Use of projection histograms for tracking faces under out-of-plane rotation

Hanjin Ryu
Myounghoon Kim
Seungwook Cha
Sanghoon Sull
Korea University
Department of Electronics and Computer Engineering
1, 5-ka, Anam-Dong, Sungbuk-Ku
Seoul 136-701, Korea
E-mail: sull@mpeg.korea.ac.kr

Abstract. We present an efficient face tracking method that is robust, especially when the face is turned away from the frontal view to the side view (out-of-plane rotation). The proposed method, consisting of three steps, utilizes the horizontal and vertical projection histograms of a face region to model the visual appearance of the face. The vertical and horizontal positions of a face are sequentially determined. First, the horizontal projection histogram of each potential face region in the current frame near the corresponding face region in the previous frame is used as an input to a back-propagation neural network (BPNN) to reliably estimate the vertical position of the face, based on the observation that the distribution of the horizontal projection histogram of a face region is stable, even when the head is rotating about an axis parallel to the vertical axis of an image plane. Second, the vertical projection histogram of an eye region derived from the estimated vertical position of the face as well as output values from the BPNN is used to estimate the horizontal position of the face. Third, the detected face region is refined by head boundary detection. These three steps are repeatedly applied to track faces in each shot of an input video. Experimental results are provided to demonstrate the efficiency of the proposed method. [DOI: 10.1117/1.2981563]

Subject terms: face tracking; projection histogram; template matching.

Paper 080148RR received Feb. 25, 2008; revised manuscript received Aug. 1, 2008; accepted for publication Aug. 4, 2008; published online Sep. 25, 2008.

1 Introduction

The purpose of face tracking is to follow one or more faces through video sequences. Face tracking has been regarded as an important research area in computer vision technology due to its wide range of applications such as video indexing, biometrics, video surveillance systems, and human-computer interfaces. For example, the positions or number of times that actors appear in a move provide good information for organizing and presenting video content. Therefore, to search for a particular shot in a movie, in which an actor is playing, face detection or tracking is a fundamental step. A wide variety of face detection and tracking methods have been proposed for static images and video sequences.[1–3]

Face detection methods[4–9] can be used to detect faces in each frame, referred to as the frame-based approach, without using temporal information. However, the frame-based approach lacks the ability to efficiently track faces due to the variability of faces in location, scale, and rotation. Therefore, it is desirable to develop a way of exploiting information available from a face detected in the previous frame, such as the position, size, and appearance for efficient and robust face tracking.[10–14]

The face tracking method can be classified as either image-based or feature-based approaches. The image-based approach builds statistical face models to guide the tracking by learning from samples.[15–17] On the other hand, the feature-based approach seeks the evidence of face existence by using single clues of facial features such as points and color.[18–21] We propose an image-based face tracking method.

The image-based method of face tracking is based on the correlations between the face template (model) and the input images. This method provides an accurate location of the face, but is limited to faces that have been predefined or learned from a training set. A straightforward way of overcoming this problem is to use multiple templates for different views, but a much higher computational cost is required to compute correlations between the multiple templates and the input images.

To overcome the problems described, we propose the use of a projection histogram of a face region where the horizontal and vertical projection histograms utilized in this work are the sums of the pixel values projected onto the vertical and horizontal axes, respectively.[22,23]

The motivation for the proposed method is based on the observation that the distribution of the gray pixel-based horizontal projection histogram is still more stable, even under out-of-plane face rotation and different facial expressions. We presented an earlier version of the face detection method based on the project histogram of binary edge pixels in our previous work,[24] but found out that the stability of an edge pixel distribution heavily depended on the choice of a threshold value and the pose change.

The proposed method consists of three steps: 1. estimation of the vertical position of the face; 2. estimation of the horizontal position of the face; and 3. face region refinement. These three steps are repeatedly applied to track

0091-3286/2008/$25.00

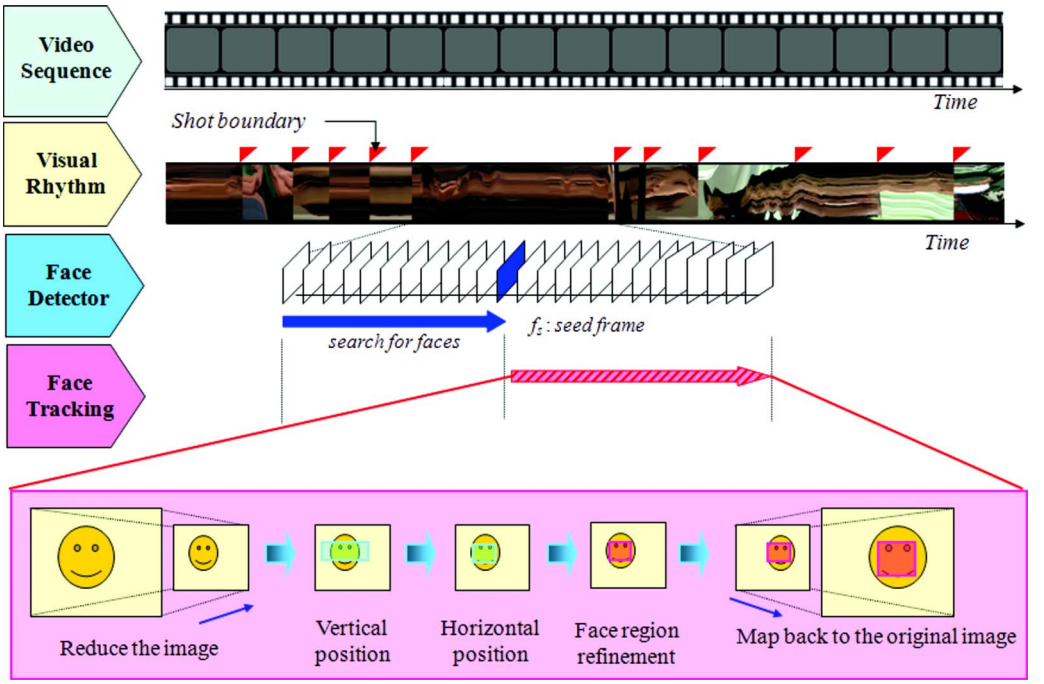


Fig. 1 Basic scheme of the proposed method.

faces in each shot of an input video, while the tracking process starts from the frame in the shot where a face is first detected by using an existing face detection method for a static image.

