Pergamon

Pattern Recognition, Vol. 29, No. 11, pp. 1877–1889, 1996
Copyright © 1996 Pattern Recognition Society. Published by Elsevier Science Ltd
Printed in Great Britain.
0031-3203/96 $15.00+.00

PII: S0031-3203(96)00036-2

AUTOMATIC HUMAN FACE LOCATION IN A COMPLEX BACKGROUND USING MOTION AND COLOR INFORMATION

CHOONG HWAN LEE, JUN SUNG KIM and KYU HO PARK*
Computer Engineering Research Laboratory, Department of Electrical Engineering, Korea Advanced Institute of Science and Technology, 373-1, Kusong-doing, Yusong-gu, Taejon, Korea

(*Received* 22 *June* 1995; *in revised form* 12 *February* 1996; *received for publication* 5 *March* 1996)

Abstract—Automatic human face location in a complex background is one of the difficult problems in developing a fully automatic face recognition system and is still a relatively unexplored problem. In this paper, we propose an effective automatic face location system that can locate the face region in a complex background when the system is used as a pre-processor of a practical face recognition system for security. We use two natural visual cues, the motion and the color, to locate face regions in a real scene. In the first stage of the proposed system, we extract a velocity vector field of the face, and by thresholding this velocity field to detect the region showing a clear motion, the face region is separated from the background. In the next stage, the method based on the thresholding in hue space and knowledge-based location process is used to find the eye, eyebrow, and mouth regions of the face. According to our experiment, the proposed system shows fast and robust location results. Copyright © 1996 Pattern Recognition Society. Published by Elsevier Science Ltd.

Face location Motion detection Color image segmentation Knowledge-based location Face recognition

1. INTRODUCTION

The automatic recognition of human faces is one of the most difficult problems in the area of pattern recognition. This subject has been attracting researchers for many years because the recognition of human face images has very important applications. Unfortunately, the progress of facial image recognition has not been satisfactory; there is still a long way before one can use the methods of facial image recognition for solving practical problems.

So far, several different techniques have been proposed. Among those, one may cite neural nets,[1] template matching,[2,3] Karhunen–Loeve expansion,[4] Gabor wavelet transformation,[5] and isodensity lines.[6] These techniques have produced fair results in the recognition of limited subjects under restricted image acquisition conditions. Most of them, however, were limited to an input image with a simple and uniform background. Thus, these methods are greatly restricted when applied to less restrictive, more realistic situations.

Development of a fully automatic human face recognition system requires two steps. The first step is to detect and find the position of faces in a given unknown picture; it is known as *automatic human face location.* The next step becomes to extract features from the faces detected in the first step and compare them with known database face models; it is known as *automatic human face identification.* To develop a successful human face recognition system, those two steps must be equally emphasized. However, much of the research in the above-mentioned human face recognition methods has focused on the task of identification.[1–6] The restrictions imposed by those systems are the result of this unequal emphasis. Thus, we must pay attention to the automatic face location in order to develop a less restrictive and more practical human face recognition system.

In fact, the automatic human face location itself is a problem of pattern recognition. It includes the automatic segmentation (finding a face region and its eye, eyebrow, nose, and mouth regions) and recognition (distinguishing a human face from other kinds of patterns in the background). Since both the location and the size of a face in a picture are unknown, it is a very difficult problem. Thus the task of automatic face location in a complex background has been relatively unexplored. Most researchers used images with a simple background and many of them limited the number of faces to one. Even works done for more realistic situations used additional *a priori* information such as captions.[7,8] Another work used a mosaic image in a complex black-and-white picture.[9] However, those systems showed a relatively poor performance.

*Author to whom correspondence should be addressed.

This paper focuses on the development of an automatic face location system, which can locate human faces and segment their feature regions in images with a complex natural background, so that the system can be used as a pre-stage of a practical face recognition system. This can be used as a security system for buildings and desk-top workstations with a camera installed. For this, there is no restriction on the number of faces in the image. However, it must be evident which one is the subject face, and the location and size of this face must not be varying critically to be used in the next recognition purpose. To achieve this goal, we propose a face location method that uses the motion and color information of a face. In fact, even with the intuition that these two natural informations can improve the face detection result, these have rarely applied in face location application.

The system proposed in this paper consists of two stages. In the first stage, the face region is separated from the complex background using its motion as a visual cue. We use sequence images of whole face as the input to the system, assuming that the object having the greatest motion in the sequence image is the face to be detected. This assumption comes from the following consideration. When the system is used in an automatic face recognition system, the inputs are generally from a camera. Next, consider the situation that the person to be recognized moves in front of the camera and the images containing his entire face are captured in sequence. In this case, the face in the input sequences will contain motion naturally and it will be the greatest moving object in the input images. In the second stage, we segment the eye, eyebrow, and mouth regions of the face obtained in the first stage. To use the color as a visual cue, we analyse the color information of Korean faces and propose a segmentation algorithm that uses their hue information. By taking the additional two visual informations, the proposed two-stage system can be applied in a general environment. Furthermore, the system, which are presented later in this paper, implements a simple and fast algorithms. Therefore, our face location system is appropriate for real time face recognition application.

The remainder of this paper is organized as follows. We show the proposed algorithm for segmenting faces from a complex background in the next section. We use velocity vector field to detect the face motion from image sequences. This field is used for face extraction. To extract the velocity field, we propose the modified constraint line clustering algorithm, which can generate the field in a very short time. The velocity field is thresholded to detect the face region. Then, by fitting the thresholded region, we obtain an ellipse. Finally, the region inside the ellipse is used as the separated face region from background. The third section shows segmentation of the eye, eyebrow, and mouth regions from the detected face region (i.e. inside the ellipse). To use the color information in order to achieve more robust segmentation, we describe the issues that must be analysed in the color image segmentation. Through this analysis, we propose an efficient algorithm based on the simple thresholding in hue space and the refinement process using *a priori* knowledge of face. The fourth section describes the implementation and the experimental results of the whole face location algorithm.

2. FACE SEGMENTATION FROM A COMPLEX BACKGROUND USING MOTION INFORMATION

The motion of a face is an effective information for separating the face region from the background with the degree of complexity considered in this paper. However, motion detection is one of the vision problems that has attracted many researchers but has not yet been solved completely. Thus we review some prior works briefly and focus on the method appropriate for our purpose of face segmentation. In this section, we also explain the details of the proposed face segmentation algorithm.

2.1. *Motion detection*

The motion of objects and regions in an image is not given directly, but must be computed from more elementary measurements. The initial registration of the light by an eye or an electronic image digitizer can be described as the process of producing a two-dimensional array of intensity values $I(x, y, t)$, which are dependent on the space and time. Motion in an image can be described in terms of a vector field $V(x, y, t)$, which gives the direction and the speed of movement of a point with image coordinated (x, y) at time t. While $I(x, y, t)$ is given directly by the initial measurement, $V(x, y, t)$ is not. The first problem of motion detection is, therefore, the computation of $V(x, y, t)$ from $I(x, y, t)$. This computation is the measurement of visual motion.

