 sets separates all movement into each part and does not know relations among them.

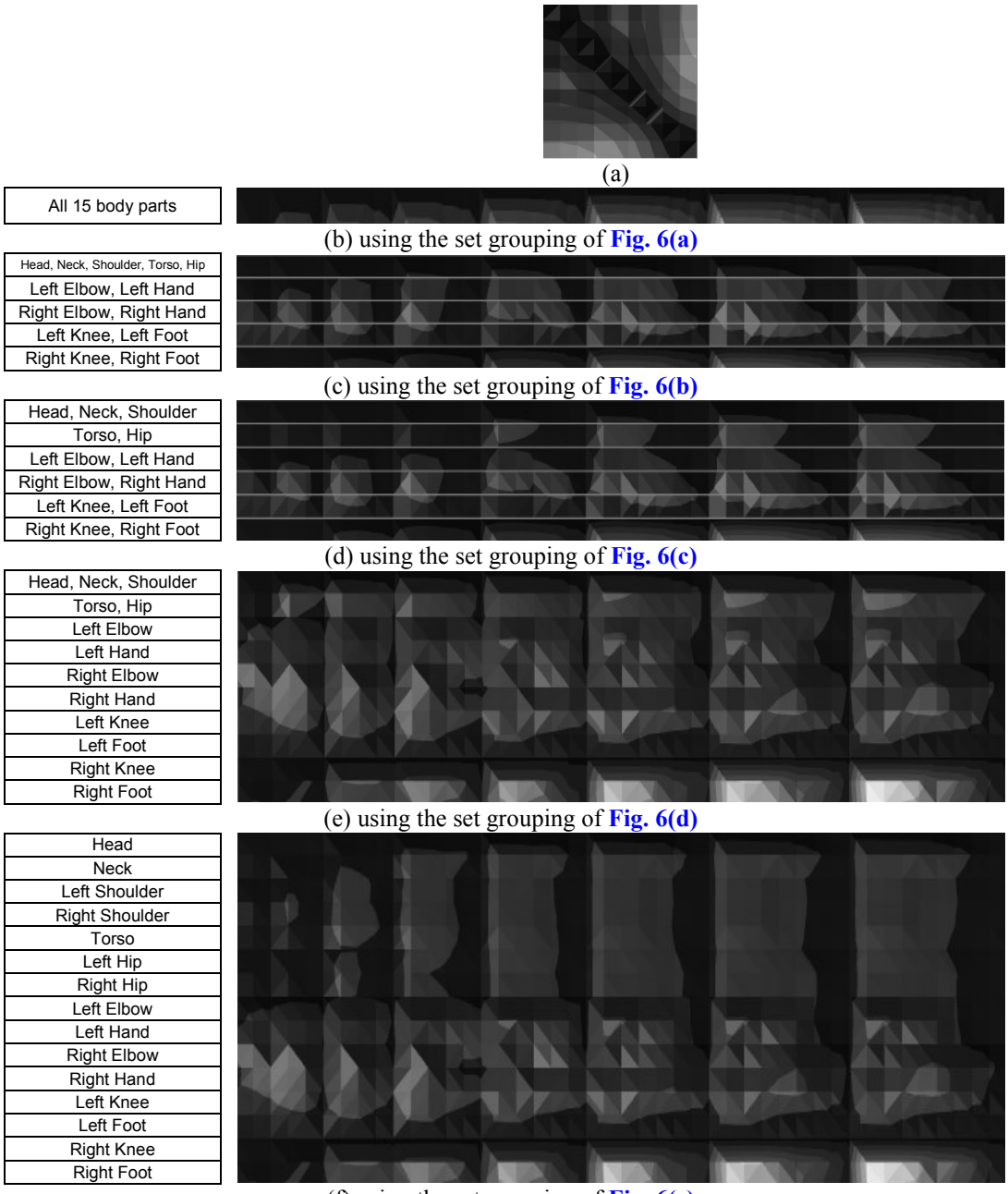



**Fig. 7.** Pattern examples of a *taebo* gesture Knee Kick: (a) TSS and (b-f) Model-based TSS using **Fig. 6** with feature sets

**Fig. 8** shows other pattern examples for 'Side Kick' and 'Cross Punch'. The two gestures move a leg and an arm, but both are similar in terms of TSS features because both have similar amounts of movements. To distinguish these two gestures, a model-based TSS feature separates arms from legs, for example **Fig. 6(c)**. **Fig. 8** presents how the proposed model-based approach differentiates such gestures.