In the first step, a gray pixel-based horizontal projection histogram of each potential face region in the current frame near the corresponding face region in the previous frame is used to estimate the vertical position of the face. For robust detection, a back propagation neural network (BPNN) is trained to distinguish face and nonface, where an exhaustive scan of the input image is avoided by increasing the size of the moving steps of the scanning window.

In the second step, a vertical projection histogram of an eye region is used to estimate the horizontal position of the face within candidate eye regions obtained from the first step, since the eye region is more reliable than other facial regions for identifying a face pattern. The horizontal position is determined by searching for the eye region in the current frame that is best matched with the vertical projection histogram of the eye region in the previous frame and the output of the BPNN, where the output values of the BPNN from the first step are also used.

In the third step, the face region is refined by comparing the head boundary in the current frame with that in the previous frame, and the scale of the face is also computed, reducing the corresponding part of the next frame in proportion to the face scale change. In this way, the tracking process is adaptively performed for efficient tracking.

The remainder of this work is organized as follows. In Sec. 2, an overview of the proposed method is provided. In Sec. 3, the detailed steps of the proposed method are presented. In Sec. 4, the implementation details, experimental results, and comparison with existing methods are provided to demonstrate the effectiveness of the proposed method. Section 5 concludes the work.

2 Overview of Proposed Approach

The basic scheme of the proposed system is presented in Fig. 1. Tracking is performed on each shot of an input video independently, and the shot transitions are rapidly determined using the visual rhythm method.[25] From the start frame of each detected shot, an existing face detection method such as the neural network-based face detector[5] is repeatedly applied to the frames of each shot until a frame containing one or more detected faces, called a seed frame, is detected. Each detected face is then tracked from the seed frame to the last frame of the shot. The tracking process is performed on an input image whose resolution is reduced by the proportion between the size of the face region in the previous frame and that of the training face pattern with a size of 32×32 pixels. However, we only reduce the image patch corresponding to the search region, which is set to 200% of the size of the face region in the previous frame in both the horizontal (x) and vertical (y) directions, rather than reducing the full input image.

The face tracking process consists of three main steps. First, the horizontal projection histogram is used within the search region with a size of 64×64 pixels in the reduced resolution image to determine the vertical position of the face. Next, the vertical projection histogram matching method is used to determine the horizontal position of the face. Finally, the face region is refined based on face scale estimation and verification, after which the refined face region is mapped back to the original image scale.

3 Proposed Method

The three steps of the proposed method are described in detail. A projection histogram-based face model is used to estimate the face region in the current frame. Then, the face region is refined to cope with scale variation. For each shot,

Fig. 2 The property of the horizontal projection histogram.

these three steps are repeatedly performed to track faces from the seed frame to the last frame of the shot.

3.1 *Projection Histogram*

The projection histogram in the horizontal and vertical directions formed by summing the gray pixel values in the rows and columns, respectively, contains the integral information of the gray pixel distributions.

Let $I^t(i,j)$ be the gray pixel value in a face region with its upper left pixel at the coordinate (i,j) at time t. The face region with a size of $w \times h$ can be modeled by the horizontal projection histogram $HP^t_{(i,j)}(y)$ and the vertical projection histogram $VP^t_{(i,j)}(x)$ as follows:

$$HP^t_{(i,j)}(y) = \sum_{x=0}^{w-1} I^t(i+x, j+y), \tag{1}$$

$$VP^t_{(i,j)}(x) = \sum_{y=0}^{h-1} I^t(i+x, j+y), \tag{2}$$

where $w \times h$ is set to 32×32 in our implementation.

The projection histogram has been widely used in many applications, such as facial feature extraction[22] and head boundary detection,[26,27] since it provides geometric information about the facial features. It has also been used for motion estimation,[23] since the dimensionality reduction involved in the projection histogram yields substantial improvements in the computational efficiency.

3.2 *Estimation of the Vertical Position of the Face*

It is observed that the distribution of the horizontal projection histogram is still stable, even when the face is translated from the face center or is rotated out-of-plane, as can be seen in Fig. 2. This observation provides a clue as to how to effectively determine the vertical position of the face (y_f) under out-of-plane rotation.

The face model based on the horizontal projection histogram is trained with a BPNN. Figure 3 shows the detection result that was obtained within the extended region with a size of 64×64 pixels from the center of the face, for a face region with a size of 32×32 pixels (the scanning window is moved pixel by pixel, and thus, the total number of windows considered for face detection is 1089, which is described in detail in Sec. 5). It was found that a larger number of image patches were detected around the face region, with more horizontally adjacent image patches being detected than vertically adjacent ones. Thus, the BPNN is applied over the search region in larger steps in the horizontal than the vertical directions, and is used to classify each image patch as a face or nonface. The proposed BPNN can be moved in steps of 4 pixels horizontally (m_x) and 2 pixels vertically (m_y) across the search region. After the search region has been explored by the BPNN with the

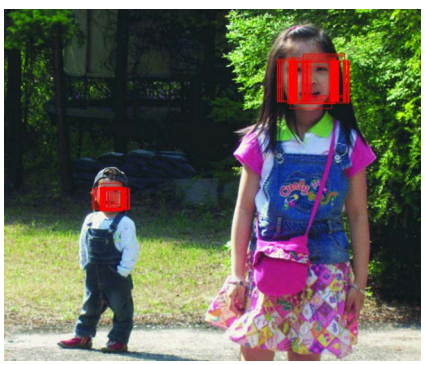


Fig. 3 Results of the designed BPNN [original image resolution 644×544 pixels, given face scale 0.94 (left) and 0.45 (right)].

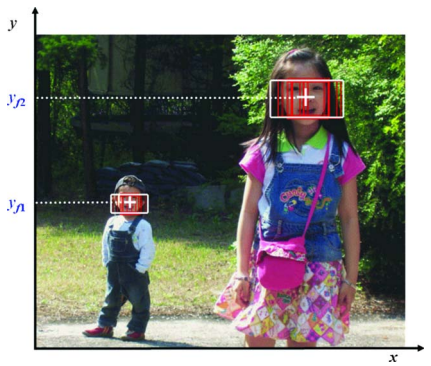


Fig. 4 Example of detection result after the merging process. (The merged regions are represented as white rectangles.)

given moving steps as in Fig. 4, the horizontally neighboring image patches that are classified as a face are merged to a single region (horizontally extended face region) with a size of 64×32 pixels per face. Then, the resulting region gives the estimate of the vertical position of the face (y_f). The implementation details of the training procedure and the moving steps of the BPNN are described in Sec. 4.

