International Journal of Innovative Computing, Information and Control
Volume **12**, Number **6**, December **2016**

ICIC International ©2016 ISSN 1349-4198
pp. 2067–2087

NOVEL FACIAL FEATURE EXTRACTION TECHNIQUE FOR FACIAL EMOTION RECOGNITION SYSTEM BY USING DEPTH SENSOR

Nattawat Chanthaphan[1], Keiichi Uchimura[1], Takami Satonaka[2] and Tsuyoshi Makioka[2]

[1]Graduate School of Science and Technology
Kumamoto University
2-39-1 Kurokami, Kumamoto 860-8555, Japan
nattawat@st.cs.kumamoto-u.ac.jp; uchimura@cs.kumamoto-u.ac.jp

[2]Electronic Systems Technology and Information Systems Technology
Kumamoto Prefectural College of Technology
Haramizu 4455-1, Kikuyo, Kikuchi-gun, Kumamoto 869-1102, Japan
{Satonaka; makioka}@kumamoto-pct.ac.jp

Received June 2016; revised October 2016

Abstract. *In this paper, the novel approach to extracting the facial features from the movement of the facial skeleton model is introduced. In this approach, the data streams, the sequences of the normalized Euclidean distance between pairwise points on the facial skeleton model, are analyzed by using the Structured Streaming Skeleton (SSS) method to construct the SSS feature vectors – SSS method was firstly introduced in body gesture recognition system to handle the persistence of intra-class variations. The assessment of the system performance and accuracy was conducted by K-Nearest Neighbors (K-NN) and the Support Vector Machine (SVM). The fifteen participants' data set collected by our designed software was used in the experiment. By considering the facial emotions as facial gestures, SSS method was extended to handle the problems of intra-class variations in facial emotion recognition system. The present approach using K-NN attained a 91.17% +/− 2.36% of accuracy rate, which was better than a 67.83% +/− 4.54% of accuracy rate obtained by that using SVM. The comparison of the presented approach with the state-of-the-art was limited due to the unavailability of their data set. It could be concluded that our approach has achieved superiority over previously reported approaches by overcoming the intra-class variations.*

Keywords: Emotion recognition, Feature extraction, Structured streaming skeleton, Depth camera

1. **Introduction.** Due to the growing interests in machine learning, computer vision and artificial intelligence, academic researchers and industrial inventors have been trying to develop the systems that could think and behave in the same way as humans do. Human emotion recognition system which has an ability to distinguish or evaluate the emotions of human is one of them and our research interests are related to this sort of system.

In this paper, we will focus on the technique to extract the facial features for human facial emotion recognition. We have discussed the problem of selecting facial feature points on a human face and we have found out that there were still uncertainties as to how many feature points are necessary or what kind of feature vector could truly represent the facial emotions despite the achievements in many previous studies.

The conventional approaches employed texture feature and 3D structure, which were obtained from color/grayscale cameras and 3D camera, respectively. In our previous works [1,2], we introduced the novel skeleton based approach to extract facial features for facial emotion recognition by using a depth camera. The proposed approach constructed

a facial wireframe model from the depth image and analyzed sequences of the skeleton features for facial expressions. Since the data set which was relevant to our research was unavailable to the best of our knowledge, we prepared the data set by using our skeleton motion capture system.

The conventional approaches for feature extraction still suffer from some constraints and limitations which directly affect the system performance. The environmental conditions, for instance, viewpoint of camera, head pose, and head location, could be considered as obstacles which could be overcome by conducting head pose estimation and head localization. Many well-known algorithms are used as preprocessing before the process of feature extraction. The algorithm of the Active Shape Models (ASM) [3] is to learn a statistical shape model. The shape model is generated through the combination of shape variations. The learning process of the ASM will try to fit the shape model to the subject face. The shape model from the ASM will be fed to the other efficient search algorithm, namely Active Appearance Models (AAM) [4] to indicate exactly where and how a model is located in a picture frame. Next algorithm is the Constrained Local Models (CLM) [5] which works very well as the texture based approach. The CLM learns a shape model and texture variation from a labeled training set in the same way as the AAM, but the texture is sampled in patches around individual feature points. As described in their work, the CLM is more robust than the AAM. The 3D Morphable Model (3DMM) [6] is a face's 3D structure estimation or reconstruction technique. This face's 3D structure could be acquired by a single or more photographs. The pose of the subject face could be estimated through this generated 3D structure. Baltrusaitis et al. [7] have proposed another 3D approach utilizing a depth camera combining with the CLM. As might be seen in these works, they still get involved with texture or 2D image which could lead us to the other constraints and limitations of the image qualities, for example, the lighting condition, skin tone, noise. Therefore, we have introduced the use of the new efficient sensory device released in past few years which could help us reduce some efforts on the preprocessing.

The Microsoft Kinect V2 is a recently-developed depth sensor and can directly provide the 3D point cloud sequences for generating the motion streams of human face. Figure 1 illustrates the facial skeleton (wire-frame) generated by using Kinect HD face API. This API utilizes depth data as the based source and this depth data is estimated by special technique called Time of Flight (TOF) which uses infrared (IR) to measure depth or

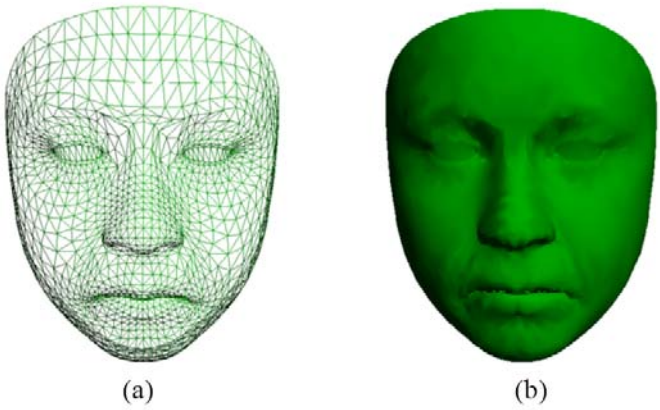


FIGURE 1. Example of facial skeleton generated by Kinect HD face API, (a) wire-frame mode, (b) solid mode

distance between the camera and the subject surface. By using IR, it can diminish the problems of image qualities.

By using the Microsoft Kinect V2, we have proposed the novel approach [1,2] based on the Structured Streaming Skeleton (SSS) method, which was firstly introduced in the human body gesture recognition by Zhao et al. [8]. They indicated possible solutions to solve problems of the following intra-class variations which occur in the system of body gesture recognition and it could obviously result in an outstanding performance and accuracy in their system.

(1) Viewpoint variation: This variation describes the relation between human body and viewpoint of the camera.

(2) Anthropometry variation: This variation is related to the difference between human body sizes which does not affect the human movement.

(3) Execution rate variation: This variation indicates the problem with different frame rate of the camera or the moving speed of human.

(4) Personal style variation: This variation is due to the difference of human performing their action differently.

Although these intra-class variations occur in the body gesture recognition system, by treating facial emotions as facial gestures, we could hypothesize that intra-class variations exist in the system of facial emotion recognition and we believed that by adopting this SSS approach, we could eliminate the intra-class variations from facial the emotion recognition system which might result in better performance and accuracy. The acquisition of dataset is the significant issue of the present approach. In previous work [1], we have the problem of unavailability of the standard database for facial emotion recognition based on the moving facial wire-frame model. Therefore, we addressed the problem of small dataset in [2] and we have obtained a promising performance by increasing the size of dataset. The acquisition of dataset is explored further in this paper.

This paper consists of five sections. In the second section, we introduce the related works regarding the emotion recognition system. In the third section, we describe the procedure of the data stream generation and SSS feature extraction in detail. In the fourth section, we present the three experimental results. Finally, we will conclude our work and show you our future work in the fifth section.

2. **Related Works.** As we have mentioned various variations in the introduction, image pre-processing is conducted prior to the feature extraction phase. One of the pre-processing technique is the head pose estimation. Baggio et al. [9] introduced the well-known method for 3D head pose estimation by using the AAM and POSIT (Pose from Orthography and Scaling with Iterations) [10]. It utilized the principal component analysis (PCA) to reduce the number of parameters of model and did the Delaunay Triangulation (DT) [11] to create the statistical model of the AAM. These works, they still get involved with texture or 2D image.

Zhu and Ramanan [12] showed that the RGB camera-based approach consumed very expensive computation time. The alternate approach was based on the new sensory device named Microsoft Kinect whose capabilities were described in the article [13] of Zhang. We could reduce our efforts on the pre-processing by utilizing the Microsoft Kinect which could provide the sequence of the facial skeleton or wireframe responding to the changes on human face; therefore, we could say that our approach was 3D based approach.

Piana et al. [14] also made use of the Kinect to recognize human emotion. They obtained the 3D human body skeleton from the Kinect, whereas the overall accuracy to distinguish human emotions was just 61.3%. The result suggested that the human body gestures were not a good description for emotions.