There exist generally two schemes for measuring the visual motion according to the type of motions. In the case of long-range motions, it is possible to first identify features such as edges, lines, blobs, or regions and then to measure the motion by matching them over time and detecting their changing positions. These kinds of systems[10-11] can be categorized into one scheme. Estimating the image flow (optical flow) velocity field from the image input is the other scheme. The image flow is the velocity field in the image plane that arises due to the projection of short-range moving patterns in the scene onto the image plane. When considering the assumption made in the previous section, the motion of a face is a short range process. Discontinuities in the image flow by the motion of a face in the scene can assist in segmenting the face region from the complex background regions. Therefore, we propose an algorithm which uses the image flow velocity field to measure the motion of face.

The image flow is modeled by the image flow constraint equation,[12]

$$I_x V_x + I_y V_y + I_t = 0. \tag{1}$$

This equation models the interaction between the velocity field (V_x, V_y) and the local changes in the space and time of the image intensity $I(x, y, t)$. I_x, I_y, and I_t are the partial derivatives of the image intensity with respect to x, y and t. The image flow, however, cannot be computed at a point in the image independently of the neighboring points without introducing additional constraints. It is because the velocity field at each image point has two components while the change in the image intensity at a point in the image plane due to motion yields only one constraint. Consider, for example, a patch of a pattern whose intensity varies as a function of one image coordinate but not the other. The movement of the pattern in one direction alters the intensity at a particular point, but the motion in the other direction yields no change. Thus the components of movement in the latter direction cannot be determined locally.

So far, several different algorithms have been developed to estimate the image flow velocity field by introducing additional constraints. The clustering technique by Hough transformation,[13] image flow estimation by regularization,[12] and correlation[14] have been proposed, but none of them are suitable for real world scenes such as those inputs used in this paper, because they cannot handle velocity field discontinuities. Furthermore, these algorithms have a large computational intractability to be used as an early vision process in the face recognition application. Schunck[15] proposed the constraint line clustering algorithm for estimating the image flow velocity when there are discontinuities in motion boundaries. He used the polar form of the image flow constraint equation and 1-dimensional clustering of the intersections of the constraint lines within a neighborhood to determine the most consistent subset of the intersections that cluster about the likely velocity. The polar form of the image flow constraint equation was used for image flow estimation with motion boundaries since the polar form will not contain δ-functions at step discontinuities. This induces several trigonometric transformations. The computational burden still remains although it is reduced than that of the previous algorithms. However, the ideal step discontinuities do not occur in a real situation.

When considering our face segmentation from a complex background, the image flow algorithm must be able to estimate the velocity field in images that contain motion boundaries with reasonable computation to be used as the first step of the face recognition. We propose the image flow estimation algorithm based on the constraint line clustering algorithm to satisfy the above conditions. The algorithm in this paper differs from the constraint line clustering algorithm in that there is no trigonometric computation due to the use of polar form. Thus, we call it modified constraint line clustering algorithm.

2.2. *Modified line clustering algorithm*

The image flow constraint equation (1) defines a line L in the velocity space as shown in Fig. 1(a). This line is called the constraint line and is uniquely defined in the position (xi, yj) of an image. This line, however, is the locus of the possible velocities by (1) and does not specify the velocity in that pixel. Therefore, we introduce additional constraint lines for the neighborhood of the (xi, yj) position. The additional constraint line, L', then defines an intersection V_1 on the constraint line L as shown in Fig. 1(a). The constraint lines are given by

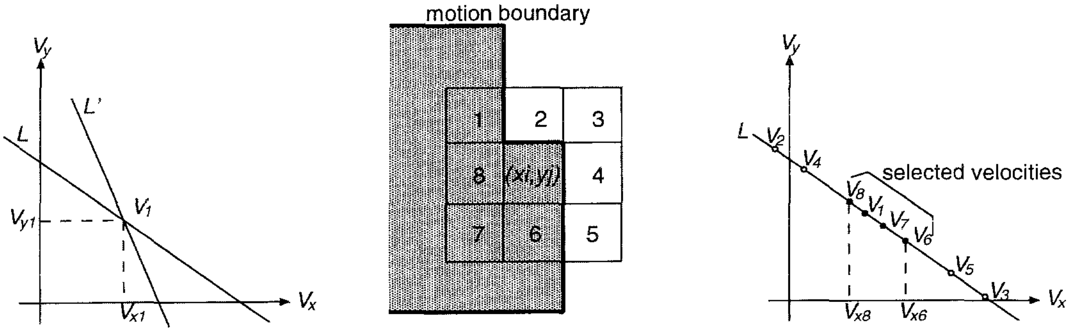
$$L\colon\ I_{xi}V_x + I_{yj}V_y + I_{(xi,yj,t)} = 0, \tag{2}$$

$$L'\colon\ I_{xi-1}V_x + I_{yj-1}V_y + I_{(xi-1,yj-1,t)} = 0. \tag{3}$$

Solving these equations yields

$$V_{x1} = \frac{I_{yj}I_{(xi-1,yj-1,t)} - I_{yj-1}I_{(xi,yj,t)}}{I_{xi}I_{yj-1} - I_{xi-1}I_{yj}}, \tag{4}$$

$$V_{y1} = \frac{I_{xi-1}I_{(xi,yj,t)} - I_{xi}I_{(xi-1,yj-1,t)}}{I_{xi}I_{yj-1} - I_{xi-1}I_{yj}}. \tag{5}$$



(a) (b) (c)

Fig. 1. (a) Intersection (V_{x1}, V_{y1}) of two constraint lines is shown. (b) Window ($N = 3$) of a pixel (xi, yj) in the motion boundary is shown. The shaded region means the same region of motion. (c) a set $\{V_i\}$ of intersections of the constraint lines within a neighborhood with the constraint line L at the center of the neighborhood (xi, yj) in (b) is shown. The index i of V_i correspond to the index of window in (b). The modified constraint line clustering algorithm selects the dotted intersections as the candidate velocities of motion.

Following this way, the intersections of the neighborhood pixels inside the window ($N \times N$) at (xi, yj) shown in Fig. 1(b) lie along the constraint line L. Any intersections that are part of the same region of motion will tend to intersect the line defined by (xi, yj) in a tight cluster around the true velocity, as shown in Fig. 1(c). The set of intersections $\{V_i\}$ must be analysed to determine the most consistent subset of intersections that cluster about the likely velocity. Then the problem is transformed into a cluster analysis problem in one dimension. In one dimension, the cluster analysis is quite easy since there is a total ordering of points along an axis-for example, V_x axis in Fig. 1(c) and the cluster analysis reduces to an interval analysis. We first sort the set of intersections $\{V_i\}$ and then look for the tightest interval that contains half of the intersections by examining successive pairs of intersections that are $\lfloor N/2 \rfloor$ intersections apart, where N is the size of neighborhood window. By doing this, the algorithm selects the pair of intersections that are closest together as the estimate of the majority cluster of intersections. Following procedure summarizes the algorithm.

```
Procedure: Modified constraint line clustering
Input: two consecutive intensity images, $I_1$ and $I_2$, of
       input sequences
Output: a 2-dimensional array in which a pixel has the
        velocity magnitude value $V(xi, yj)$ in its position
Begin
  1: Find $I_x$, $I_y$, and $I_t$ using $I_1$ and $I_2$ in each pixel
     position.
  2: For all pixels, compute
    2.1: For $N^2 - 1$ neighborhood pixels of $(xi, yj)$,
      2.1.1: compute the intersections $\{V_i\}$ defined
             by the constraint lines on the constraint
             line at $(xi, yj)$ using equation (4) and (5).
      2.1.2: Sort $\{V_i\}$ in order of $\{V_{xi}\}$.
    2.2: For $\lfloor N^2/2 \rfloor$ pixels from the $V_i$ having the
         $\min\{V_{xi}\}$, search $V_i$ which has minimum interval
         between the pair, $V_{xi}$ and $V_{xi+\lfloor N/2 \rfloor}$.
    2.3: Select $\lfloor N^2/2 \rfloor$ pixels from the $V_i$ found in step
         2.2 as the selected estimated velocities.
    2.4: compute the velocities $\{|V_i|\}$ of the pixels
         selected in step 2.3.
    2.5: compute the estimated velocity $|V(xi, yj)|$ as
         the average of $\{|V_i|\}$ obtained in step 2.4.
  3: Return $V(xi, yj)$ array.
End
```