3.3 *Estimation of the Horizontal Position of a Face*

The vertical projection histogram of a rectangular eye region is used to estimate the horizontal position of the face (x_f) by performing the vertical projection histogram matching procedure at every location within the candidate eye region with a size of 64×12 pixels derived from the horizontally extended face region.

Let $VP^{t-1}_{(i,j)}(x)$ and $VP^{t}_{(i+p,j)}(x)$ be the vertical projection histogram of the eye region with its upper left pixel at (i,j) in the previous frame, and that of the candidate eye region with its upper left pixel at $(i+p,j)$ in the current frame, respectively. In this work, the sum of the absolute differences is used to measure the degree of matching as follows:

$$\mathrm{SAD}(p) = \sum_{x=0}^{31} \left| VP^{t-1}_{(i,j)}(x) - VP^{t}_{(i+p,j)}(x) \right| \quad (p = 0,1, \ldots ,63). \tag{3}$$

The projection histogram bin-wise difference measure is simpler and faster than the 2-D pixel-wise difference one, but the matching accuracy may be reduced. This is mainly because a 1-D projection histogram alone cannot represent a 2-D image completely. To enhance the matching accuracy in our system, we combine the normalized sum of the absolute differences with the output value of the BPNN as follows: assume M is a max value of the $\mathrm{SAD}(p)$, then the normalized sum of the absolute differences $[n\mathrm{SAD}(p)]$ can be obtained:

$$n\mathrm{SAD}(p) = \begin{cases} \dfrac{\mathrm{SAD}(p)}{M} & (M \neq 0) \\ 1 & (M = 0) \end{cases}. \tag{4}$$

Instead of applying the BPNN again, we use the output value as it is obtained in the detection of the vertical position of the face (y_f). As described in a previous section, we do not apply the BPNN at every location. That is, the BPNN is moved in steps of 4 pixels horizontally across the search region. Thus, the output values of image patches that are not obtained by the BPNN are estimated based on linear interpolation between the given output values, expressed by $f_{HP}(p)$. In this way, the horizontal position of the face region (x_f) is found as follows:

$$x_f = \arg\max_{0 \le p \le 63} \left\{ \frac{1}{2} \times f_{VP}(p) + \frac{1}{2} \times f_{HP}(p) \right\}, \tag{5}$$

$$f_{VP}(p) = 1 - n\mathrm{SAD}(p), \quad (p = 0,1, \ldots ,63). \tag{6}$$

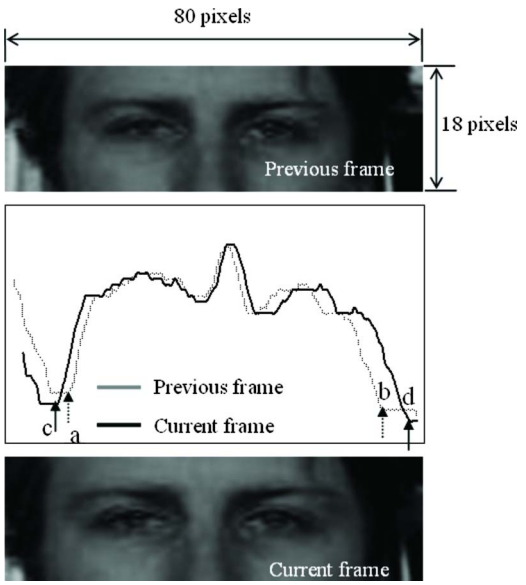


Fig. 5 Head boundary detection.

3.4 *Face Region Refinement*

One obvious drawback of the image-based approach lies in the need for an exhaustive scan of the image at different resolutions to determine the presence of faces. To overcome this problem, we dynamically resize a part of an input frame whose resolution is reduced by the ratio between the size of the face region in the previous frame and that of the training face pattern with a size of 32×32 pixels. For example, assume the size of the face region in the previous frame is 124×124 pixels. Thus, the scale factor (α_{t-1}) to reduce the current input image is 0.26 (=32/124).

The scale factor is estimated by head boundary detection. As shown in Fig. 5, when the face is moved toward the camera, the shapes of the vertical projection histogram of the extended eye region with a size of 80×18 pixels in the previous frame and that in the current frame are similar to each other, but the location of the head boundary in the current frame is translated due to the scale change. Therefore, the left and right boundaries of the head can be determined by finding two points corresponding to the two local minima on both sides of the extended eye region.

Let (a,b) and (c,d) be the locations of the two local minima in the previous and the current frames, respectively. The face scale variation $(\Delta\alpha_t)$ is calculated by the operation $(c-d)/(a-b)$. Thus, the scale factor (α_t) to resize the resolution of the face region in the next frame can be obtained as $\alpha_{t-1} \times \Delta\alpha_t$. Note that the scale factor is estimated by head boundary detection because it is clear and simple to estimate the size of the face. However, it may provide an inaccurate location of the face boundary due to face conditions such as light hair color or side lighting. Therefore, a projection histogram-based difference measure is used to verify the scale variation.

The verification process is performed as follows. First, the current extended eye region with a size of 80×18 pixels is rescaled by the estimated scale factor (α_t). Then, after the eye region with a size of 32×12 pixels in