Huang et al. [15] proposed the Modified Active Shape Model (MASM), which was modification of the ASM. They presented the triangular facial feature extraction method to reduce the effects of environmental variation and noisy facial feature. They claimed that the MASM could offer a faster way of facial landmark searching, and could greatly reduce feature dimensions and improve the performance of emotion recognition. However, Mao et al. [16] described that the 2D images which were captured by RGB camera were insufficient to represent the geometrical feature since the human faces were 3D object. 2D images which were captured by RGB camera were insufficient to represent the geometrical feature. Therefore, they utilized 3D object features consisting of the Animation Unit (AU) and Feature Point Positions (FPPs). The Kinect face tracking Software Development Kit (SDK) supports six AUs (brow raiser, brow lower, lip raiser, lip stretcher, lip corner depressor and jaw lower), whose model parameters range from -1 to 1. It also provides the FPPs which are 45 coordinates of 45 points. The classification performance was computed from the AUs and the FPPs by fusing the results of 30 consequent frames. This was regarded as the pre-segmentation based approach which could not handle execution rate variation. For the evaluation, they constructed the UJS Kinect Emotion Database (UJS-KED), which were obtained from 10 actors with variations in five poses ($-30°$, $-15°$, $0°$, $15°$ and $30°$) for seven emotions (anger, disgust, fear, happiness, neutral, sadness and surprise). Note that multi-pose samples were required in their database because the approach could not deal with the viewpoint variation and the anthropometry variation.

Zhao et al. in [8] introduced the novel approach called Structured Streaming Skeleton (SSS). Their approach succeeded in handling all those intra-class variations by utilizing the streams generated from the moving body skeleton. Kinect V1 was used in the research. Even though the system was designed for body gestures recognition, it might be applicable to facial expression recognition by treating facial expressions as facial gestures. Therefore, we have decided to modify the SSS feature extraction method for our approach in order to distinguish a human facial emotion rather than a human body gesture.

The other approaches related to the SSS feature extraction were the Facial Action Coding System (FACS) proposed by Ekman and Friesen [17] and the Dynamic Time Warping (DTW) distance. The FACS is based on facial muscle change and can characterize facial actions that constitute an expression irrespective of emotions. The DTW method was introduced to find the minimal alignment between two sequences by Sakoe and Chiba [18] and was further extended to find the optimal end frame of scanning sequences by Sakurai et al. [19].

3. Proposed Approach.

The present framework in Figure 2 could be separated into three phases: data stream generation, SSS feature extraction and classification respectively. Each phase will be explained step by step in following subsections. The overall of the framework might look quite similar to the framework of SSS feature extraction approach for body gesture recognition. However, there are significant modifications in some parts of the present framework, which will be explained later in this section.

3.1. Data stream generation.

At first, we present the brief explanations of the facial wire-frame model. We construct the facial wire-frame model consisting of 1347 vertices by using the HD face Application Programming Interface (API) for Kinect. The appropriate vertices corresponding to the feature points of the FACS of Ekman et al. [17] are selected for generating the motion streams. Table 1 summarizes the corresponding points for representing each emotion.

We present the calculation of the motion streams by using Equations (1), (2) and (3). Figure 3 shows the facial skeleton model consisting of 18 feature points. Each joint (i) has

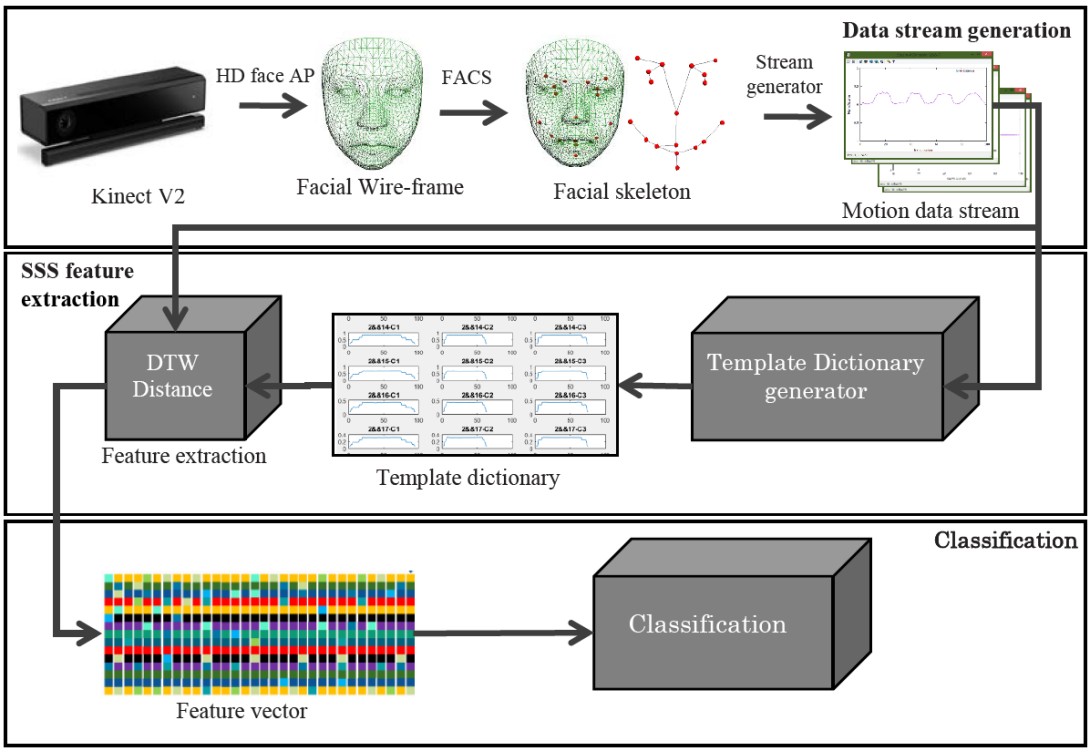


FIGURE 2. Framework of the proposed approach

TABLE 1. FACS (Facial Action Coding System)

Emotions	FACS
Anger	Brow Lowerer + Upper Lid Raiser + Lid Tightener + Lip Tightener
Contempt	Lip Corner Puller + Dimpler
Disgust	Nose Wrinkler + Lip Corner Depressor + Lower Lip Depressor
Fear	Inner Brow Raiser + Outer Brow Raiser + Brow Lowerer + Upper Lid Raiser + Lid Tightener + Lip Stretcher + Jaw Drop
Happiness	Cheek Raiser + Lip Corner Puller
Sadness	Inner Brow Raiser + Brow Lowerer + Lip Corner Depressor
Surprise	Inner Brow Raiser + Outer Brow Raiser + Upper Lid Raiser + Jaw Drop

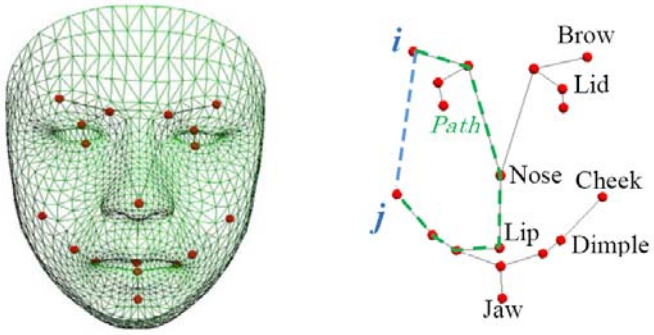


FIGURE 3. Facial skeleton model and feature points

3 coordinates $p_i(t) = (x_i(t), y_i(t), z_i(t))$ at frame t. The red vertices are based on FACS. Every point is linked together to construct a facial skeleton. The blue dotted line indicates the direct distance between points i and j. The Euclidean distance $E(p_i(t), p_j(t))$ between

$p_i(t)$ and $p_j(t)$ is normalized by using the path distance $Path_{ij}(t)$. The green dotted one indicates the path distance between points i and j. For each pairwise joints i and j, $1 <= i < j <= N$, we calculate their normalized distances $S_{ij}(t)$ as shown in Equation (1).

$$S_{ij}(t) = \frac{E(p_i(t), p_j(t))}{Path_{ij}(t)} \tag{1}$$

$$Path_{ij}(t) = \sum_{m=1}^{\#Node\ between\ point\ ij\ -\ 1} E\left(p_{L_m}(t), p_{L_{m+1}}(t)\right) \tag{2}$$

$$Rows\ of\ Streams = \frac{N(N-1)}{2} \tag{3}$$

where L_m is sorted point index list for particular $Path_{ij}$ indexed by m. The combination ${}_NC_2$ becomes 153 for $N = 18$. Finally, we will have the normalized distances of all 153 rows for all frames and the motion data streams are generated.

3.2. **SSS feature extraction.** We present the procedure of the SSS feature extraction consisting of two steps: template dictionary generation and feature extraction.

3.2.1. *Template dictionary generation.* The SSS feature extraction technique is originally designed for a body gesture stream. We modify the SSS feature extraction technique for generating facial gesture stream by considering the significant difference between a facial gesture stream and a body gesture stream. We present the difference between a body gesture stream and a facial gesture (emotion) stream. Figure 4 shows the signal waveform for a body gesture of hand waving and that of a facial gesture of smiling to represent each gesture. The body gesture contains several numbers of waves. In contrast, the facial gesture does a single pulse wave with one positive edge, one steady state and one negative edge. Consequently, the size of the template dictionary identifying facial gesture is less than that identifying body gesture if we use the same method to generate a template dictionary.