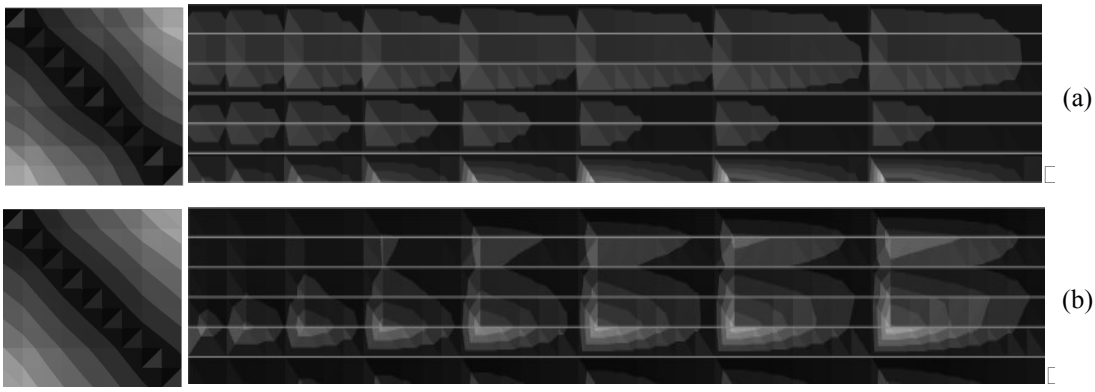


**Fig. 8.** Pattern examples: (a) Side Kick and (b) Cross Punch. Right: TSS. Left: Model-based TSS using **Fig. 6(c)**

### 3.4 Gesture Recognition using SVM

After computing features of time-varying gestures, a gesture recognition algorithm is created. In general, dynamic gestures can be represented as multi-dimensional and time-varying data. For time-varying gestures, the use of the hidden Markov model(HMM) is a common approach [10], but HMM requires a process to convert multi-dimensional gesture data into discrete one-dimensional data. In this paper, time-dependent features are computed, thus gesture recognition can focus on the problem of multi-dimensional data. We used the support vector machine, which is widely used in multi-dimensional classification [11]. The SVM converts data into multi-dimensional spaces and divides them into several classes by rendering them in multi-dimensions in calculating a support vector. The SVM has a non-linear, complicated decision boundary and it is difficult to optimize them. However, because the SVM just requires selecting a kernel, the kernel's parameters, and a soft margin parameter, it is relatively easy to achieve optimization and it has superior generalization functions in classification.

Because one-to-many classification, not one-to-one classification, is needed to recognize several dynamic *taebo* gestures, we adopt multiclass SVM developed by Crammer and Singer [12]. They find the following optimization problem:

$$\min_{w,\xi} \left[ \frac{1}{2} C \sum_{r=1..k} \overline{w}_r^2 + \sum_{i=1}^{t} \xi_i \right] \tag{4}$$

$$\text{Subject to} : \forall i, r \ \overline{w}_{y_i} \bullet \overline{x}_i + \delta_{y_i,r} - \overline{w}_r \bullet \overline{x}_i \geq 1 - \xi_i$$

where $\overline{x}_i$ is an $i$-th example of $t$ training data in $N$ features and is mapped to the multiclass set $\{1,...,k\}$. To recognize the nine gestures in **Fig. 5**, $k$ is 9. $w$ is a matrix of size $k \times N$ and $\overline{w}_r$ is the $r$-th row of $M$. $C > 0$ is a regularization constant that trades off margin size and training error. In this paper, we set the value of C to 900,000. $\xi_i \geq 0$ are slake variables. The multiclass SVM optimizes the basis function with an algorithm that is based on Structural SVMs [13]. Among structural learning algorithms, we use the 1-slake algorithm whose value is 99.75046 on a working set and is 99.78659 for a global. The loss function $\delta_{y_i,r}$ is equal to 1 if $y_i = r$ and 0 otherwise. Then the remaining parameters of the multiclass SVM are set by default values.