The constraint line clustering algorithm is very robust; it works even in the case of motion boundary corners. The performance and error analysis of the constraint line clustering algorithm is well provided by Schunck.[15] Hence, in this paper, we show the performance of the modified constraint line clustering algorithm by comparing the error rate and run time with the constraint line clustering algorithm. We use 256 × 256 synthetic image sequences which have a moving sphere with 1 pixel/frame velocity for measuring errors listed in Table 1. The Yosemite sequence[16] is also used for comparing the computation time. Table 1 shows the estimated velocity and the error rate of two algorithms according to the noise level. The modified constraint line clustering algorithm shows a better estimation performance. This phenomenon comes from the fact that the line clustering algorithm does suffer from errors induced by an coordinate transformation and trigonometric computation. This fact is also the reason for the speedup shown in Table 2. The use of the effective 1-dimensional clustering of the modified constraint line clustering algorithm is another reason for the speedup. As the result, we conclude that the modified constraint line clustering algorithm is an effective method to estimate the velocity field for our face segmentation problem as

Table 1. Error analysis of clustering algorithm

Noise level (%)	Constraint line clustering		Modified constraint line clustering	
	Estimated velocity	Error (%)	Estimated velocity	Error (%)
0	0.68	32	1.00	0
2	0.54	46	1.00	0
4	0.61	39	1.06	6
8	0.84	16	0.78	22
16	1.41	41	1.06	6

The test image is the synthetic image which have 1 pixel/frame velocity. Uniform noise at various amplitudes are added. Noise level 4% means that the variation of the noise amplitudes is in the range [−5.12, 5.12] when the maximum pixel amplitude is 256.

Table 2. Comparison of the computation time (Sun Sparc user time)

Image	Size (pixels)	Constraint line clustering (s)	Modified constraint line clustering (s)	Speed up
Synthetic	256 × 256	3.91	2.17	1.80
Yosemite	316 × 252	24.90	17.86	1.40

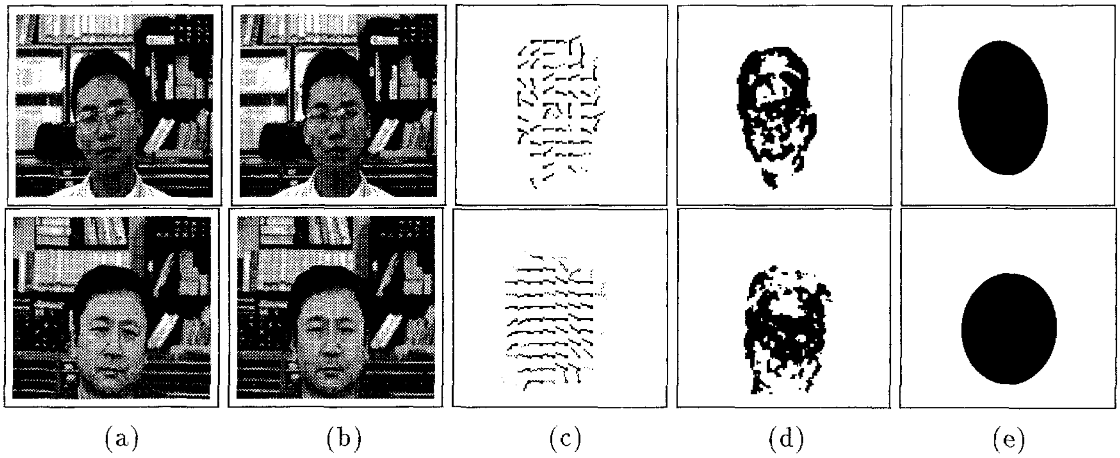


Fig. 2. (a), (b) Two consecutive intensity images of input sequences, (c) the velocity field after applying the modified line clustering algorithm to the (a) and (b) images, (d) thresholding result of the step 5 in algorithm *face segmentation*, (e) final segmented face region of ellipse.

it has more robustness and reduced computation time than the constraint line clustering.

2.3. *Proposed face segmentation from complex background*

We present the proposed face segmentation algorithm. Since we already have the velocity field of input image by the modified line clustering algorithm which show clear discontinuities in the region of face motion, we can segment that region by simply taking the positions that have the velocity value between the maximum histogram velocity and a certain velocity level. By applying this method, we get the face region showing the major short range motion and separated from the complex background of input. The resultant face region, however, does not define exact face region because of the nonuniformity of face motion. Thus we fit this region into an ellipse and consider it as the final segmented face region. In the algorithm*face segmentation*, $H_v(k)$ is the number of pixels in an input velocity array whose amplitude is quantized to the k velocity value, V_{max} is the velocity of $\max_{k\in\{V(xi,yj)\}} H_v(k)$, and V_{th} is the threshold interval of the velocity determined experimentally.

```
Algorithm: face segmentation
Input: two consecutive intensity images, $I_1$ and $I_2$, of
       input sequences
Output: ellipse region as the segmented face region
        from a complex background
Begin
  1: Do spatial filtering of $I_1$ and $I_2$.
  2: Downsample the spatial filtered images.
  3: Procedure Modified line clustering
       Input: results of step 1
       Output: arrays of velocity field, $V(xi, yj)$.
  4: Calculate velocity histogram, $H_v(k)$, of array of
     $V(xi, yj)$ and find $V_{max}$.
  5: Threshold the array of velocity field by
        $|V_{max} - V(xi, yj)| < V_{th}$.        (6)
  6: Remove the noise region by expanding and deleting
     scattered regions.
  7: Fit the region obtained step 6 to an ellipse.
  8: Return the ellipse region.
End
```

Figure 2 shows the segmented results. In Fig. 2, (a) and (b) are the 512×480 input sequence images and (c) is the velocity field obtained with a 5×5 neighborhood window after downsampling by a factor of 4 in step 2. Even after spatial filtering, the noise of the input image makes the quality of the motion estimation data poor, so we downsample the filtered images to reduce the computation and increase the spatial extent of the constraint line clustering neighborhood in step 2. Before fitting into an ellipse region as shown in (e) in the Fig. 2, the result of thresholding has noise regions as shown in (d). These are due to the errors of the modified constraint line clustering algorithm and the noise included in the original input.