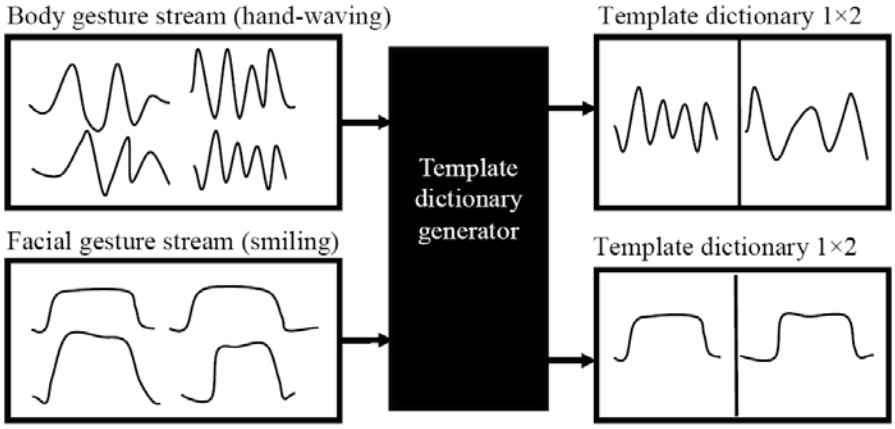


FIGURE 4. Examples of body gesture streams and facial gesture streams (Example $G = 2$)

Our approach can dispense with the warping process, which is required in the original SSS feature extraction technique [8]. Every signal is warped to be the same period before the averaging process.

We briefly explain the template dictionary generation as follows. Figure 5 shows an example of data streams for generating template dictionary. Firstly, we manually split the

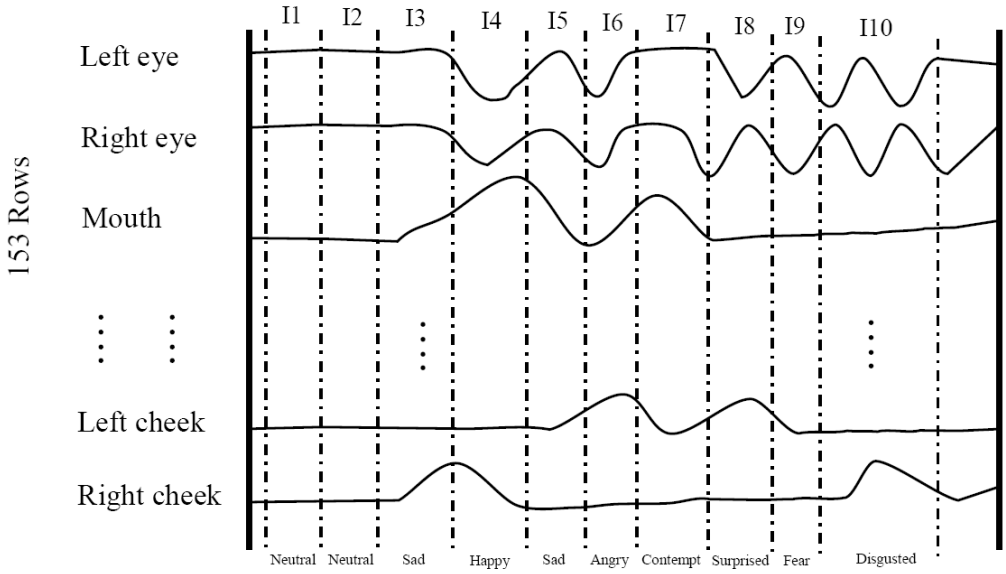


FIGURE 5. Examples of data streams for generating template dictionary

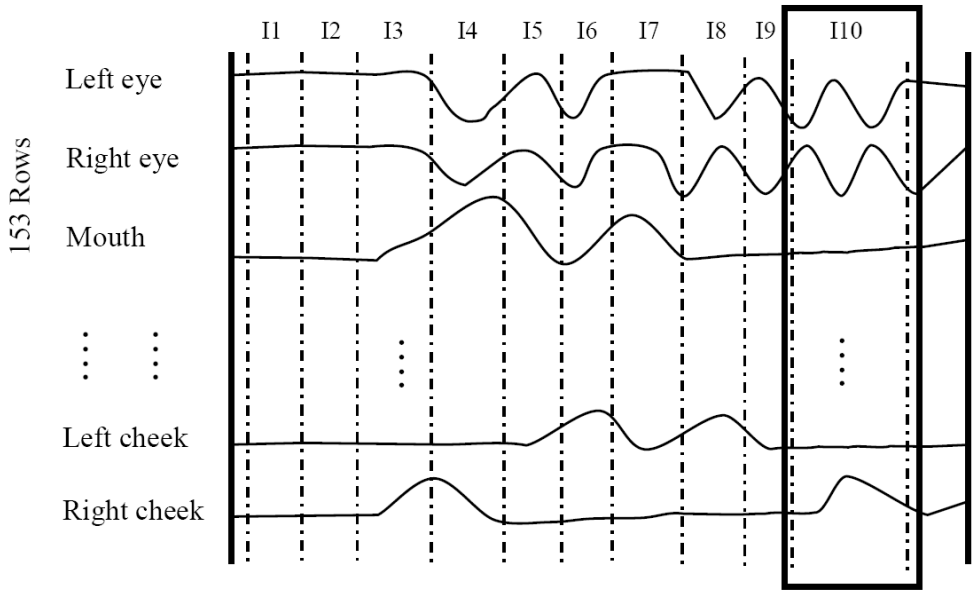


FIGURE 6. Example of the reference instance (I10)

motion streams into several gesture instances. Then we have an emotion per instance. The process of splitting is performed only once to generate the template dictionary. Secondly, we select the instance with the highest frame number as the reference instance. Figure 6 shows an example of the reference instance. The longest sequence (I10) is selected in this example. We compute the DTW distance between the reference sequence and the rest of sequences within the same row. Table 2 shows the DTW distance between each instance and the reference one.

We sort gesture instances in ascending order of the DTW distances for each of these pairs by using quick sort. It is noted that DTW distance can indicate the similarity between two sequences – low value means high similarity, high value means low similarity

TABLE 2. DTW distances between each instance and reference (I10)

	I1	I2	I3	I4	I5	I6	I7	I8	I9	I10
Left eye	31	28	28	17	10	15	30	12	11	0
Right eye	20	21	18	10	9	11	8	5	7	0
Mouth	10	11	20	25	23	18	21	11	10	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
Left cheek	11	10	11	11	16	20	26	24	18	0
Right cheek	22	25	15	17	25	26	22	24	28	0

TABLE 3. Sorted DTW distances for five clusters

	Cluster 1		Cluster 2		Cluster 3		Cluster 4		Cluster 5	
Left eye	I10	I5	I9	I8	I6	I4	I2	I3	I7	I1
	0	10	11	12	15	17	28	28	30	31
Right eye	I10	I8	I9	I7	I5	I4	I6	I3	I1	I2
	0	5	7	8	9	10	11	18	20	21
Mouth	I10	I1	I9	I2	I8	I6	I3	I7	I5	I4
	0	10	10	11	11	18	20	21	23	25
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
Left cheek	I10	I2	I1	I3	I4	I5	I9	I6	I8	I7
	0	10	11	11	11	16	18	20	24	26
Right cheek	I10	I3	I4	I1	I7	I8	I2	I5	I6	I9
	0	15	17	22	22	24	25	25	26	28

and zero means exactly the same. The gesture instances that resemble each other will be sorted to be closely located by using the DTW distance.

Next, we categorize the gesture instances that have DTW distance being close to each other as the same cluster as shown in Table 3, and then average all gesture instances in the same cluster to be just one sequence per cluster. Therefore an important parameter here is number of clusters (G) which affects the dimension of template dictionary. To average the gesture instances that have different frame numbers, we need to shift the shorter sequence with the difference of frame number divided by two. We obtain the motion template dictionary from the following procedure for clusters of gesture instances.

Input: Cluster of Gesture instance
Output: Template dictionary
For $i := 0$ to $R - 1$
  For $j := 0$ to $G - 1$
    For $k := 0$ to $N[i][j] - 1$

$$Fs := \frac{F[i][j] - f[i][j][k]}{2}$$

$$\text{For } l := Fs \text{ to } f[i][j][k] + Fs$$

$$A[i][j][l] := \frac{A[i][j][l] + S[i][j][k][l - Fs]}{N[i][j]}$$

The first subscript i is the row index ranging from 0 to $R-1$ and the second subscript j is the cluster index ranging from 0 to $G-1$. The R or G is the index size. The third subscript k is the sequence index ranging from 0 to $N[i][j]-1$. The $N[i][j]$ is the

sequence size in the ith row of the jth cluster. The fourth subscript l is the frame index ranging from Fs to $Fs + f[i][j][k]$. The Fs denotes the frame number to be shifted and the $f[i][j][k]$ is the maximal frame number of the kth sequence in the ith row of the jth cluster. The $F[i][j]$ is the frame size of the longest sequence in the ith row of the jth cluster. The $S[i][j][k][l]$ is the value of the lth frame of the kth sequence in the ith row of the jth cluster and $N[i][j]$ is the maximal sequence number in the ith row of the jth cluster. $A[i][j][l]$ is averaged sequence (template dictionary) of the lth frame in the ith row of the jth cluster. Then, we will have the template dictionary generated from all gesture instances. Figure 7 shows the template dictionary. The number of clusters per row is fixed to be five.