## 4. Experimental Results and Analysis

The proposed dynamic gesture feature extraction and recognition algorithm was implemented on a Pentium IV 3.0 GHz CPU with 3 GB of memory running C++, OpenCV, and OpenNI with Microsoft Visual Studio 2010under Microsoft Windows XP. The experimental video frames were taken by a KINECT camera in a room with illumination of 600 Lux on average. The distance between the camera and a person was about1.5m. The gesture database was built by gathering 50 samples per gesture from 10 persons and each person performed each gesture five times. Thirty samples from each gesture were used as training data and the rest for testing.

### 4.1 Experimental Results

**Table 1.** *Taebo* gesture recognition rate(%) of human models in **Fig. 6**

| Features / Gestures | TSS | Model-based TSS | | | | |
|---|---|---|---|---|---|---|
| | | Fig. 6(a) ($p$=1) | Fig. 6(b) ($p$=5) | Fig. 6(c) ($p$=6) | Fig. 6(d) ($p$=10) | Fig. 6(e) ($p$=15) |
| Front Kick | 0 | 0 | 100 | 100 | 100 | 100 |
| Side Kick | 80 | 80 | 90 | 100 | 100 | 100 |
| Knee Kick | 55 | 55 | 80 | 80 | 65 | 65 |
| Cross Punch | 10 | 10 | 80 | 70 | 55 | 55 |
| Uppercut | 35 | 35 | 65 | 70 | 80 | 80 |
| Jab | 15 | 15 | 65 | 85 | 85 | 85 |
| Stand | 15 | 15 | 75 | 75 | 75 | 75 |
| Step 1 | 0 | 0 | 90 | 90 | 100 | 100 |
| Step 2 | 45 | 45 | 90 | 90 | 80 | 80 |
| Average | 28.33 | 28.33 | 81.67 | 84.44 | 82.22 | 82.22 |

**Table 1** presents the recognition rates of nine *taebo* gestures based on how they were made by a human model. Experimental tests are performed after finding the best parameters for gesture recognition depending on the human model. The conventional TSS computes the sum of the distance of all body parts. So, if the sums of distances, **Eq. (2)**, among gestures are pretty much the same, the recognition rate of TSS is low regardless of which body parts are moved. In TSS, 'Front Kick' and 'Step 1' are not differentiated because the sum of the gesture movements of 'Front Kick,' is similar to both 'Knee Kick' and 'Step 2'. In 'Step 1' as well, the sum of the gesture movements is similar to those of most *taebo* gestures.

On the other hands, the proposed model-based TSS features classifies more properly because the model-based features can identify gestures for which different body parts make the same amount of movements. In the model-based TSS, the recognition rate of 'Front Kick' improved by 100% and 'Step 1' by at least 90%, regardless of the human model, except when $p$=1 (which is the same as TSS). The recognition rates of 'Cross Punch', 'Uppercut', 'Jab', 'Stand' and 'Step 2' also are improved drastically by at least 30% compared to TSS.

As a whole, in **Table 1**, the recognition rates of punch gestures vary depending on how hands, elbows, and shoulders are grouped, and those of kick gestures vary depending on how feet, knees, and hips are grouped. In **Fig. 6(b)**, the human model is grouped into five sets in which torso, shoulders and hips are grouped into one set, and thus it is slightly difficult to recognize punch gestures using shoulders from kick gestures using hips, compared to **Fig. 6(c)** which separates shoulders from torso and hips. When the human model is divided into 10 or 15 sets that separate four limbs into hand and elbow or foot and knee, the recognition rates increase for 'Uppercut' and 'Step 1' but decrease for 'Knee Kick' and 'Cross Punch', compared to **Fig.**

6(c).