3. EYE, EYEBROW, AND MOUTH REGION LOCATION USING COLOR INFORMATION AND *A PRIORI* KNOWLEDGE OF FACE

In this section we describe how eye, eyebrow, and mouth regions of the segmented face region are located by using color information and *a priori* knowledge of the face.

Since the computer performance has improved in recent years, it has become easier to deal with a large amount of data in the form of color image. Furthermore, since a color image provides more information than a gray-scale image does, it is natural to use a color image as the input of face recognition. The regions in a face image corresponding to the eyes, skin, and

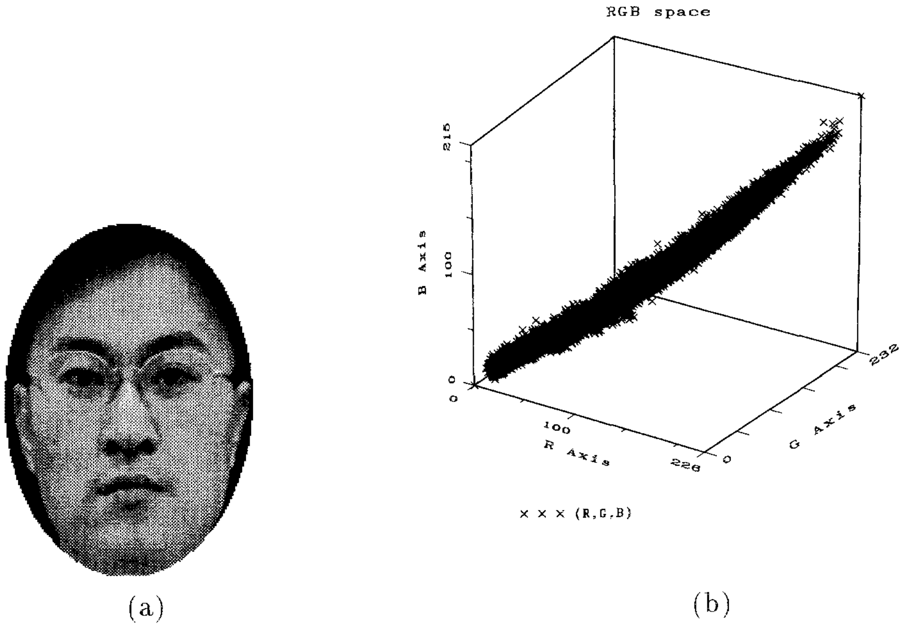


Fig. 3. (a) An ellipse image of typical faces obtained in the Section 2, (b) scattered diagram of (a) in RGB coordinate.

mouth usually exhibit regular properties in a color space, so that we can robustly segment those regions by using those properties. For that purpose, we first analyse the issues regarding the color image segmentation of a face region and then propose an efficient method based on thresholding in hue space and refinement process using *a priori* knowledge of face.

3.1. *Color image segmentation of face*

Segmentation of a color image is the problem of classifying color pixels to a set of clusters showing an uniform color characteristic. Although we focus on the face image, color image segmentation of the face has no difference from the general color image segmentation problem. Thus we must consider the following two points as in the general color image segmentation. First, we must decide on a color space in which the segmentation process is to be performed effectively. Since the shapes and the distribution of the clusters to be segmented depend on the color space, its selection is an important task.[17] Second, we must find a color segmentation algorithm in the selected color space. In fact, many color image segmentation algorithms[17–20] have been proposed, but not all of them work well in the selected color space because of the color data distribution. Thus we must choose an effective method after observing the data in the selected color space.

To find the color space suitable for the segmentation, we examine the color data distribution of a face on four different color coordinate systems (RGB, HSI, L^* u^* v, and the principal component coordinates by Karhunen–Loeve transformation). These four coordinate systems are frequently selected color spaces for testing the performance of many proposed color image segmentation algorithms. We believe that these four coordinate systems are the best candidates that can cover all the possible color spaces for testing the suitability of segmentation.

The tristimuli R(red), G(green) and B(blue) are the usual pixel values of color image processing. Figure 3(a) shows a typical color image of the face obtained from Section 2(though printed in gray scale) and (b) is the corresponding scattered diagram of (a) in the RGB coordinate system. All the data vectors are distributed along the diagonal axis and constitute one cluster. Thus, the regions of the face such as skin, mouth, and eyes do not show clear clusters in this color space.

As one of the color coordinates calculated from the tristimuli, we also examine H(hue), S(saturation), and I(intensity) color coordinate system which is conveniently used for representing the human color perception. In this color space, the data distribution shows well separated clusters as shown in Fig. 4. The mouth is clearly separated from other regions in the hue axis, and the skin, eye, and eyebrow regions are clustered in the hue-intensity plain. Furthermore, the color data corresponding to the skin is heavily concentrated on the lower level of the hue axis and the data for the eye and eyebrow regions are scattered on the mid range of the hue value.

The CIE recommended CIE-L^* u^* v coordinate system is a perceptually uniform color space by a nonlinear transformation of the tristimuli RGB values.[21] This space defines an uniform metric-space representa-

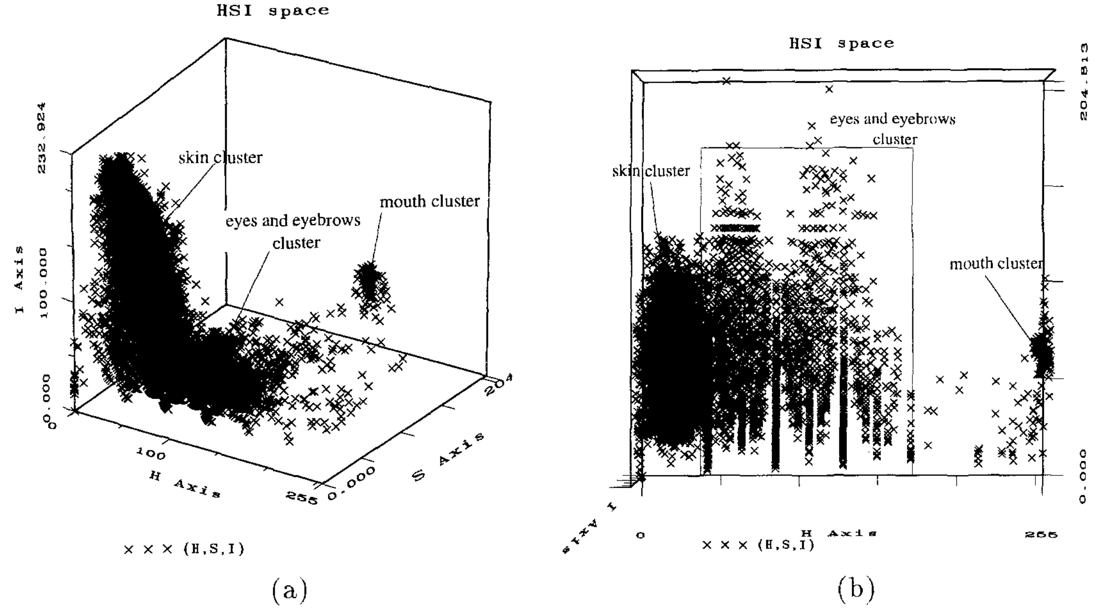


Fig. 4. (a) Scattered diagram of the image in Fig. 3 (a) in HSI coordinate, (b) top view of (a).