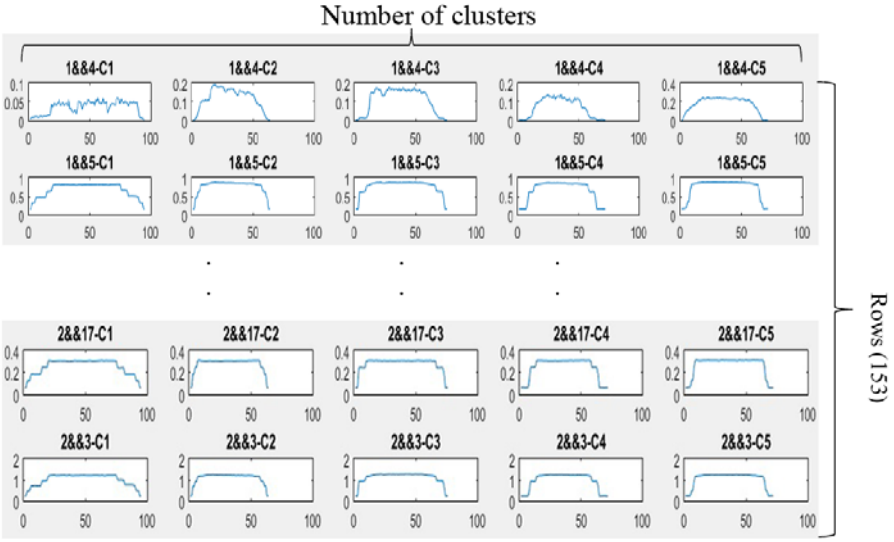


FIGURE 7. Example of template dictionary (rows × number of clusters)

3.2.2. *Feature extraction.* We explain the procedure of the feature extraction by using the template dictionary which is generated from the gesture instances of human face. To obtain the feature vectors from the stream, we calculate DTW distance starting from the current frame for each sequence stored in the template dictionary. It is noted that all the streams are required to be scanned again. The minimal DTW distance will be selected as one attribute of feature vector. Equations (4), (5), (6) and (7) are the equations to calculate DTW distance.

Given sequences X and Y

$$D(X, Y) = f(n, m) \tag{4}$$

$$f(0, 0) = 0, \quad f(i, 0) = f(0, j) = \infty \tag{5}$$

$$f(i, j) = (x_i - y_j)^2 + \min \begin{cases} f(i, j-1) \\ f(i-1, j) \\ f(i-1, j-1) \end{cases} \qquad (i = 1, \ldots, n; j = 1, \ldots, m) \tag{6}$$

$$C = \begin{bmatrix} 0 & \infty & \cdots & \infty \\ \infty & \ddots & & \\ \vdots & & \ddots & \\ \infty & & & f(n, m) \end{bmatrix}; \quad C \in R^{n \times m} \tag{7}$$

where $D(X, Y)$ is DTW distance between X and Y, $f(i, j)$ is the elements i, j, C is DTW cost matrix, and n, m are maximum frame of sequence X and Y respectively. In our

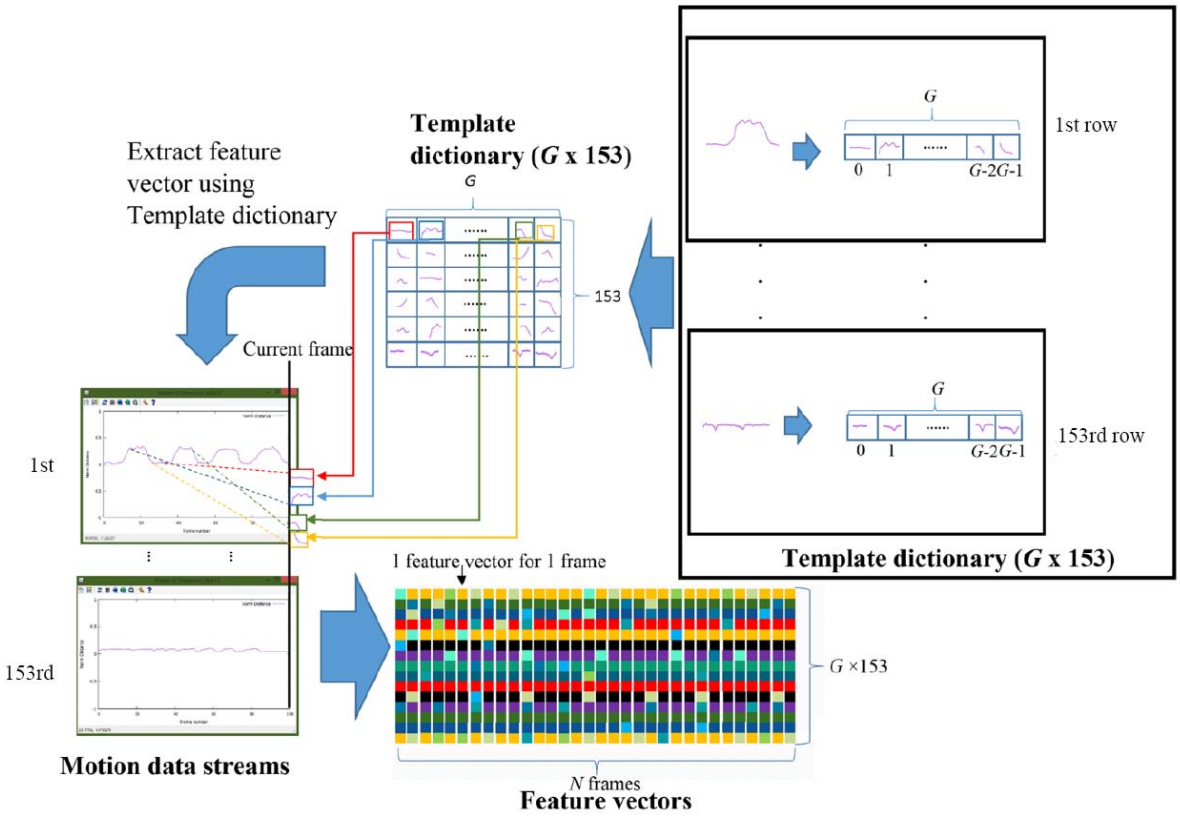


FIGURE 8. The overall process of feature extraction

approach, the stream monitoring technique proposed by Sakurai et al. [19] is applied to determining the number of frame of sequence Y by finding the optimal ending point.

Figure 8 illustrates the overall process of the feature extraction step. Each row of streams will produce G attributes of feature vector. After the process of scanning, the feature vector for one frame will consist of G (number of clusters of template dictionary) $\times$ number of rows attributes.

Each row of streams, starting from current frame to the optimal ending point, will be scanned with every sequence in the same row of template dictionary to get DTW distance. Each row will produce G attributes of feature vector. Therefore, after this step, the feature vector for one frame will consist of $153 \times 5 = 765$ attributes (in case of $G = 5$) and be ready for the classification process.

4. Experiment.

4.1. Dataset acquisition and feature extraction.

We present the facial expression recognition performance of the present approach using the SSS method. To construct the recognition system, we prepare the moving sequences of coordinates on facial wireframe as the training dataset since there was no dataset available for our proposed approach. We designed the facial motion stream acquisition software to extract a facial wire-frame model from 3D point cloud sequences collected by the Kinect. Figure 9 illustrates the layout of Graphical User Interface (GUI) of our software developed by using Kinect for Windows SDK 2.0.

The dataset acquisition software supports eight emotions including happiness, sadness, surprise, fear, anger, disgust, contempt and neutral. The participants are instructed to express each emotion for 15 seconds (325 frames). We construct a facial wire-frame model

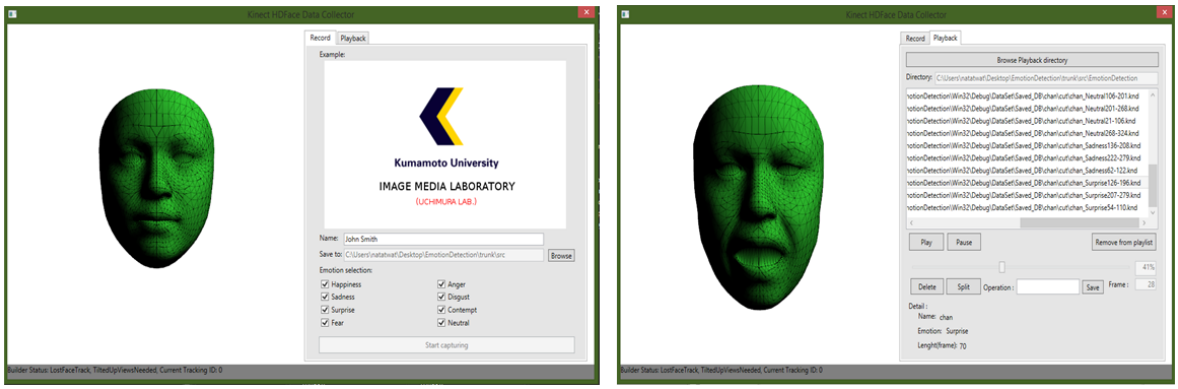


FIGURE 9. Layout of Graphical User Interface of Dataset Acquisition Software

with 1347 vertices for each frame. The number of vertices is reduced by using the FACS based selection process.