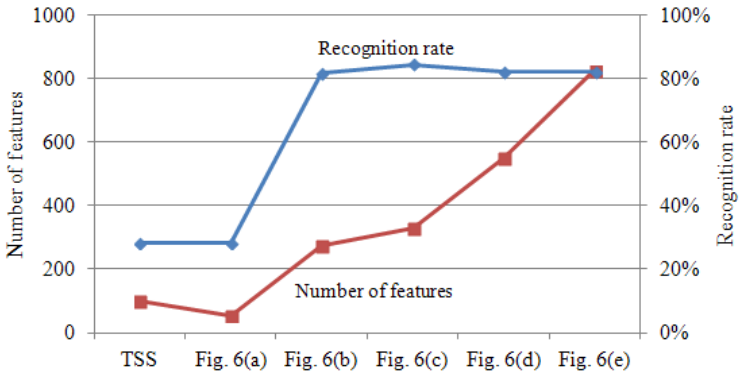



**Fig. 9.** Recognition results and number of features of the human models in **Fig.6(a-e)**

To evaluate the recognition performance of the algorithm, we used two measurements, recognition rates and the number of features. The number of features is calculated as seen in **Fig. 9**. In this paper, we set the number of normalized frames $n$ to 10 and compute $N = n^2 = 100$ as the number of features of the TSS matrix. The number of features of the model-based TSS is computed as $N = p \times \frac{n(n+1)}{2}$ by eliminating an upper triangle of the TSS matrix for each set, and thus a total number of 55 features are calculated in each set. In this paper, a total of 55 features are saved when $p$ is 1, while 275 features are saved when $p$ is 5. When $p$ is 6, 10, and 15, the number of features is 330, 550, and 825, respectively. As shown in **Fig. 9**, the recognition rates differ depending on the number of sets in the model-based TSS, except when $p$=1 which is the same as TSS. Regardless of the human model, all results of the model-based TSS are much higher than that of TSS. Also, the number of features does not help to improve recognition performance. When the human model is divided into more detail, the inclusion of too many features with confused properties degrades the performance of the SVM classifier and thus gestures could not be recognized. As a result, an optimum recognition rate can be obtained if the human model is divided into an optimal number of sets.

**Table 2(a)** shows that the TSS model confuses some Knee Kicks with Uppercuts. They are similar gestures in that both are moving one limb upward, but differ in the limb moved upward. In **Table 2(b)**, the proposed model-based TSS can reduce confusions of similar movements with different parts and thus differentiate Knee Kick from Uppercut. In the case of Jab and Side Kick, they stretch one limb, but use either the hand or foot. TSS has some confusion in classifying them, but the model-based TSS can recognize Jab well.

### 4.2 Experiments with the Same Number of Sets, but Different Combinations

In this section, we conduct experiments on optimal combinations. When the human model is divided into 10 sets, as in **Fig. 6(d)**, different combinations are possible, for example, in **Fig. 10**. **Table 3** presents the recognition results compared with **Fig. 6(d)**. The model in **Fig. 10(a)** decreases the recognition rate to the lowest value. Such a combination especially confuses some Cross Punches with Knee Kicks and some Knee Kicks with Front Kicks. The reason is that grouping frequently-used elbows with relatively less-used shoulders and frequently-used knees with the relatively little-used hips decreases the recognition rate for Cross Punches and Knee Kicks. Any combinations of shoulder–elbow(**Fig. 10(d)**) and knee–foot(**Fig. 10(b.c)**)

achieve relatively lower results than that of **Fig. 6(d)**, but much higher than that of **Fig. 10(a)**. It confirms that the best way to find the optimal combination is separating frequently used body parts from less-used parts.

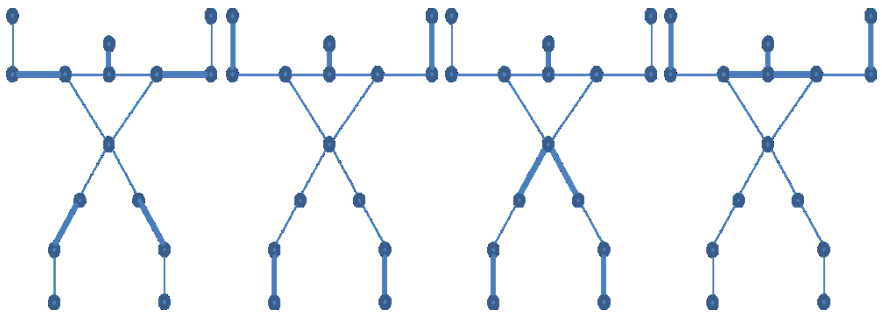
**Table 2.** Confusion matrix on 20 test data values of each gesture