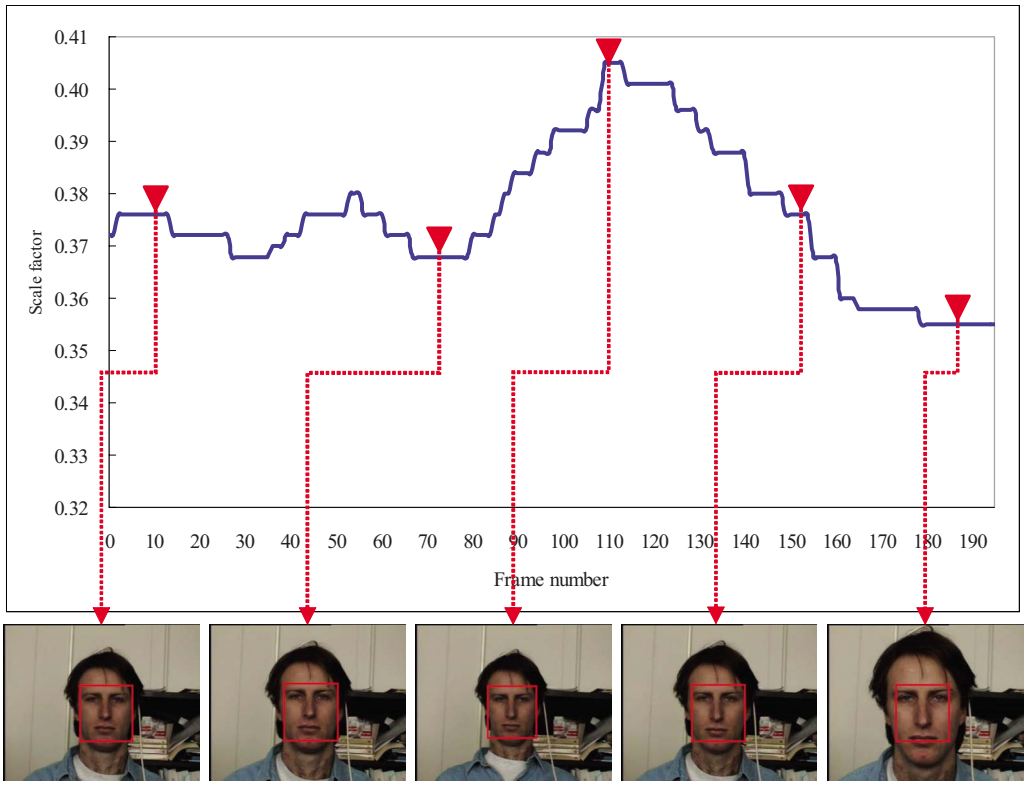


Fig. 6 Scale adaptation result.

the rescaled image is determined, the difference (D_s) between the eye region in the current frame and that in the previous frame is computed as in the following equations.

$$D_s = \frac{1}{2} \times d_h + \frac{1}{2} \times d_v, \tag{7}$$

$$d_h = \sum_{y=0}^{12} \left| \widetilde{HP}^t_{(i,j)}(y) - HP^{t-1}_{(i,j)}(y) \right|,$$

$$d_v = \sum_{x=0}^{31} \left| \widetilde{VP}^t_{(i,j)}(x) - VP^{t-1}_{(i,j)}(x) \right|, \tag{8}$$

where $\widetilde{HP}^t_{(i,j)}(y)[\widetilde{VP}^t_{(i,j)}(x)]$ and $HP^{t-1}_{(i,j)}(y)[VP^{t-1}_{(i,j)}(x)]$ are the normalized horizontal (vertical) projection histogram of the eye region in the current frame and that in the previous frame, respectively.

If the difference (D_s) is smaller than a predefined threshold value (τ), the estimated scale factor is determined as the scale factor to reduce the next frame. Otherwise, the face scale variation is not updated and the scale factor in the previous frame is applied to the next frame. When such a case arises for ten consecutive frames, to avoid accumulating a tracking error, the face detector is applied again to reinitialize the tracking process.

Figure 6 presents the detection result in the case of a scale change. The scale factor is dynamically updated in proportion to the face scale change, allowing the proposed method to overcome the scale variation problem as well as to further improve the tracking performance.

4 Implementation and Experimental Results

In this section, we present the implementation details and the experimental results obtained from the various video sequences listed in Table 1. First, the training of the proposed BPNN and the tracking results of the proposed method are presented. Next, the performance of the proposed method is compared with that of the other methods in terms of the detection rate, speed and accuracy. Finally, limitation of the proposed method is discussed. The applied threshold value (τ) was 0.28, and the experiments were carried out on a Pentium IV 2.56 GHz (1-GB memory) PC platform.

4.1 *Training of the Proposed Back-Propagation Neural Network*

4.1.1 *Data preparation and training*

The 1056 training images of the face pattern were obtained from benchmark face databases such as the Yale,[28] AT&T,[29] BioID,[30] and Stirling datasets.[31] The mirrored version of each image was generated and each image was rotated by two out-of-plane rotation angles (± 10 deg) to produce a total of 4224 examples of faces. The nonface patterns were collected by using an iterative bootstrap technique.[4,5,15] Before training, we used an initial training set of 2080 nonface patterns from background images. After the bootstrap pro-

Table 1 Test sequences.

Test sequence	Total frames	Total number of faces	Resolution	Remarks
Video 1	23,278	17,234	352×240	TV drama
Video 2	15,720	9274	352×240	TV drama
Video 3	26,381	18,765	352×240	TV drama
Video 4	2087	2408	352×240	Home made video
Video 5	200	200	320×240	Face zoom in / out ("jaTz")
Video 6	200	200	320×240	Illumination change ("mll1")
Video 7	200	200	320×240	Pose variation ("vam8")

cess, 15,798 nonface patterns were obtained. The face and nonface samples were manually normalized to 32 ×32 pixels.

To compensate for the gray-level differences resulting from different lighting conditions, an intensity normalization process was performed. First, a face mask was used to eliminate the background pixels around the chin region. Second, we fit a linear model to the intensities of the image, having the following form:

$$f(x,y) = ax + by + cxy + d, \tag{9}$$

where $f(x,y)$ denotes the image and a, b, c, and d are the coefficients to be determined. To solve for the coefficients, we use a least squares approach. Then, the linear fit image is subtracted from the original image to account for lighting differences.

Finally, a histogram equalization process was performed to obtain a new enhanced image with a uniform histogram. In our system, the number of equally spaced bins was 256. The normalized gray values in the projection histogram bins was input to the BPNN used as its input vectors. Table 2 presents a summary of all the parameters of the BPNN. The weights were updated over all the training data (batch learning). The detection rate of the BPNN is 98.4% for the training images and 93.7% for the test images obtained from the World Wide Web and the Caltech database,[32] neither of which were used in the training process. Examples of the training and test patterns are shown in Fig. 7.

Table 2 BPNN parameters.

Parameters	Value
BPNN size	Input layer: 32 units
	Hidden layer: 10 units
	Output layer: 1 unit
Learning rate	0.1
Momentum	0.97
Activation function	Sigmoid
Initial weight	[−1.0, 1.0]
Number of iterations	3000

4.1.2 *Test results with respect to pose variations*

To evaluate the performance with respect to pose variations, we applied the BPNN to the test dataset,[33] which contains 93 head pose images for 15 subjects. As can be seen in Fig. 8, the BPNN detects the face when it is rotated in the range of ±60 deg out-of-plane rotation, ±20 deg in-plane rotation, and ±30 deg up and down nodding.

4.1.3 *Moving steps of the back-propagation neural network*

To determine the size of the moving steps of the BPNN for face tracking, we analyzed the sensitivity of the BPNN with respect to the degree of translation. We collected a set of 50 test images containing a face from the World Wide Web, and cropped image patches with a size of 32×32 pixels around the face location in each image by successively translating the center of the face by an amount varying from −8 to +8 pixels horizontally and from −6 to +6 pixels vertically. We thus obtained a total of 3150 examples (63 image patches ×50 images). One example of the test sets is shown in Fig. 9(a).