We have collected dataset from 15 actors for our experiment (eight emotions per each). In each emotion, they have been asked to freely act three times of emotions according with the emotion label shown on the screen because we wanted to get the data that intuitively represented their emotions. Therefore we had $325 \times 8 \times 15 = 39,000$ frames which could be separated into $3 \times 8 \times 15 = 360$ gesture instances for template dictionary generation phase. The system environment for our experiment was Intel® core^{TM} i5-4570 3.20 GHz, 4 GB DDR2, Windows 8.1x64 and GPU NVIDIA GeForce GTX 750 Ti.

We implemented the present feature extraction into parallel processing by using native C++ CUDA [20] program, which could activate multi-thread processing on the NVIDIA GeForce GPU.

4.2. **Classification and evaluation.** For the classification models used in the experiment, we have used K-Nearest Neighbors (K-NN) and Support Vector Machine (SVM). K-NN is known as non-parametric lazy learning algorithm used for classification and regression. It utilizes the k closest training examples with the input in the feature space. The benefits of K-NN over the other classification models are fast, robust to noisy training data, effective if the training data is large. SVM is one of the most popular algorithms to solve classification and regression problem. Unlike K-NN, SVM needs learning phase to learn the classification model. Furthermore, SVM supports several kernel functions to transform the data into a higher dimensional space, where each feature pattern becomes easily separable. There are plenty of kernels that can be used depending on what kind of dataset is used, for instance, linear, polynomial, Radial Basis Function (RBF), dot, sigmoid. 10-fold cross validation was used for the performance and accuracy evaluation. We evaluated the recognition performance by using RapidMiner, which is an open software for machine learning as described in paper of Jović et al. in [21].

4.3. **Experiments.** We have conducted three experiments to obtain the best sort of feature vectors as well as model parameters. In each experiment, the factors attaining the best accuracy are selected as the base conditions for the next experiment. We can expect further improvements in the following experiments.

4.3.1. *The first experiment on the SSS and stream feature vectors.* In this first experiment, we conducted the qualitative experiment by using two kinds of feature vectors: the SSS feature vector and the simple stream feature vector. The SSS feature vector is prepared by using the method described in third section of this paper and consists of 765 rows. The

stream feature vector consists of 153 attributes to define Euclidean distance between the particular pairs on the facial skeleton. It seems that each attribute of the stream feature vector is self-descriptive by itself. Due to the constraints of time and system environment, we have fixed the number of clusters in template dictionary to five clusters ($G = 5$) in this experiment.

In the paper of Kim et al. [22], $k = 5$ was the best value. A linear kernel function was used in SVM. In the paper of Hsu et al. [23], the linear kernel function was highly recommended when the numbers of instances and features were high. Therefore, the linear kernel function is to be selected for SVM in the first experiment.

Tables 4, 5, 6 and 7 present the results of facial expression recognition by using the confusion matrices. Table 8 shows accuracy comparison between our approach and state-of-the-art approach [16].

TABLE 4. Stream feature vector with K-NN

		Predicted							
		Happiness	Sadness	Surprise	Fear	Anger	Disgust	Contempt	Neutral
Actual class	Happiness	**82.28**	1.90	5.06	5.06	2.53	2.53	0.63	0.00
	Sadness	1.71	**68.38**	5.98	3.42	10.26	5.98	2.56	1.71
	Surprise	4.32	5.04	**75.54**	12.23	0.00	2.88	0.00	0.00
	Fear	1.67	5.83	7.50	**74.17**	4.17	5.83	0.83	0.00
	Anger	4.73	5.33	1.78	7.10	**64.50**	10.65	3.55	2.37
	Disgust	3.36	2.52	6.72	4.20	9.24	**73.11**	0.84	0.00
	Contempt	0.79	7.09	3.15	2.36	0.79	2.36	**80.31**	3.15
	Neutral	3.28	7.10	3.28	1.09	4.92	3.83	12.02	**64.48**

TABLE 5. Stream feature vector with SVM

		Predicted							
		Happiness	Sadness	Surprise	Fear	Anger	Disgust	Contempt	Neutral
Actual class	Happiness	**74.84**	3.77	3.14	5.66	2.52	3.14	3.14	3.77
	Sadness	9.16	**38.17**	10.69	3.05	15.27	1.53	13.74	8.40
	Surprise	6.00	4.67	**68.67**	1.33	2.67	2.67	9.33	4.67
	Fear	12.86	9.29	47.86	**12.14**	2.14	7.86	3.57	4.29
	Anger	12.58	10.60	5.30	4.64	**44.37**	6.62	11.26	4.64
	Disgust	7.30	5.84	16.06	5.84	22.63	**25.55**	14.60	2.19
	Contempt	16.91	8.82	2.94	0.00	5.15	6.62	**52.21**	7.35
	Neutral	13.28	7.81	7.03	3.91	12.50	7.81	18.75	**28.91**

TABLE 6. SSS feature vector with K-NN

		Predicted							
		Happiness	Sadness	Surprise	Fear	Anger	Disgust	Contempt	Neutral
Actual class	Happiness	**82.67**	0.00	0.00	8.00	0.00	2.00	0.67	6.67
	Sadness	0.67	**83.33**	2.00	3.33	2.00	3.33	2.67	2.67
	Surprise	1.33	8.67	**72.67**	9.33	0.67	4.00	1.33	2.00
	Fear	2.67	1.33	4.67	**77.33**	3.33	5.33	2.67	2.67
	Anger	0.00	2.67	5.33	0.00	**78.67**	3.33	6.00	4.00
	Disgust	0.00	2.00	1.33	1.33	4.00	**81.33**	8.00	2.00
	Contempt	0.67	1.33	0.00	0.00	3.33	1.33	**91.33**	2.00
	Neutral	2.67	0.67	0.00	0.67	1.33	0.00	0.00	**94.67**

TABLE 7. SSS feature vector with SVM

		Predicted							
		Happiness	Sadness	Surprise	Fear	Anger	Disgust	Contempt	Neutral
Actual class	Happiness	**88.00**	0.00	1.33	0.67	2.67	0.00	2.00	5.33
	Sadness	5.33	**70.00**	5.33	1.33	2.00	5.33	2.67	8.00
	Surprise	2.67	3.33	**76.67**	8.00	0.67	2.00	3.33	3.33
	Fear	20.00	3.33	16.00	**52.00**	0.67	2.00	3.33	2.67
	Anger	4.67	4.00	2.00	0.67	**52.00**	6.67	12.00	18.00
	Disgust	8.67	11.33	5.33	3.33	9.33	**48.00**	9.33	4.67
	Contempt	11.33	1.33	0.00	1.33	10.00	11.33	**55.33**	9.33
	Neutral	4.67	0.00	2.00	3.33	4.00	0.00	5.33	**80.67**

TABLE 8. Accuracy comparison

Approach↓	Happiness	Sadness	Surprise	Fear	Anger	Disgust	Neutral	**Average**
State-of-the-art approach [16]	75.58	73.74	96.40	80.00	79.27	79.54	79.52	**80.57**
SSS feature (K-NN)	82.67	83.33	72.67	77.33	78.67	81.33	94.67	**81.52**
SSS feature (SVM)	88.00	70.00	76.67	52.00	52.00	48.00	80.67	**66.76**
Stream feature (K-NN)	82.28	68.38	75.54	74.17	64.50	73.11	64.48	**71.78**
Stream feature (SVM)	74.84	38.17	68.67	12.14	44.37	25.55	28.91	**41.80**

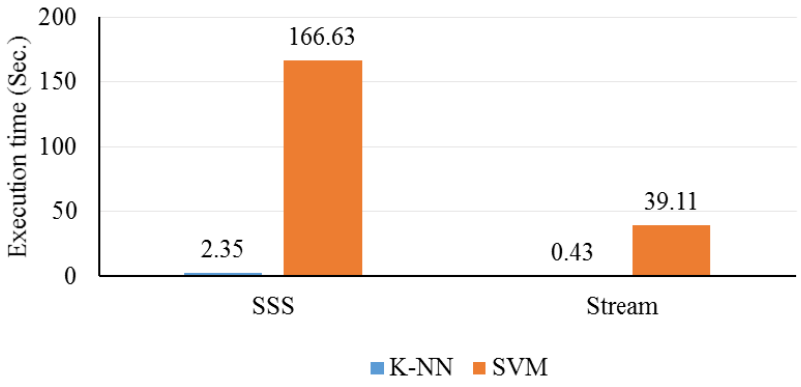


FIGURE 10. Execution time required for training and testing

The K-NN algorithm using the SSS features achieved accurate rate of 82.75% +/− 3.03%, which was better than these of 72.45% +/− 3.82% obtained by the K-NN algorithm using the stream features. The SVM using the SSS features provided lower accurate rates of 65.33% +/− 3.3%. In this experiment, a linear function is selected as a kernel function of the SVM. The SVM using the stream features resulted in the lowest accurate rates of 44.07% +/− 4.7%. The results of the K-NN algorithm is better than these of the SVM. The performance might be improved slightly by selecting another kernel functions.