| Predicted / True | Front Kick | Side Kick | Knee Kick | Cross Punch | Uppercut | Jab | Stand | Step 1 | Step 2 |
|---|---|---|---|---|---|---|---|---|---|
| Front Kick | | | 7 | | | | | | 13 |
| Side Kick | | 16 | 2 | | | | | | 2 |
| Knee Kick | | | 11 | | 4 | | | | 5 |
| Cross Punch | | | 2 | 2 | 5 | 5 | 4 | | 2 |
| Uppercut | | 3 | | | 7 | | 3 | | 7 |
| Jab | | 3 | | | 3 | 3 | 8 | | 3 |
| Stand | 3 | 11 | | | 3 | | 3 | | |
| Step 1 | 5 | 2 | 2 | | 4 | | 2 | | 5 |
| Step 2 | 2 | | 4 | | 5 | | | | 9 |

(a) TSS

| Predicted / True | Front Kick | Side Kick | Knee Kick | Cross Punch | Uppercut | Jab | Stand | Step 1 | Step 2 |
|---|---|---|---|---|---|---|---|---|---|
| Front Kick | 20 | | | | | | | | |
| Side Kick | | 20 | | | | | | | |
| Knee Kick | | 2 | 16 | | | | | 2 | |
| Cross Punch | | | 2 | 14 | | 4 | | | |
| Uppercut | | | | | 14 | 6 | | | |
| Jab | | | | 3 | | 17 | | | |
| Stand | | | | | | | 15 | | 5 |
| Step 1 | | 2 | | | | | | 18 | |
| Step 2 | | | | | | | | 2 | 18 |

(b) Model-based TSS in **Fig. 6(c)** ($p$=6)



(a) (b) (c) (d)

**Fig. 10.** Human model examples that group 15 body parts into 10 sets, but with different combinations

**Table 3.** *Taebo* gesture recognition rate(%)of human models in **Fig. 10**

| Feature / Gesture | Model-based TSS ($p$=10) | | | | |
|---|---|---|---|---|---|
| | Fig. 6(d) | Fig. 10(a) | Fig. 10(b) | Fig. 10(c) | Fig. 10(d) |
| Front Kick | 100 | 80 | 90 | 90 | 90 |
| Side Kick | 100 | 100 | 100 | 100 | 100 |
| Knee Kick | 65 | 45 | 55 | 60 | 45 |
| Cross Punch | 55 | 25 | 55 | 60 | 40 |
| Uppercut | 80 | 60 | 70 | 70 | 80 |
| Jab | 85 | 70 | 85 | 85 | 85 |
| Stand | 75 | 70 | 75 | 75 | 75 |
| Step 1 | 100 | 100 | 100 | 100 | 100 |
| Step 2 | 80 | 70 | 80 | 80 | 80 |
| Average | 82.22 | 68.89 | 78.89 | 80.00 | 77.22 |

## 5. Conclusion

Feature extraction of gestures is essential for gesture recognition. In this paper, we introduce a new feature-extraction algorithm for dynamic gesture recognition using a model-based TSS. In the conventional TSS, it is difficult to represent detailed dynamic gesture features because relationships among body parts are ignored even though the number of features of the TSS could be substantially reduced. Calculating sums of distances on all features makes it difficult to identify gestures if the distances moved are pretty much the same, even though different body parts are used. To overcome such a problem, we propose the model-based TSS, which has advantages in gesture recognition, because it is possible to get the information on what body parts are moved and how much they are moved. In the model-based TSS, we calculate by grouping the parts with similar movements into sets and then apply to *taebo* feature recognition. According to experimental results, the recognition rate of the model-based TSS(84.44%) is increased by 56.11% compared to that of TSS (28.33%) when the number of sets is 6. The punch gestures have the lowest recognition rate in the model-based TSS because it cannot determine their direction. In the model-based TSS, more sets do not mean a higher result, even though there are slightly different results according to the number of sets. Also, different combinations of grouping in the same number of sets does not affect the recognition rates much if frequently used parts are separated from relatively little-used parts.

We believe that our proposed algorithm is applicable to more general gesture recognition tasks, and in future work, we will apply our algorithm to various gesture databases. We are also currently extending our work in several directions. For instance, we are exploring ways to incorporate a direction of gestures without an excessively large number of features and a velocity of body parts to distinguish more complex gestures. Finally, the multiclass SVM for time-varying data can be improved by using incremental learning.