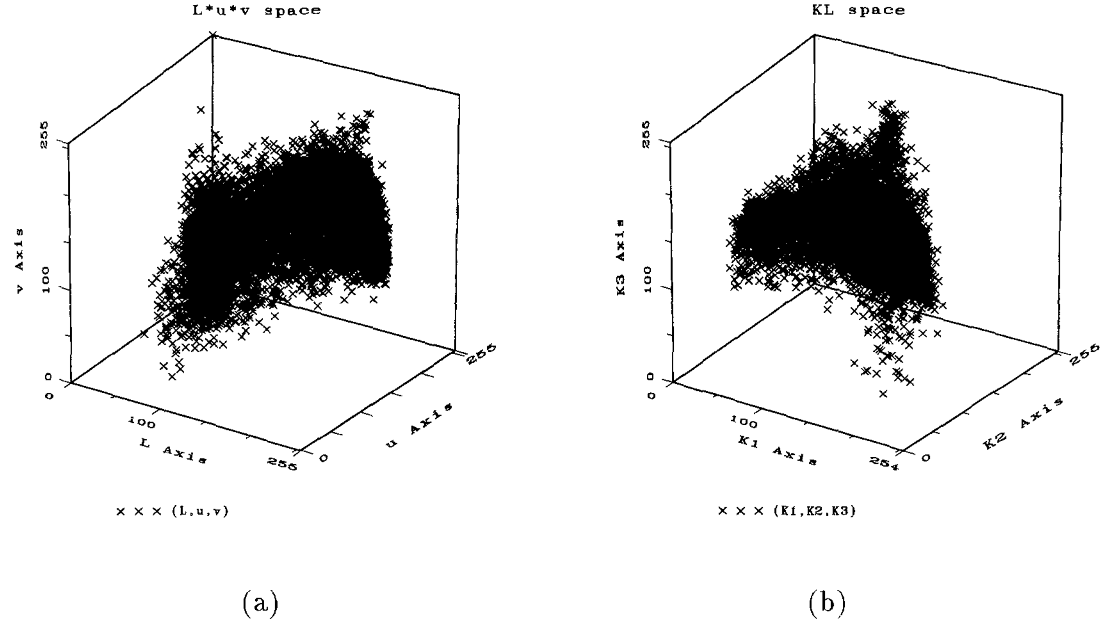


Fig. 5. (a) Scattered diagram of the image in Fig. 3(a) in L^*u^*v coordinate, (b) corresponding scattered diagram in KL space.

tion of color so that a perceptual color difference is represented by the Euclidean distance. It seems that this coordinate is useful for clustering since the three-dimensional color data are distributed in an uniform color space. However, Fig. 5(a) shows that the clusters in the face region do not have a large color distance between them. The entire data is scattered in the center because of ambiguous color differences.

The principal component coordinates (KL space) are obtained from the eigenvalues and eigenvectors of the covariance matrix of the color image values.(21) This process is the well-known Karhunen–Loeve transformation of the RGB tristimulus values. This space is said to have a large discriminant power since the principal coordinates are an orthogonal coordinate system in which the components are uncorrelated. So, many color classification algorithms used this transformed coordinate, and the space like Ohta's,(18) which is based on the KL transformation, was proposed. However, as shown in Fig. 5(b), we can say that the color data of skin, eye, and eyebrow regions does not have a large variance. Therefore, they are not clearly partitioned even in the principal component coordinates.

With the above observations, we can conclude that the HSI color coordinate is the best candidate space for our color image segmentation of face. We tried the multiple histogram-based thresholding schemes[18] and multidimensional clustering algorithms such as K-means (C-means)[19,22] and ISODATA[22] to classify the data in RBG, L^*u^*v, and KL spaces. However, we could not obtain good classification results. Since the data distributions in these three spaces inherently show the ambiguity to be partitioned, it is difficult to detect good clusters. Compared with these three color spaces, HSI space shows a better shape and distribution of the clusters to be divided. The mouth color data shows a clearly separated cluster from other clusters, the skin, eyes and eyebrows data, which can be grouped along the hue axis.

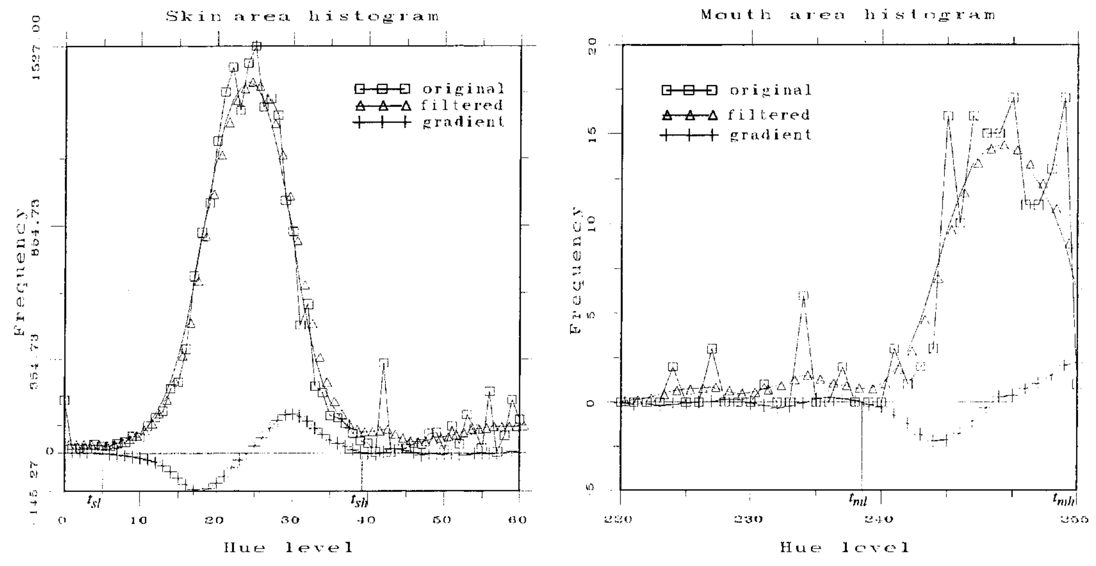
After choosing the HSI space as the suitable space for color image segmentation of face, we can determine an effective segmentation method. Since the clusters are well partitioned, there is no need to apply the multidimensional clustering techniques[19,22] which use a computationally expensive recursive method. We can easily separate the mouth region from other regions through the thresholding scheme in the hue axis. Furthermore, the skin region can be also detected through thresholding since the skin cluster is positioned in the low level of the hue value. Figure 6 shows histograms of those two regions in the hue axis for the thresholding process. The histograms of the skin and mouth regions show clear peaks but contain meaningless local extrema. So to find the threshold values, we must smooth the histograms. For this purpose, Witkin[23] defined the Gaussian convolution of an 1-dimensional signal $f(x)$ as the well-known scale space filter. The convolution is given by:

$$F(x,\tau) = f(x) * g(x,\tau)$$
$$= \int_{-\infty}^{\infty} f(u) \frac{1}{(2\pi)^{1/2}\tau} \exp\left[\frac{-(x-u)^2}{2\tau^2}\right] du \quad (7)$$

where "$*$" implies an 1-dimensional convolution and τ is the Gaussian deviation which must be determined as the optimum scale constant of the $f(x)$, where only the significant peaks remain. Figure 6 also shows the smoothed histogram obtained by applying the scale-space filter. Then we can easily find the threshold values for the peak intervals obtained by computing the first and second derivative of a histogram. Those threshold values are the values satisfying the following conditions.

$$\frac{\partial F}{\partial x} = f * \frac{\partial g}{\partial x} = 0, \quad \frac{\partial^2 F}{\partial x^2} > 0 \quad (8)$$

With the above color segmentation method based on the thresholding in the hue axis, we can find the mouth and skin regions robustly. However, we need additional processing for the segmentation of those regions since the higher levels of eye and eyebrow regions show the ambiguity for selecting threshold values. In general, we can assume that the eye and eyebrow regions are included in the skin region, but as not part of it. These exclusive regions of the skin region are obtained automatically with the skin thresholding process. However, we must select the correct regions



(a)

(b)

Fig. 6. The histograms of skin region, (a) and mouth region, (b) in the hue axis.