Figure 10 presents the comparison of the execution times required for training and testing of the K-NN algorithm and the SVM using the SSS feature and/or the stream feature.

The K-NN algorithm using the SSS feature and the stream one required execution times of 2.35 and 0.43 seconds, respectively. The ratio of 2.35 to 0.43 becomes 5.60. The execution time using the stream features is 5.60 times faster than that using the SSS features for the K-NN algorithms. The SVM using the SSS feature and the stream one spent these of 166.63 and 39.11 seconds, respectively. The ratio of 166.63 to 39.11 becomes 4.86. The execution time using the stream features is 4.86 times faster than that using the SSS features for the SVM.

We have to consider the trade-offs between accuracy rates and the execution times in real-time applications. The K-NN algorithm using the SSS feature achieved the best accuracy rate and the execution time of 2.35 second is 5.6 times slower than that for the K-NN algorithm using the stream feature. The stream feature vector requires less computation times and becomes one of good candidates set of features for uses in real time application.

To the best of our knowledge, we conducted fair comparisons between our approach and the state-of-the-art approach [16] by considering different factors such as data set, and number of classes. Table 8 shows the result of our approach and the state-of-the-art approach. Since they did not include "contempt" emotion in their list, we had to recalculate the accuracy again. The presented approach using the SSS feature attained an accuracy rate of 81.52% which was comparable to that of the state-of-the-art approach and it could reduce the effects of intra-class variations in the system compared with the state-of-the-art approach as follows.

(a) To reduce the viewpoint variation, we employed normalized distance between each pairwise joints as the fundamental elements for feature extraction instead of utilizing image pixel values. Then, the orientation or direction of face and camera would not affect the performance and accuracy of the system.
(b) To eliminate the anthropometry variation, the direct distance between each pairwise joint was normalized by a path distance between pairwise joint. Therefore, the size or the distance between human face and camera would not affect the performance and the accuracy of the recognition.
(c) To deal with the execution rate variation, the well-known DTW method was applied to the present SSS feature extraction method, which enables us to find the minimal alignment between two sequences with different lengths and frequencies.
(d) To deal with personal style variation, we constructed the template dictionary to represent all gesture instances. The sequences in the template dictionary were fine-tuned to facial-part-level movement as described in the paper [8].

By eliminating intra-class variation from the system, our proposed approach could outperformed the state-of-the-art approach in terms of intra-class variation handling. Some constraints in state-of-the-art approach can be addressed. For example, the dataset does not need to be limited to five poses ($-30°$, $-15°$, $0°$, $15°$ and $30°$). As the normalized Euclidean distance based attribute, only one pose is enough to represent every pose. Therefore, the size of dataset will be smaller than the dataset used in the state-of-the-art approach.

4.3.2. *The second experiment on K-NN and SVM classifiers.* We investigate the performance dependence on the model parameters of the K-NN and SVM classifiers, which were also employed in our previous experiments. In this experiment, we present a detailed explanation of the performance of K-NN and SVM classifiers. For K-NN, it has a k parameter representing the number of neighbors to be used for considering the cluster in which the testing data belongs to. By changing this k parameter, it might result in a better accuracy or a worse accuracy. In the second experiment, we have increased the

number of neighbors (k) from one to ten and measured both accuracy and execution time. SVM, on the other hand, has several kernel functions as the alternatives for us to find the best one in terms of accuracy and execution time. Five kernel functions are used, dot kernel function, radial basis function, the linear function, the second degree and third degree polynomial functions. The results of 10-fold cross validation will be presented for these classifiers with various k values and kernel functions. For classification tasks, we employ the RapidMiner which is a popular, reliable open source software.

Figure 11 shows the results of the K-NN and SVM classifiers. The 1,200 frames, which were sampled from 39,000 frames, were used for training and testing. Figure 11(a) shows the performance dependence on the k parameter of K-NN classifier. The accuracy rate is decreased from 90.33% to 75.5% linearly when the value of k, namely, the number of neighbors is increased from one to ten. The execution time of K-NN varied between 1.79 and 1.67 as shown in Figure 11(b).

Figure 11(c) and (d) present the results of the SVM classifier with five kernel functions, namely, the dot kernel (inner product) function, radial basis function, the linear function, the second degree and third degree polynomial functions. The accuracy rates of the linear function and the second degree polynomial function were 64.5% and 67.0%, respectively. These figures are lower than 75.5%, that is, the worst accuracy rate of the K-NN classifier. The execution times of the linear function and the second degree polynomial function are 38.38 and 278.52 seconds, respectively. The execution times of SVM classifier became extremely higher than these of the K-NN classifier as shown in Figure 11(d).

The present results suggested that the K-NN classifier outperformed the SVM classifier. In the present feature extraction technique, the dimension of feature vectors became too large due to the high numbers of rows and attributes. This degraded the performance of the SVM classifiers even though they have many kernel functions to transform the input feature vectors onto a higher dimensional space.

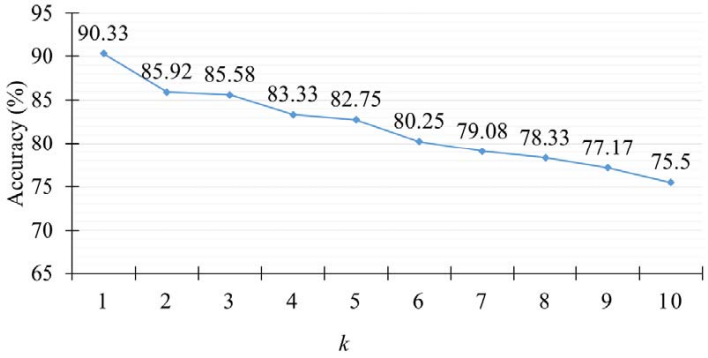
Tables 9 and 10 show the confusion matrices of K-NN with $k = 1$ and SVM with second degree polynomial respectively. The accuracy rates for 'Happiness and 'Neutral are 94.67% and 96.0%, respectively. According to our observations, these motions of the Neutral and Happiness expressions seem to be fairly distinguishable. The accuracy rate for 'Fear' is 82.67%. The motion of the 'Fear' expression is similar to that of the 'Surprise' expression. There is a high probability that the 'Fear' expression would be misclassified as the 'Surprise' expression.

Table 11 shows the accuracy comparison between proposed approach and state-of-the-art approach [16]. This comparison is based on the same condition as in the first experiment.

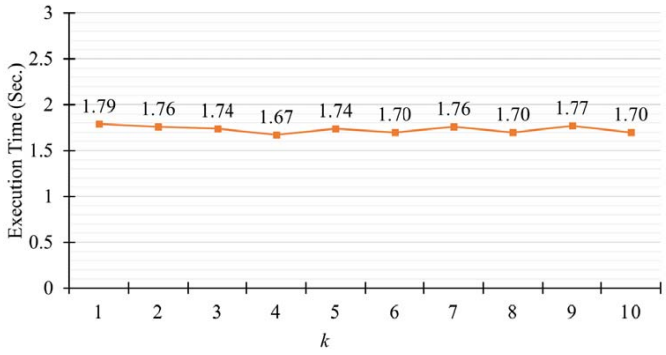
TABLE 9. SSS feature vector with K-NN ($k = 1$)

		Predicted							
		Happiness	Sadness	Surprise	Fear	Anger	Disgust	Contempt	Neutral
Actual class	Happiness	**94.67**	0.67	0.00	1.33	0.00	0.00	0.67	2.67
	Sadness	2.00	**88.67**	2.00	1.33	1.33	1.33	2.00	1.33
	Surprise	1.33	2.67	**88.67**	4.00	0.00	1.33	0.67	1.33
	Fear	5.33	2.67	3.33	**82.67**	0.67	3.33	0.67	1.33
	Anger	0.00	2.67	4.00	0.00	**86.00**	2.67	2.00	2.67
	Disgust	0.00	0.00	0.67	0.67	0.67	**94.67**	3.33	0.00
	Contempt	1.33	1.33	0.00	1.33	2.67	0.67	**91.33**	1.33
	Neutral	1.33	0.67	0.00	0.67	0.67	0.00	0.67	**96.00**

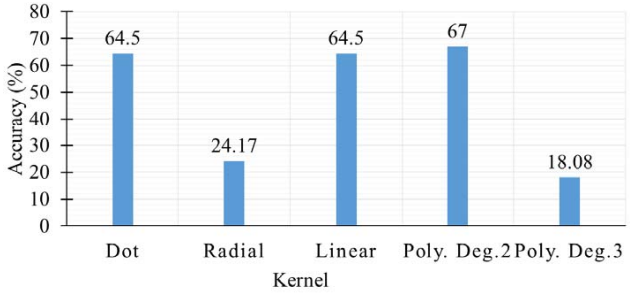
The average accuracy of the proposed approach is 66.66% for SVM with second degree polynomial and 90.19% with $k = 1$ for K-NN which is higher than the average accuracy of the state-of-the-art approach which is 80.57%.