## References

[1] S. Mitra, “Gesture recognition: A survey,” *IEEE trans. Systems, Man, and Cybernetics*, Part C, vol. 37, no. 3, pp. 311-324, 2007. http://dx.doi.org/doi:10.1109/TSMCC.2007.893280

[2] T. Wang, W. Hu, and T. Tan, “A survey of advances in vision-based human motion capture and analysis,” *Computer Vision and Image Understanding*, vol. 103, no. 2-3, pp. 90-126, 2006.

http://dx.doi.org/ doi:10.1109/34.910878 (CrossRef Link)

[3] A. Bobick and J. Davis, "The recognition of human movement using temporal templates," *IEEE trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 3, pp. 257–267, 2001.
http://dx.doi.org/ doi:10.1109/34.910878

[4] H. Li and M. Greenspan, "Model-based segmentation and recognition of dynamic gestures in continuous video streams," *Pattern Recognition*, vol. 44, no. 8, pp. 1614-1628, 2011.
http://dx.doi.org/ doi:10.1016/j.patcog.2010.12.014 (CrossRef Link)

[5] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," *IEEE trans. Pattern Analysis and Machine Intelligence*, vol. 29, no. 12, pp. 2247–2253, 2007.
http://dx.doi.org/ doi:10.1109/TPAMI.2007.70711
PMid:17934233 (CrossRef Link)

[6] Q. Shi, L. Wang, L. Cheng and A. Smola, "Discriminative human action segmentation and recognition using semi-markov model," in *proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-8, 2008.
http://dx.doi.org/ doi:10.1109/CVPR.2008.4587557

[7] I.N. Junejo, E. Dexter, I. Laptev, and P. Pérez, "View-independent action recognition from temporal self-similarities," *IEEE trans. Pattern and Machine Intelligence*, vol. 33, no. 1, pp. 172-185, 2011.
http://dx.doi.org/ doi:10.1109/TPAMI.2010.68
PMid:21088326 (CrossRef Link)

[8] H.K. Lee and J.H. Kim, "An HMM-based threshold model approach for gesture recognition," *IEEE trans. Pattern Analysis and Machine Intelligence*, vol. 21, no. 10, pp. 961-973, 1999.
http://dx.doi.org/ doi:10.1109/34.799904 (CrossRef Link)

[9] H. Kang, C. Lee, and K. Jung, "Recognition-based gesture spotting in video games", *Pattern Recognition Letters*, vol. 25, no. 15, pp. 1701-1714, 2004.
http://dx.doi.org/ doi:10.1016/j.patrec.2004.06.016 (CrossRef Link)

[10] L.R. Rabiner and B. Juang, "An introduction to Hidden Markov Models," *IEEE ASSP Magazine*, vol. 3, no. 1, pp. 4-16, 1986.
http://dx.doi.org/ doi:10.1109/MASSP.1986.1165342 (CrossRef Link)

[11] V.N. Vapnik, *The nature of statistical learning theory*, Springer-Verlag, 1995.
http://dx.doi.org/ doi:10.1007/978-1-4757-2440-0 (CrossRef Link)
PMid:8555380

[12] K. Crammer and Y. Singer, "On the algorithmic implementation of multi-class kernel-based vector machines," *Journal of Machine Learning Research*, vol. 2, pp. 265–292, 2001.

[13] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altu, "Support vector machine learning for interdependent and structured output spaces," in *proc. of the 21st International Conference on Machine Learning*, pp. 104, 2004.
http://dx.doi.org/ doi:10.1145/1015330.1015341

**Kyoung-Mi Lee** received a B.S. degree in computer science from Duksung Women's university in 1993, a M.S. degree in computer science from Yonsei university in 1996 and a Ph.D degree in computer sciences from the University of Iowa, Iowa City, in 2001. She is currently a professor in the Department of Computer Science, Duksung Women's University, Seoul, Korea. Her research interests include multimedia information processing, in particular, image and video processing, multimedia indexing and retrieval, and multimedia mining.

**Hye-Min Won** received a B.S. degree in computer science from Duksung Women's university in 2010 and a M.S. degree in computer science from Duksung Women's university in 2012. She is a researcher in Intelligent Multimedia Lab., Duksung Women's university, Seoul, Korea. Her research interests include computer vision, signal processing, and image security technology.