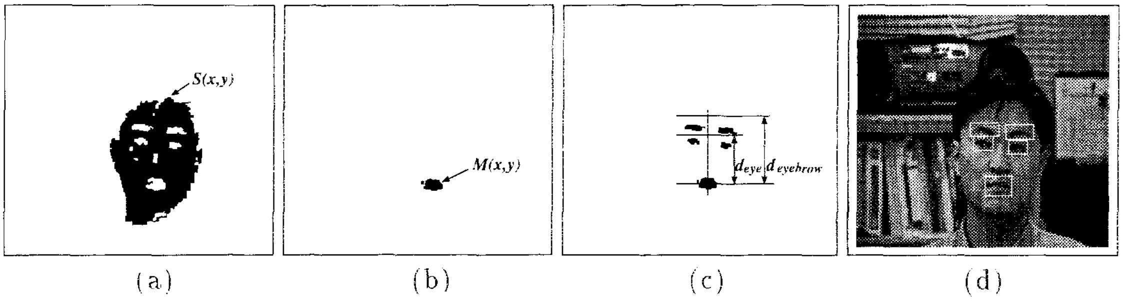


Fig. 7. (a) The thresholded image, T_{skin}, by equation (9), (b) the thresholded image, T_{mouth}, by equation (10), (c) the located regions after the step 4.2 in the algorithm *eyes, eyebrows, and mouth location*, (d) the final located image.

corresponding to the eyes and eyebrows since the thresholding yields more regions than it is expected. In the next section, we describe such selection process using *a priori* knowledge of the facial structure.

3.2. *Proposed eye, eyebrow, and mouth region location algorithm*

We now present the eye, eyebrow, and mouth region location algorithm that we use in this paper. With the histogram analysis of the hue image in step 1 of the algorithm *eye, eyebrow, and mouth region location*, threshold values, t_{sl}, t_{sh}, t_{ml}, and t_{mh}, shown in Fig. 6, are determined. In eyes, eyebrows, and mouth region location, a_{noise} and a_{eye} are the values of the area used to eliminate the noise regions. They are determined from the training set experimentally. In candidate region extraction process, we first locate the mouth region, $M(x,y)$, then find the eyes and eyebrows candidate regions by finding holes in the skin region, $S(x,y)$. In step 4.2, we assume four candidate regions as the left and right eye and the eyebrow regions for the final region location. However, four candidate regions are not always extracted according to the results of the thresholding and the candidate region extraction process, which are found in step 2 and 3. So, to determine the four eyes and eyebrows regions uniquely, we use the rules outlined in step 4.3 ~ 4.5, which use *a priori* knowledge of facial structure. Then we can classify those regions completely in all 16 ($=(\sum_{i=0}^{2}\binom{2}{i})^2$) cases. In the case when only one eye and eyebrow region on one side is extracted, we use the two distances, d_{eye} and $d_{eyebrow}$ shown in Fig. 7(c), to determine that region as either the eye or the eyebrow region.

```
Algorithm eye, eyebrow, and mouth region location
Input: $I_{ellipse}$, hue image of elliptical face region which
       is extracted in the algorithm face segmentation
Output: minimum bounding rectangles of eyes, eye-
        brows, and mouth of $I_{ellipse}$
Begin
  1: Histogram analysis:
    1.1: compute the hue histogram, $H_{hue}$, of $I_{ellipse}$.
    1.2: do the scale space filtering of $H_{hue}$ by equation (7), and compute the first and second
         derivatives of the filtered histogram.
    1.3: find the threshold values, $t_{sl}$, $t_{sh}$, $t_{ml}$, and $t_{mh}$,
         satisfying equation (8).
  2: Compute the two thresholded images, $T_{skin}$ and
     $T_{mouth}$, by
       $T_{skin}(x,y)=\begin{cases}1 & \text{if } t_{sl}<I_{ellipse}(x,y)<t_{sh},\\ 0 & \text{otherwise}\end{cases}$   (9)
       $T_{mouth}(x,y)=\begin{cases}1 & \text{if } t_{ml}<I_{ellipse}(x,y)<t_{mh}\\ 0 & \text{otherwise}\end{cases}.$   (10)
  3: Candidate region extraction:
    3.1: by component labeling of $T_{skin}$ and $T_{mouth}$,
         find the regions with the maximum area,
         $S(x,y)$ and $M(x,y)$, and assume those as the
         skin region and mouth region respectively.
    3.2: by contour following, find a set of boundary
         contours, $\{C_k\}$, inside $S(x,y)$, where
         $k=1,\ldots,$ the number of holes inside $S(x,y)$.
    3.3: for $\{C_k\}$,
      3.3.1: compute the inner area of a contour.
      3.3.2: remove the contours of noise region-
             having smaller area than $a_{noise}$-from the
             $\{C_k\}$.
  4: Region location:
    4.1: for $\{C_k\}$,
      4.1.1: if the center position of a contour is on
             the upper-left side of the center of $M(x,y)$
             region and the area inside the contour is
             greater than $a_{eye}$, then it is classified as the
             set of left eye and eyebrow regions, $\{LE_i\}$.
      4.1.2: if the center position of a contour is on
             the upper-right side of the center of
             $M(x,y)$ region and the area inside the
             contour is greater than $a_{eye}$, then it is
             classified as the set of right eye and eye-
             brow regions, $\{RE_i\}$.
    4.2: find two sets, $\{le_i\}$ and $\{re_i\}$, whose elements
         are the largest areas of $\{LE_i\}$ and $\{RE_i\}$
         respectively.
    4.3: if the number of element in $\{le_i\}$ or $\{re_i\}$ is
         two, then
        4.3.1: if all two sets have two elements each, then the element which is on the upper side of the other element in the same set is classified as the eyebrow region while the other element is classified as the eye region.
        4.3.2: else if only one set has two elements, then
            (a) In the set with two elements, the element which is on the upper side of the other element is classified as the eyebrow region and the other is classified as eye region.
            (b) In the other set having only one element, the element is classified as the eye region if the center of its $y$ position is in the interval of $y$ positions of the eye region determined in the above step (a). Otherwise, it is classified as the eyebrow region.
    4.4: else if the number of elements in $\{le_i\}$ and $\{re_i\}$ is one, then
        4.4.1: if the element of one set is the upper side of that of the other set, then it is classified as the eyebrow region and the other element as the eye region.
        4.4.2: else then
            (a) the element is classified as the eye region if the distance between its center of $y$ position and that of the mouth is smaller than $d_{eye}$.
            (b) the element is classified as the eyebrow region if the distance between its center of $y$ position and that of the mouth is between $d_{eye}$ and $d_{eyebrow}$.
    4.5: else if the set $\{le_i\}$ or $\{re_i\}$ is empty, then
        4.5.1: if all two sets are not empty, then
            (a): the element of the nonempty set is classified as the eye region if the distance between its center of $y$ position and that of the mouth is smaller than $d_{\mathrm{eye}}$.
            (b) the element of the nonempty set is classified as the eyebrow region if the distance between its center of $y$ position and that of the mouth is between $d_{eye}$ and $d_{eyebrow}$.
            (c) cannot detect the eye and eyebrow regions which are on the side of the empty set.
        4.5.2: else then can't detect any eyes and eyebrows regions.
    4.6: draw bounding rectangles of $M(x, y)$, eyes, and eyebrows regions found in the step 4.1 ~ 4.5.
End
```