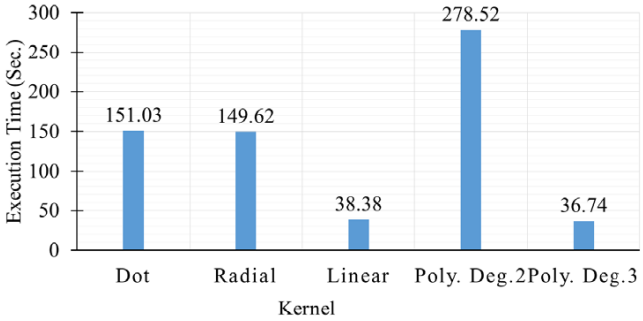
(a)



(b)



(c)



(d)

FIGURE 11. Performance dependence on the model parameters, (a) accuracy of K-NN, (b) execution time of K-NN, (c) accuracy of SVM, (d) execution time of SVM

TABLE 10. SSS feature vector with SVM (second degree polynomial)

		Predicted							
		Happiness	Sadness	Surprise	Fear	Anger	Disgust	Contempt	Neutral
Actual class	Happiness	**78.00**	4.67	0.00	2.67	2.67	4.00	0.00	8.00
	Sadness	2.67	**69.33**	4.00	5.33	3.33	1.33	4.00	10.00
	Surprise	4.67	6.00	**67.33**	10.00	1.33	2.00	3.33	5.33
	Fear	15.33	7.33	9.33	**50.00**	3.33	3.33	4.00	7.33
	Anger	2.00	4.00	0.67	2.00	**63.33**	2.67	14.00	11.33
	Disgust	6.00	7.33	1.33	4.67	12.00	**55.33**	8.67	4.67
	Contempt	6.00	1.33	0.00	2.67	2.67	7.33	**69.33**	10.67
	Neutral	5.33	4.00	0.67	2.67	1.33	2.67	0.00	**83.33**

TABLE 11. Accuracy comparison with state-of-the-art approach

Approach↓	Happiness	Sadness	Surprise	Fear	Anger	Disgust	Neutral	**Average**
State-of-the-art approach [16]	75.58	73.74	96.40	80.00	79.27	79.54	79.52	**80.57**
Proposed approach*1	94.67	88.67	88.67	82.67	86.00	94.67	96.00	**90.19**
Proposed approach*2	78.00	69.33	67.33	50.00	63.33	55.33	83.33	**66.66**

*1 K-NN with $k = 1$, *2 SVM with second degree polynomial

4.3.3. *The third experiment on template dictionary.* In this experiment, we investigate the performance dependence on the number of clusters (G) in the template dictionary. The dimension of template dictionary depends on G parameter and the size of feature vector depends on that dimension of template dictionary. Therefore, we need to evaluate the performance dependence on G. The number of clusters was set to five in the first and second experiments. We prepare the template dictionaries with two, five and seven clusters, that is, $G = 2$, $G = 5$ and $G = 7$ by following the previous procedures and construct the K-NN and SVM classifiers. The value of the k parameter was set to one ($k = 1$) for K-NN and the second degree polynomial function was selected as the kernel function for SVM since we obtained the best performance by using these conditions.

Table 12 presents the performance dependence on the number of clusters. The execution times and the accuracy rates or K-NN and SVM classifiers are presented. The recognition performances for eight expressions are summarized in Tables 13 to 18 as confusion matrices obtained from the K-NN and SVM classifiers with $G = 2$, $G = 5$ and $G = 7$. Assume

TABLE 12. Execution times and accuracy for $G = 2$, 5 and 7

Number of clusters (G)	Template dictionary generation: Td (hour:minute)	Feature extraction: Tf (hour:minute)	Classification: Tc (Sec.)		Accuracy	
			K-NN	SVM	K-NN	SVM
2	7:46	4:00	0.68	50.469	91.17% +/− 2.36%	66.00% +/− 5.32%
5	7:40	6:34	1.828	144.153	90.33% +/− 1.91%	67.00% +/− 4.00%
7	7:34	7:52	2.5	205.844	90.42% +/− 2.36%	67.83% +/− 4.54%

TABLE 13. SSS feature vector with K-NN ($G = 2$)

		Predicted							
		Happiness	Sadness	Surprise	Fear	Anger	Disgust	Contempt	Neutral
Actual class	Happiness	**93.33**	0.67	0.00	2.00	0.00	0.00	0.67	3.33
	Sadness	1.33	**87.33**	1.33	2.00	1.33	1.33	2.67	2.67
	Surprise	0.67	1.33	**92.00**	4.67	0.00	0.67	0.00	0.67
	Fear	4.00	2.67	1.33	**86.00**	0.00	4.67	0.67	0.67
	Anger	0.00	2.67	4.00	0.00	**88.00**	1.33	1.33	2.67
	Disgust	0.00	0.67	0.00	0.67	1.33	**93.33**	4.00	0.00
	Contempt	1.33	0.67	0.00	0.67	3.33	0.00	**93.33**	0.67
	Neutral	1.33	0.67	0.00	0.67	1.33	0.00	0.00	**96.00**

TABLE 14. SSS feature vector with SVM ($G = 2$)

		Predicted							
		Happiness	Sadness	Surprise	Fear	Anger	Disgust	Contempt	Neutral
Actual class	Happiness	**77.33**	4.67	1.33	2.00	2.67	2.00	0.67	9.33
	Sadness	2.67	**72.00**	3.33	5.33	2.00	2.00	3.33	9.33
	Surprise	4.00	7.33	**77.33**	3.33	2.00	1.33	0.67	4.00
	Fear	13.33	9.33	17.33	**42.67**	2.67	4.00	4.00	6.67
	Anger	2.00	4.00	4.00	2.00	**62.00**	2.00	12.00	12.00
	Disgust	6.67	6.00	4.00	8.67	10.67	**48.67**	10.00	5.33
	Contempt	6.67	2.00	0.00	3.33	3.33	10.67	**63.33**	10.67
	Neutral	4.00	2.67	0.00	2.67	1.33	4.67	0.00	**84.67**

TABLE 15. SSS feature vector with K-NN ($G = 5$)

		Predicted							
		Happiness	Sadness	Surprise	Fear	Anger	Disgust	Contempt	Neutral
Actual class	Happiness	**94.67**	0.67	0.00	1.33	0.00	0.00	0.67	2.67
	Sadness	2.00	**88.67**	2.00	1.33	1.33	1.33	2.00	1.33
	Surprise	1.33	2.67	**88.67**	4.00	0.00	1.33	0.67	1.33
	Fear	5.33	2.67	3.33	**82.67**	0.67	3.33	0.67	1.33
	Anger	0.00	2.67	4.00	0.00	**86.00**	2.67	2.00	2.67
	Disgust	0.00	0.00	0.67	0.67	0.67	**94.67**	3.33	0.00
	Contempt	1.33	1.33	0.00	1.33	2.67	0.67	**91.33**	1.33
	Neutral	1.33	0.67	0.00	0.67	0.67	0.00	0.67	**96.00**

that Td, Tf and Tc denote the execution time for template dictionary generation, feature extraction and classification. In feature extraction phase, the 1,200 frames (total frame number) were sampled from 39,000 frames. When the number of cluster is set to two, five and seven, the Tf became 4 hours, 6 hours 34 minutes and 7 hours 52 minutes, respectively. With $G = 2$, the Tc of K-NN was 0.68 seconds, that Tc of SVM was 50.469 seconds. Each of Tc is fairly smaller than these of Td and Tf. It took fairly long time for Td, but this phase was conducted just once. It is noted that the Tf and Tc for feature extraction and classification would become significant issues for real-time classification. When the number of cluster is set to two, the accuracy rates of K-NN and SVM were 91.1% and 66.0%. The performance did not vary significantly when the cluster number was increased for $G = 5$ and $G = 7$. In the future work, we need to reduce the

TABLE 16. SSS feature vector with SVM ($G = 5$)

		Predicted							
		Happiness	Sadness	Surprise	Fear	Anger	Disgust	Contempt	Neutral
Actual class	Happiness	**78.00**	4.67	0.00	2.67	2.67	4.00	0.00	8.00
	Sadness	2.67	**69.33**	4.00	5.33	3.33	1.33	4.00	10.00
	Surprise	4.67	6.00	**67.33**	10.00	1.33	2.00	3.33	5.33
	Fear	15.33	7.33	9.33	**50.00**	3.33	3.33	4.00	7.33
	Anger	2.00	4.00	0.67	2.00	**63.33**	2.67	14.00	11.33
	Disgust	6.00	7.33	1.33	4.67	12.00	**55.33**	8.67	4.67
	Contempt	6.00	1.33	0.00	2.67	2.67	7.33	**69.33**	10.67
	Neutral	5.33	4.00	0.67	2.67	1.33	2.67	0.00	**83.33**

TABLE 17. SSS feature vector with K-NN ($G = 7$)