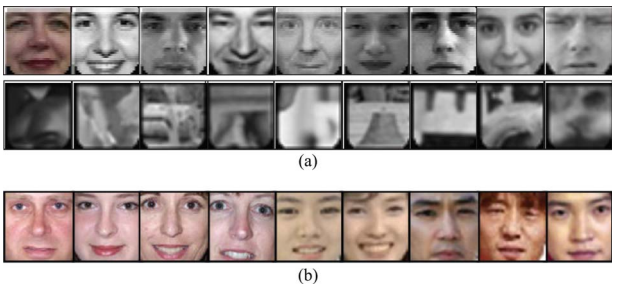
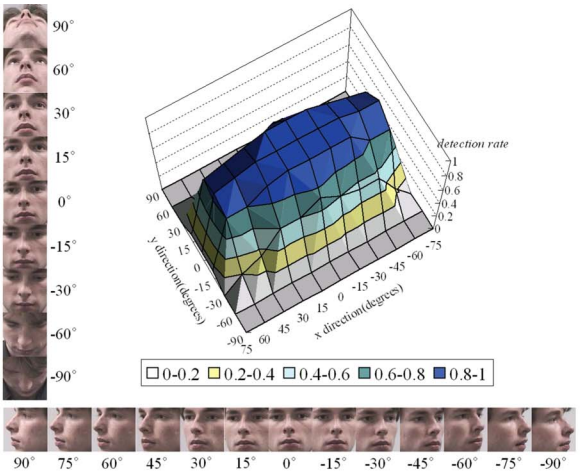
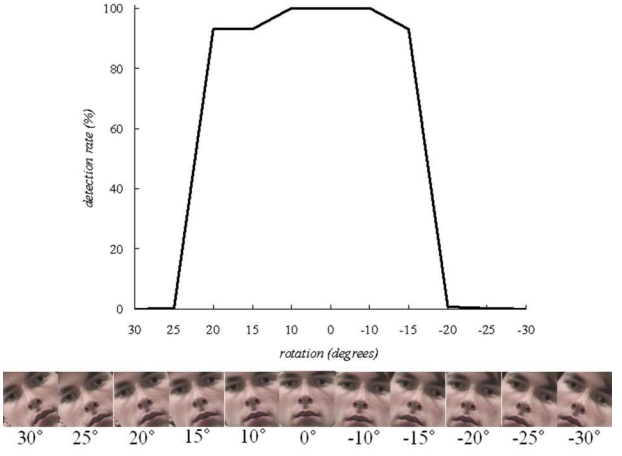


Fig. 7 Examples of training and test patterns. The first row of (a) has positive samples, and the second row are the negative samples for the face category. (a) Training patterns. (b) Test patterns.

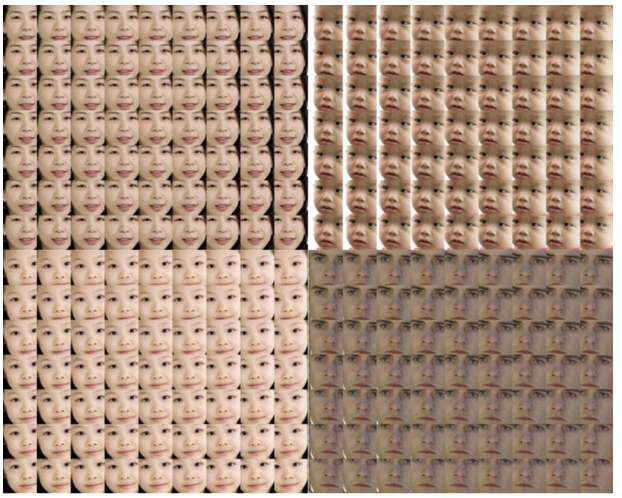


(a)

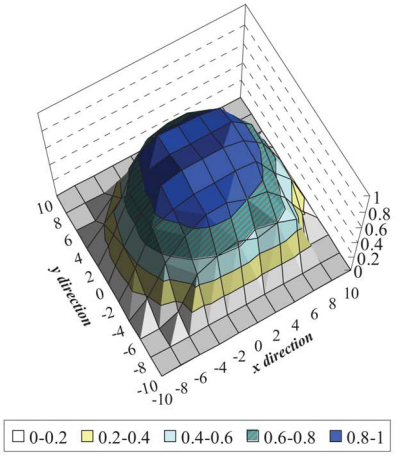


(b)

Fig. 8 Detection rates with respect to pose variations: (a) out-of-plane rotation and (b) in-plane rotation.



(a)



(b)

Fig. 9 Sensitivity analysis with respect to translation. (a) example sets of test images and (b) detection rate over all example sets.

Figure 9(b) presents the detection rate of the BPNN with respect to translation in both the x and y directions when the output value of the BPNN is above 0.8. The detection rate was over 90% when the images were translated within 4 pixels in the x direction and 2 pixels in the y direction. This result allows us to move the BPNN in the steps of up to 4 pixels horizontally and 2 pixels vertically.

4.1.4 *Computational efficiency*

The computational cost of the proposed method is determined by the total number of search points. Let m_x and m_y be moving steps in the x and y directions, respectively, The number of search points per face with a size of $w \times h$ within the search region with a size of $w_s \times h_s$ required to determine the vertical position of the face (y_f) is $\{(w_s-w)/m_x)+1\}\times\{(h_s-h)/m_y)+1\}$. Thus, the number of search points required to estimate the vertical position of the face (y_f) is $\{(64-32)/4)+1\}\times\{(64-32)/2)+1\}=153$. By adding the number of search points required to estimate the horizontal position of the face (x_f), where the scanning window is moved pixel by pixel $(m_x=m_y=1)$, the total number of search points per face is found to be $153+33=186$. Compared to the exhaustive full search per face $(33\times33=1089)$, the number of search points per face is reduced by about 83%.