We first find the mouth region in step 3.1, and then use this region as the reference position to exclude false regions in step 4.1. The distances d_{eye} and $d_{eyebrow}$ also use the center position of the mouth region $M(x, y)$ as the reference. Thus our algorithm does not work when the mouth region is not detected correctly. However, as observed in Section 3.1, the data distribution of the mouth region appears as well-separated cluster compared with the other regions. Thus the mouth region can be robustly detected in our thresholding scheme. Furthermore, even though we use some parameters that are determined in the training image set in the candidate region extraction and region location process, our algorithm can work relatively independent of the input conditions to a certain degree. It is also due to the robustness of the thresholding scheme in the hue space.

By the way, we process only the ellipse region of the face and use 1-dimensional histogram analysis in the above algorithm. Furthermore, we do not need any additional gray scale processing for locating the eyes and eyebrows regions since we only use *a priori* information of the facial structure. Because of these facts, our algorithm is very fast and suitable for real time face location.

4. EXPERIMENTAL RESULTS

This section presents the experimental results obtained using the algorithms described in Sections 2 and 3.

Because input sequence images are obtained in a real environment, they have complex background. Without forcing a person to move the face on purpose, we capture the sequence images which have natural short-range motion. The image scene is illuminated by a single light source or by some combination of several sources each of which generates light with approximately the same spectral power distribution. With few exceptions, most of the input images contain one face. Input images of 512 × 480 resolution are downsampled to 128 × 120 images for velocity field estimation. 256 × 240 hue images are used for region location process. We implemented the whole algorithm on RV860-PIPE[24] system developed at KAIST, which is a four processor (INTEL i860, 40 Mhz) multicomputer system with a color frame grabber. In this system, we can obtain sequence images in real time.

We used a set of 60 images, which were sampled from 15 people, as the training set; this gave us two sequences per person and two images for each sequence. To test the practicality of the proposed system, we gathered a set of 450 sequence images from 45 people. To measure the confidence level of our system, we considered the successful location rate as a random variable (X). Because of various conditions of the subject face, a small number of test sample per subject person does not produce a meaningful performance data. Thus, we measured it after testing 10 sample sequence images per subject face. Table 3 shows the statistics of sample mean (μ_X) and sample standard deviation (σ_X) of successful location rate with its 99% confidence interval[25] which is calculated by assuming

Table 3. Statistics of successful location rate (X) according to the number of subject people; 10 sample sequence images per subject are tested, μ_X means the sample average of X, σ_X means the sample standard deviation of X, Z is the standard normal distribution of X and α, the level of significance, is 0.01

Subject (n)	$\mu_X(\%)$	$\sigma_X(\%)$	$Z_{\alpha/2}\cdot\sigma_X/\sqrt{n}$	99% Confidence interval
10	89.00	11.35	9.27	[79.73 ~ 98.23]
30	86.67	11.64	5.48	[81.18 ~ 92.15]
45	87.56	9.70	3.73	[83.82 ~ 91.29]

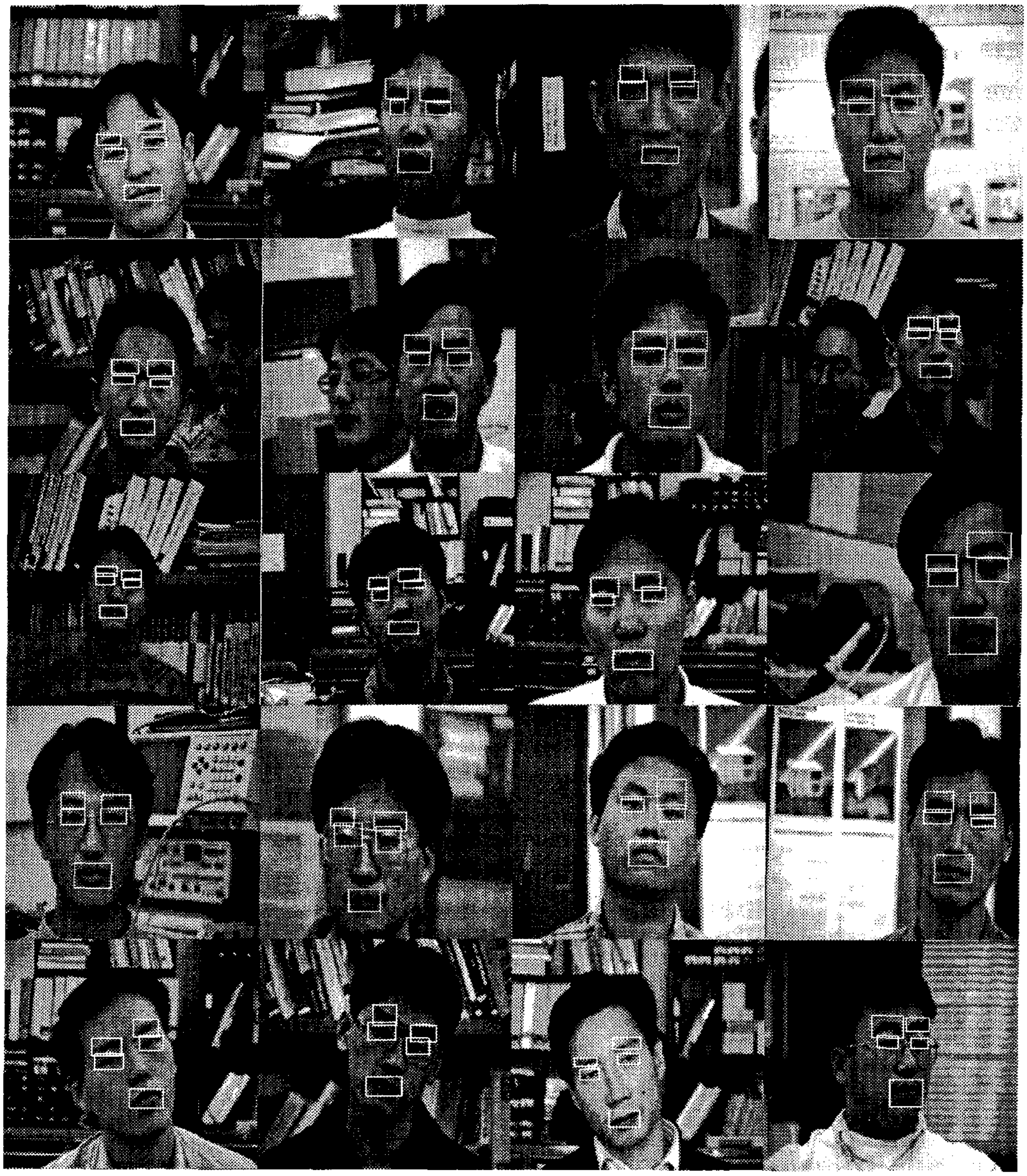


Fig. 8. Examples of the face location results; typical images are in the first row, the images in the second row have multiple faces, the third row is the case of images with faces of various sizes, and the fourth and the fifth rows are the cases of images showing the varying location and rotation of faces.