		Predicted							
		Happiness	Sadness	Surprise	Fear	Anger	Disgust	Contempt	Neutral
Actual class	Happiness	**93.33**	0.67	0.67	2.00	0.00	0.00	0.67	2.67
	Sadness	1.33	**89.33**	1.33	1.33	1.33	0.67	2.67	2.00
	Surprise	1.33	1.33	**88.67**	5.33	0.00	0.67	1.33	1.33
	Fear	4.00	2.67	4.00	**82.67**	1.33	2.67	1.33	1.33
	Anger	0.00	2.67	2.00	0.00	**87.33**	2.67	2.00	3.33
	Disgust	0.00	0.00	0.67	1.33	0.67	**94.00**	3.33	0.00
	Contempt	2.00	0.67	0.00	0.00	2.67	0.67	**92.00**	2.00
	Neutral	0.67	0.67	0.67	0.67	0.67	0.00	0.67	**96.00**

TABLE 18. SSS feature vector with SVM ($G = 7$)

		Predicted							
		Happiness	Sadness	Surprise	Fear	Anger	Disgust	Contempt	Neutral
Actual class	Happiness	**76.67**	4.00	0.00	6.67	2.67	4.00	0.00	6.00
	Sadness	3.33	**69.33**	4.00	7.33	2.67	2.67	4.67	6.00
	Surprise	0.67	6.67	**74.67**	5.33	2.00	0.67	1.33	8.67
	Fear	15.33	6.00	18.00	**44.67**	2.00	3.33	4.00	6.67
	Anger	1.33	4.67	4.00	2.00	**63.33**	1.33	13.33	10.00
	Disgust	6.00	8.67	6.00	4.00	8.00	**54.00**	8.67	4.67
	Contempt	6.00	1.33	0.00	2.00	3.33	4.67	**73.33**	9.33
	Neutral	5.33	1.33	0.00	2.67	1.33	2.67	0.00	**86.67**

time required for generating one frame by optimizing the algorithm and re-managing the threads. It took 12 seconds to generate one frame, which is identical to Tf/Nf (where Nf is number of total frames.)

4.4. **Discussion.** In the first experiment, we introduced two kinds of feature vectors. The first one generated from the motion stream was referred to as the stream feature vector. The second one generated from the stream feature was referred to as SSS feature vector. We classified these feature vectors by using the K-NN and SVM methods and discussed advantage and disadvantage of these feature vectors.

The present results clarified that the advantage of SSS feature vector was the accuracy. It attained higher precision than the stream feature vector. The stream feature could cope with only viewpoint and anthropometry variations. The SSS feature vector could

handle all of the variations by utilizing the DTW distance and template dictionary to generate attributes. The DTW distance provides a measure of the similarity between two different sequences even in the different time periods. Thus, execution rate variation would be reduced by using DTW distance. Template dictionary was pre-learned pattern generated from the stream feature. Each sequence in the template dictionary was fine-tuned to facial-part-movement level which helped SSS feature vector to avoid personal style variation.

The present results showed that the disadvantage of SSS feature vector was the longer computation time. The SSS feature vector had more complicated attributes than the stream feature and required more time in calculating one feature vector than stream feature vector. In the future work, we must consider that the recognition system is expected to respond as fast as possible for realizing real-time classification.

In the second experiment, we examined the parameter dependence of classifiers. The k parameter of K-NN was modified from one to ten and several kernel functions were selected in the SVM. We evaluated both accuracy and execution time to derive the best fitting parameter. We obtained the best results of accuracy and speed from K-NN with $k = 1$. In the last experiment, we modified the number of clusters of template dictionary (G). The number of clusters G was assumed to be two, five and seven since it took long time to extract new feature. The number of vertices on facial skeleton was fixed to 18 vertices. Then, the number of streams became 153. We obtained $G \times 153$ template dictionary for $G = 2$, 5 and 7. We employed different sizes of the attributes for the template dictionary dimension of SSS feature vector. The sizes of attributes became $G \times 153$ for $G = 2$, 5 and 7 and increased the execution time significantly. However, the accuracy did not change. We obtained the best performance for $G = 2$ and it can be used for real-time classification in the future work.

5. **Conclusion.** The proposed approach had reduced the effects of intra-class variations in the human facial emotion recognition system in the following points: the viewpoint variation, the anthropometry variation, the execution rate variation and the personal style variation.

The proposed approach could outperform the state-of-the-art approach [16] by reducing the effect of intra-class variations and attained 10% better accuracy. Some constraints in state-of-the-art approach was discussed. For example, five poses were assumed for the dataset. We could assume only one pose for representing all possible poses. Therefore, the size of dataset will be smaller than the dataset used in the state-of-the-art approach.

For the future work, we consider that is necessary to build a bigger database, and real-time application.

REFERENCES

[1] N. Chanthaphan, K. Uchimura, T. Satonaka and T. Makioka, Facial emotion recognition based on facial motion stream generated by Kinect, *Proc. of the 11th International Conference on Signal-Image Technology & Internet-Based Systems*, Bangkok, pp.117-124, 2015.

[2] N. Chanthaphan, K. Uchimura, T. Satonaka and T. Makioka, New feature extraction method for facial emotion recognition by using Kinect, *Proc. of the Korea-Japan Joint Workshop on Frontiers of Computer Vision*, Takayama, pp.200-205, 2016.

[3] T. F. Cootes and C. J. Taylor, Active shape models – Smart snakes, *Proc. of British Machine Vision Conference*, pp.266-275, 1992.

[4] T. F. Cootes, G. J. Edwards, and C. J. Taylor, Active appearance models, *Proc. of the 5th European Conference on Computer Vision*, vol.2, no.1, pp.484-498, 1998.

[5] D. Cristinacce and T. F. Cootes, Feature detection and tracking with constrained local models, *Proc. of British Machine Vision Conference*, vol.3, no.1, pp.929-938, 2006.

[6] V. Blanz and T. Vetter, A morphable model for the synthesis of 3D faces, *Proc. of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, pp.187-194, 1999.

[7] T. Baltrusaitis, P. Robinson and L.-P. Morency, 3D constrained local model for rigid and non-rigid facial tracking, *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp.2610-2617, 2012.

[8] X. Zhao, X. Li, C. Pang, Q. Z. Sheng, S. Wang and M. Ye, Structured streaming skeleton – A new feature for online human gesture recognition, *ACM Trans. Multimedia Comput. Commun. Appl.*, vol.11, no.1, pp.1-18, 2014.

[9] D. L. Baggio, S. Emami, D. M. Escrivá, K. Ievgen, N. Mahmood, J. Saragih and R. Shilkrot, *Mastering OpenCV with Practical Computer Vision Projects*, Packt Publishing Ltd., Birmingham, 2012.

[10] D. F. Dementhon and L. S. Davis, Model-based object pose in 25 lines of code, *International Journal of Computer Vision*, vol.15, no.1, pp.123-141, 1995.

[11] C. L. Lawson, Transforming triangulations, *Discrete Mathematics*, vol.3, no.1, pp.365-372, 1972.

[12] X. Zhu and D. Ramanan, Face detection, pose estimation, and land mark localization in the wild, *Proc. of IEEE Computer Vision and Pattern Recognition*, pp.2879-2886, 2012.

[13] Z. Zhang, Microsoft Kinect sensor and its effect, *IEEE Computer Society*, vol.19, no.2, pp.4-12, 2012.

[14] S. Piana, A. Staglianò, F. Odone, A. Verri and A. Camurri, Real-time automatic emotion recognition from body gestures, *Cornell University Library: Computing Research Repository*, arXiv:1402.5047, 2014.

[15] K.-C. Huang, Y.-H. Kuo and M.-F. Horng, Emotion recognition by a novel triangular facial feature extraction method, *International Journal of Innovative Computing, Information and Control*, vol.8, no.11, pp.7729-7746, 2012.

[16] Q. R. Mao, X. Y. Pan, Y. Z. Zhan and X. J. Shen, Using Kinect for real-time emotion recognition via facial expressions, *Frontiers of Information Technology & Electronic Engineering*, vol.16, no.4, pp.272-282, 2015.

[17] P. Ekman and W. Friesen, Measuring facial movement, *Environmental Psychology and Nonverbal Behavior*, pp.56-75, 1976.

[18] H. Sakoe and S. Chiba, Dynamic programming algorithm optimization for spoken word recognition, *IEEE Trans. Acoustics, Speech and Signal Processing*, vol.26, no.1, pp.43-49, 1978.

[19] Y. Sakurai, C. Faloutsos and M. Yamamuro, Stream monitoring under the time warping distance, *Proc. of IEEE International Conference on Data Engineering*, pp.1046-1055, 2007.

[20] Z. Yang, Y. Zhu and Y. Pu, Parallel image processing based on CUDA, *Computer Science and Software Engineering*, vol.3, no.1, pp.198-201, 2008.

[21] A. Jović, K. Brkić and N. Bogunović, An overview of free software tools for general data mining, *Proc. of Information and Communication Technology, Electronics and Microelectronics*, pp.1112-1117, 2014.

[22] J. Kim, Y. S. Kim and S. Savarese, Comparing image classification methods: K-Nearest-Neighbor and Support-Vector-Machines, *Proc. of the 6th WSEAS International Conference on Computer Engineering and Applications, and the 2012 American conference on Applied Mathematics*, pp.133-138, 2012.

[23] C.-W. Hsu, C.-C. Chang and C.-J. Lin, *A Practical Guide to Support Vector Classification*, http://www.csie. ntu.edu.tw/∼cjlin/papers/guide/guide.pdf, 2010.