4.2 *Tracking Results of the Proposed Method*

To show the robustness of the proposed method, we present the tracking results for various face conditions. Figure 10(a) illustrates the result of the proposed method when the camera is moving toward the face. The scale and position of the face vary gradually and smoothly over time as a result of the face region refinement process. The ability of the proposed method to handle changes in viewpoint is demonstrated in Fig. 10(b). Also, the proposed method can track multiple faces simultaneously, as the tracking process is assigned to each face individually. The robustness of the proposed method is shown in Fig. 10(c), where it is able to keep track of the face when it is partially occluded by an object for a while. In this case, the vertical position of the

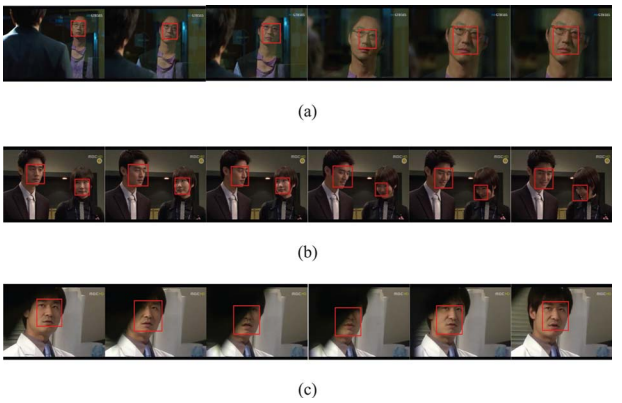


Fig. 10 Examples of tracking result: (a) scale variation (shot 1 in video 1), (b) multiview and multiple faces (video 2), (c) and temporary occlusion (shot 1 in video 3).

face is robustly determined, since the property of the horizontal projection histogram in the right-hand image patches from the face center is still valid. On the other hand, the horizontal position of the face is effectively determined as a result of the adaptive projection histogram matching based on face region refinement.

4.3 *Performance Comparison with Other Methods*

We compared the performance of the proposed method with that of two other methods: the cascade-based[15] and template matching-based methods.[11] The cascade-based method is one of the most well-known image-based methods and is widely used in many applications. This method is based on the idea of a boosted cascade of weak classifiers, and exploits the local contrast configurations of the luminance channel to detect the image regions with human faces.

The template matching-based method uses a 2-D face region as its template and is performed on an input image. An initial template is generated from the sample images and the template is updated at every frame by adding the detected face region to the template. For this experiment, the tracking process of the method is initialized using our employed face detector, and then the faces are tracked in each shot. To cope with scale variations, the template matching procedure is performed on multiple resolutions of an input image.

Table 3 shows the comparison results when only face regions that enclose the eyes and upper lip are considered. In terms of the detection rate (what percentage of the detected faces are correct), the proposed method shows improvements of up to 26.6% and 17.1% for video 4, compared to the cascade-based and template matching-based methods, respectively. This is mainly because the proposed method tracks faces in the range of about ± 60 deg out-of-plane rotation. Furthermore, the proposed method is less sensitive to slight variations in in-plane rotation and up-down movement.

The cascade-based method did not consistently detect the faces, since it is only able to recognize those types of faces that are used in the training sets. That is, this method is limited to upright frontal faces. Thus, a straightforward method to overcome the pose problem is to build multiple classifiers of perspective views, but a much higher computational cost is required.

The template matching-based method also provided accurate locations in the case of upright frontal faces. However, it was also sensitive to pose and scale variations that reduce the tacking accuracy. Although the face template was updated by adding the detected face region to the face template in the previous frame, the face template could not adapt to change in pose and scale.

On the other hand, the proposed method consists of three steps in which the x and y locations of the face are sequentially determined. Particularly, the learning-based face model and the combined use of vertical histogram matching and the output from the BPNN give higher accuracy of the proposed method. Thus, the proposed method can reliably track faces not only in the case of frontal view faces, but also faces looking to the left and downward. Tracking examples of these two methods are shown in Fig. 11.

In terms of the tracking speed, the proposed tracking method required 0.034 sec per frame, which is about 27.7 and 33.3% faster than the cascade-based and template matching-based methods, respectively. This is mainly because the proposed method avoids an exhaustive search and makes use of projection histogram matching.

Table 3 Performance evaluation in terms of detection accuracy.

	Proposed method		Cascade-based		Template matching	
Test sequence (total number of faces)	Detection rate	Number of false alarms	Detection rate	Number of false alarms	Detection rate	Number of false alarms
Video 1(17,234)	84.3%	2711	64.2%	6170	71.1%	4981
Video 2(9274)	85.7%	1326	60.7%	3644	72.3%	2569
Video 3(18,765)	85.4%	2740	64.4%	6680	70.8%	5479
Video 4(2408)	92.5%	181	65.9%	821	75.4%	592

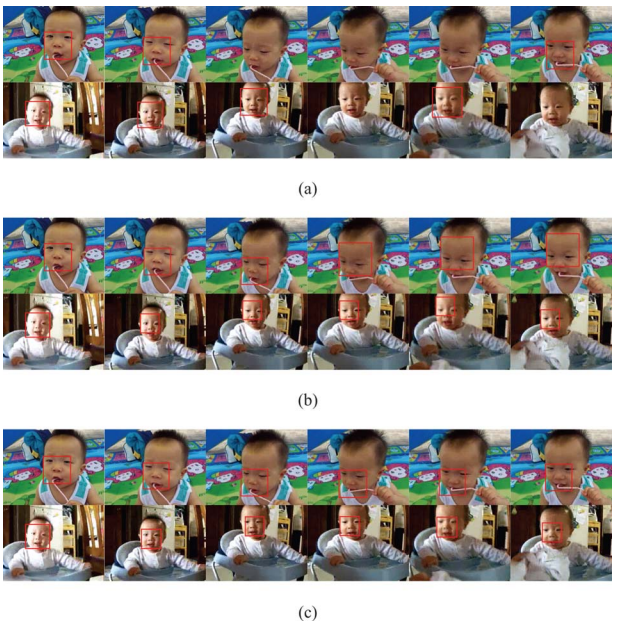


Fig. 11 Tracking examples of the three methods (video 4): (a) cascade-based, (b) template matching-based, and (c) proposed methods.

To evaluate the tracking accuracy, we compared the location of the face determined by the proposed method with the manually identified ground truth. For the purpose of comparison, we used three video sequences (videos 5, 6, and 7)[34] with face zoom in/out, and changes in illumination and pose. The graphs in Fig. 12 compare the tracking results obtained with the ground truth and show example images of the tracking results. The location error of the proposed method is about 2.2 pixels. We observed that the tracking trajectories of the other methods basically followed the same trend.

4.4 *Limitations*

The proposed method cannot identify a new face that appears or whose profile changes to a frontal viewpoint during tracking. One example is shown in Fig. 13(a). One method of overcoming this problem would be to apply the face detector at regular intervals in the frames in each shot.

In addition, the proposed method is initialized by finding faces in each shot, a process that depends on the face detector that is employed. As can be seen in Fig. 13(b), the face detector may identify a background image similar to a face pattern as a face. In this way, the proposed tracking process depends on the face detector. Furthermore, beard and moustache effects have an influence on the distribution of the projection histogram. If the seed frame is detected by the face detector, the proposed method has the potential to speed up the tracking process. Otherwise, the weakness of the projection histogram must be overcome to improve the performance. Thus, an additional measure to further validate the face region or a more robust face detector is required to improve the performance of the proposed method. Future work will focus on this issue.

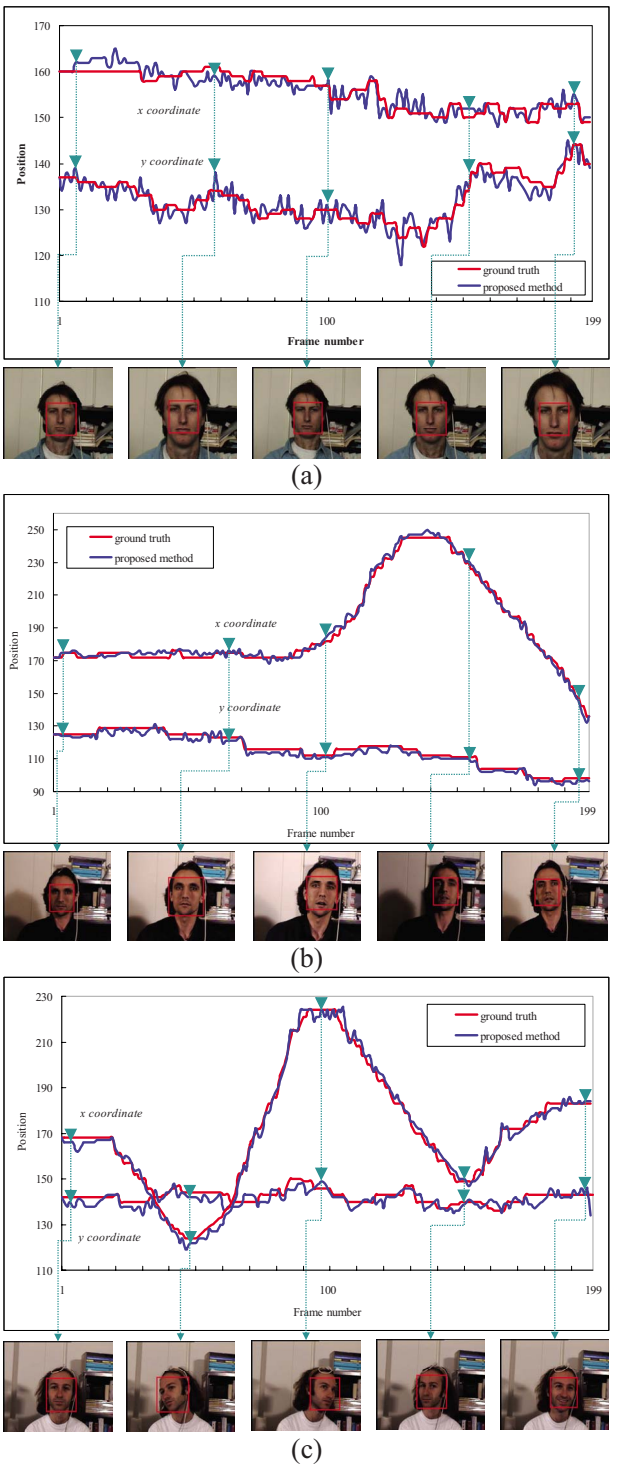


Fig. 12 Tracking accuracy compared with the ground truth: (a) face zoom in/out (video 5), (b) illumination changes (video 6), (c) pose variation (video 7).

5 Conclusions

We develop a face model based on a projection histogram for face tracking. By taking advantage of the projection histogram, the proposed method not only provides an accurate location of the face even under out-of-plane rotation, but also reduces the computational cost. The learning-based

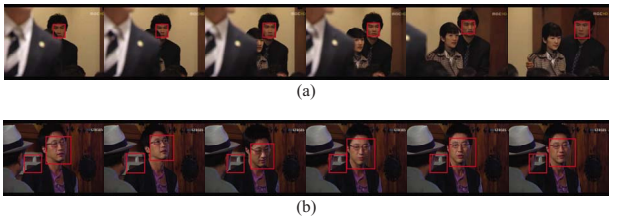


Fig. 13 Limitations of the proposed method: (a) shot 2 in video 3 and (b) shot 2 in video 1.

face model does not require training samples consisting of different views to cope with pose variations. Moreover, the combined use of vertical histogram matching and the output from the BPNN give higher accuracy of the proposed method. Furthermore, considerable computational efficiency is achieved by reducing the number of search points as compared to the exhaustive search. To make the proposed method adaptable to scale changes, we suggest the use of a reduced image whose resolution is reduced in proportion to the face scale. This process requires neither multiple templates of different sizes, nor multiple resolutions of an input image. Therefore, the proposed method further reduces the computational cost while providing reasonable results. A large number of experimental results are obtained that show that the proposed method enables the tracking to be stabilized in real time and provides improvements in both the detection rate and accuracy, compared to previous approaches. Although our results are encouraging, there are some limitations that need to be overcome to improve the performance. We will continue to integrate other features to improve both the accuracy and speed.

Acknowledgments

We thank the anonymous reviewers whose comments and corrections significantly improved the quality of the work.

References

1. E. Hjelmas and B. K. Low, "Face detection: a survey," *Comput. Vis. Image Underst.* **83**(3), 236–274 (2001).
2. M. H. Yang, D. J. Kriegman, and N. Ahuja, "Detecting faces in images: a survey," *IEEE Trans. Pattern Anal. Mach. Intell.* **24**(1), 34–58 (2002).
3. J. J. Wang and S. Singh, "Video analysis of human dynamics-a survey," *Real-Time Imag.* **9**(5), 321–346 (2003).
4. K.-K. Sung, and T. Poggio, "Example based learning for view based human face detection," *IEEE Trans. Pattern Anal. Mach. Intell.* **20**, 39–51 (1998).
5. H. Rowley, S. Baluja, and T. Kanade, "Neural network based face detection," *IEEE Trans. Pattern Anal. Mach. Intell.* **20**(1), 23–38 (1998).
6. C. Garcia and M. Delakis, "Convolution face finder: a neural architecture for fast and robust face detection," *IEEE Trans. Pattern Anal. Mach. Intell.* **26**(11), 1408–1423 (2004).
7. E. Osuna, R. Freund, and F. Girosi, "Training support vector machines: An application to face detection," in *Proc. Intl. Conf. Computer Vision Patt. Recog. Natural Computation*, pp.130–136 (1997).
8. B. Menser and F. Muller, "Face detection in color images using principal component analysis," in *Proc. 7th Intl. Congress Image Process. Appl.*, pp.13–15 (1999).
9. J. J. de. Dios, "Skin color and feature-based segmentation for face localization," *Opt. Eng.* **46**(3), 037007 (2007).
10. R. Qian, M. Sezan, and K. Matthews, "A robust real-time face tracking algorithm," in *Proc. Intl. Conf. Image Process.*, pp. 131–135 (1998).
11. Z. Liu and Y. Wang, "Major cast detection in video using both speaker and face information," *IEEE Trans. Multimedia* **9**(1), 89–101 (2007).
12. R. S. Feris, T. E. de Campos, and R. M. Cesar Junior, "Detection and tracking of facial features in video sequence," *Lecture Notes Artif. Intell.*, pp. 197–265 (1998).
13. R. C. Verma, C. Schmid, and K. Mikolajczyk, "Face detection and tracking in a video by propagating detection probabilities," *IEEE Trans. Pattern Anal. Mach. Intell.* **25**, 1215–1228 (2003).
14. T. Burghardt and J. Ćalić, "Analysing animal behavior in wildlife videos using face detection and tracking," *IEE Proc. Vision Image Signal Process.* **153**, 305–312 (2006).
15. P. Viola and M. Jones, "Robust real-time face detection," *Int. J. Comput. Vis.* **57**, 137–154 (2004).
16. C. J. Edward, C. J. Taylor, and T. F. Cootes, "Learning to identify and track faces in an image sequence." in *Proc. Intl. Conf. Auto. Face Gesture Recog.*, pp. 265–270 (1998).
17. K. Toyama and A. Blake, "Probabilistic exemplar based tracking in a metric space," in *Proc. Intl. Conf. Computer Vision*, pp. 50–57 (2001).
18. G. Hager and K. Toyama, "X vision: A portable substrate for real time vision applications," *Comput. Vis. Image Underst.* **69**(1), 23–37 (1998).
19. K. Schwerdt and J. Crowley, "Robust face tracking using color," in *Proc. Intl. Conf. Auto. Face Gesture Recog.*, pp. 90–95 (2000).
20. D. Comaniciu, V. Ramesh, and P. Meer, "Real time tracking of non-rigid objects using mean shift," in *Proc. Computer Vision Patt. Recog.*, pp. 142–149 (2000).
21. C. Lerdsudwichal and M. Abdel-Mottaleb, "Algorithm for multiple faces tracking," in *Proc. Multimedia Expo*, pp. 777–780 (2003).
22. T. Lee, S. K. Park, and M. Park, "An effective method for detecting facial features and face in human-robot interaction," *Inf. Sci. (N.Y.)* **176**, 3166–3189 (2006).
23. C. Tu, T. D. Tran, J. L. Prince, and P. Topiwala, "Projection-based block matching motion estimation," *Proc. SPIE* **4115**, 374–384 (2000).
24. H. J. Ryu, S. S. Chun, and S. H. Sull, "Multiple classifiers approach for computational efficiency in multiscale search based face detection," in *Proc. Intl. Conf. Natural Computation*, pp. 483–492 (2006).
25. H. Kim, J. Lee, Y. Yang, S. Sull, W. Kim, and S. M. Song, "Visual rhythm and shot verification," *Multimed. Tools Appl.* **15**, 227–245 (2001).
26. T. Kawaguchi and M. Rizon, "Iris detection using intensity and edge information," *Pattern Recogn.* **36**, 549–562 (2002).
27. J. Song, Z. Chi, and J. Liu, "A robust eye detection method using combined binary edge and intensity information," *Pattern Recogn.* **39**, 1110–1125 (2006).
28. A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Trans. Pattern Anal. Mach. Intell.* **23**, 643–660 (2001).
29. See http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html.
30. See http://www.bioid.com/downloads/facedb.
31. See http://pics.psych.stir.ac.uk.
32. See http://vision.caltech.edu/html-files/archive.html.
33. N. Gourier, D. Hall, and J. L. Crowley, "Estimating face orientation from robust detection of salient facial structures," in *Proc. Pointing 2004, ICPR, Intl. Workshop on Visual Observ. Deictic Gestures* (2004).
34. See http://www.cs.bu.edu/groups/ivc/HeadTracking.

Hanjin Ryu received the BS degree in mechanical engineering from Korea Military Academy, Seoul, Korea, in 1993, and the MS degree in industrial systems and information engineering from the Korea University, Seoul, Korea, in 2002, and his PhD degree in electronics and computer engineering at Korea University, in 2008. His research interests are pattern recognition, image processing, and computer vision.

Myounghoon Kim received the BS and MS degree in computer science from the Kookmin University, Seoul, Korea, in 2003 and 2005, respectively. He is currently working toward the PhD degree in electronics and Computer Engineering at Korea University. His research interests are video indexing for content-based retrieval, image processing, video signal processing, digital broadcasting, and other issues on image and video technologies.

Seungwook Cha received the BS degree in electronic engineering from Korea University, Seoul, Korea, in 2006. He is currently working toward the MS and PhD joint degree in electrical engineering at Korea University. His research interests are image and video signal processing, digital broadcasting, and other issues on image and video technologies.

Sanghoon Sull received the BS degree (with honors) in electronics engineering from the Seoul National University, Korea, in 1981, the MS degree in electrical engineering from the Korea Advanced Institute of Science and Technology in 1983, and the PhD degree in electrical and computer engineering from the University of Illinois, Urbana-Champaign, in 1993. From 1983 to 1986, he was with the Korea Broadcasting Systems, working on the development of the Teletext system. From 1994 to 1996, he conducted research on motion analysis at the NASA Ames Research Center. From 1996 to 1997, he conducted research on video indexing/browsing and was involved in the development of the IBM DB2 Video Extender at the IBM Almaden Research Center. He joined the School of Electrical Engineering at the Korea University as an assistant professor in 1997 and is currently a professor. His current research interests include multimedia data management, including search/browsing, image processing, Internet applications, and digital broadcasting.