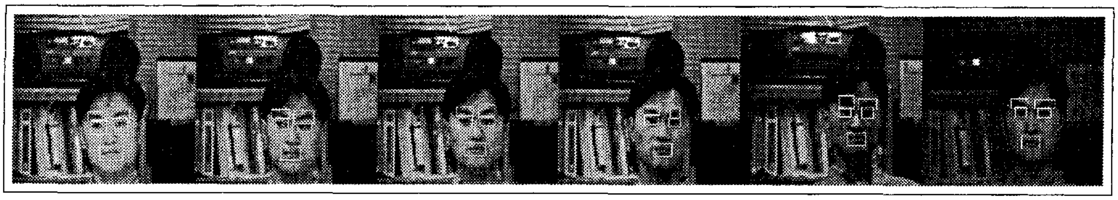


Fig. 9. Face location results in various lighting conditions.

the normal distribution of the X with generality. Our system located all five (eye, eyebrow, and mouth) regions of 45 subject people successfully with average 87.56% rate in the resonable confidence interval. Though not shown in the table, the case of locating more than three out of five regions successfully was 94.75%. Another important fact that can be found in this table is that the average successful location rates is uniform even when the number of subjects is increased. This comes from the robustness of the proposed system and shows that our system can be used stably independent of the subject person. Figure 8 shows examples of the successful location.

In most of the cases, the proposed *face segmentation* algorithm was able to separate the face region from its complex background successfully. The motion of the pixels in the face region showed the average velocity of 4 pixels/s. Even motion with a large variance from the average velocity did not affect the overall performance of the segmentation. Because of the motion caused by noise or objects in the background, background regions around the face were included in the segmented ellipse region. However, it did not produce any severe false candidate regions during the later location process because the proposed location scheme was specifically based on the color image segmentation of the face region. Most of the cases that could not locate the true regions occurred in the step 4.2 of the algorithm, *eyes, eyebrows, and mouth region location.* Since we did not consider explicitly the case of tilted or turned faces in the existing algorithm and the rules used in our algorithm were obtained from the training set, some false regions were induced. Therefore, even though the true region was extracted as a candidate region in the step 4.1 of the algorithm, it was excluded by the selection rule in the step 4.2. However, we can improve this algorithm by adding supplementary rules to deal with these cases. Furthermore, when a region is not located, we can estimate the location of that region by considering additional knowledge-based rules which use the positions of the correctly located regions.

The proposed system showed better results compared with previous systems, which were focused on the task of automatic face location in a complex background. Govindaraju,[7] who used captions as an additional priori information, reported a 73% successful location rate for newspaper photographs. He also reported that on average-two false located face regions per one image were extracted. Yang[9] produced better results than the former method by using the mosaic images in gray scale. He reported that in a test set of 60 images the system successfully located 83% of them. However, false faces appeared in 28 images. By the way, in our experiments, the proposed motion-based face segmentation scheme did not show such false face regions. Furthermore, the color-based region location method showed a relatively small dependency on the lighting conditions. Figure 9 shows the results of the proposed system in various lighting conditions.

In our current implementation, the system can track the located regions by the proposed algorithm in 4 frames/s rate. This means that the run time for two consecutive color images of input sequence is 500 msec.

5. CONCLUSION

In this paper, we proposed an effective automatic face location system that can locate the face region in a complex background when the system is used as a preprocessor of a practical face recognition system for security. We used the two relatively unexplored cues, the motion and the color, to locate face regions in a real scene. First, to segment the face region from the complex background, we used the modified line clustering algorithm that could extract velocity vector field of the face motion. By thresholding this velocity field, the face region was separated from the background. Second, we discussed the issues regarding the color segmentation of the face region. Also, an efficient algorithm based on the thresholding in hue space and knowledge-based location process was proposed to find the eye, eyebrow, and mouth regions. Our experiment showed that the proposed system is very robust in locating the desired face regions. Furthermore, the proposed location system is suitable to be used as the first stage in real time face recognition application since our method is very simple and fast.

REFERENCES

1. T. Kurita, N. Otsum and T. Sato, A face recognition method using higher order local autocorrelation and multivariate analysis, *ICPR* **2**, 2–11 (1992).
2. A. L. Yullie, D. S. Cohen and P. W. Hallinan, Features extraction from faces using deformable templates, *Proc. CVPR* 104–109 (1989).

3. R. Brunelli and T. Poggio, Face recognition: features versus templates, *IEEE Tr. on PAMI*, **15**, 1042–1052 (October 1993).
4. M. Kirby and L. Sirovich, Application of the Karhunen–Loeve procedure for the characterization of human faces, *IEEE Tr. on PAMI*, **12**, 103–108 (January 1990).
5. B. S. Manjunath, R. Chellappa and C. von der Malsburg, A feature based approach to face recognition, *Proc. CVPR* 373–378 (1992).
6. O. Nakamura, S. Mathur and T. Minami, Identification of human faces based on isodensity maps, *Pattern Recognition*, **24**, 263–272 (1991).
7. V. Govindaraju, S. N. Srihari and D. B. Sher, A computational model for face location, *Proc. IEEE, 3rd Conf. Comput. Vis.* pp. 718–721. Osaka, Japan (1990).
8. V. Govindaraju, S. N. Srihari and D. B. Sher, A computational model for face location based on cognitive principles, *Proc. AAAI'92*, pp. 330–355. San Jose, California (1992).
9. G. Yang and T. S. Huang, Human face detection in a complex background, *Pattern Recognition*, **27**, 53–63 (1994).
10. J.-Q. Fang and T. S. Huang, Some experiments on estimating the 3-D motion parameters of a rigid body from two consecutive image frames, *IEEE. Tr. on PAMI*, **PAMI-6**, 545–594 (September 1984).
11. T. J. Broida and R. Chellappa, Estimation of object motion parameters from noisy images, *IEEE Tr. on PAMI*, **PAMI-8**, 90–99 (January 1986).
12. B. K. P. Horn and B. G. Schunck, Determining optical flow, *Artif. Intell.* **17**, 185–203 (1986).
13. C. L. Fennenma and W. B. Thompson, Velocity determination in scenes containing several moving objects, *Comput. Graphics Image Process.* **9**, 301–315 (1979).
14. W. Reichardt and T. Poggio, Figure-ground discrimination by relative movement in the visual system of the fly, *Biol. Cybern.* **35**, 81–100 (1979).
15. B. G. Schunck, Image flow segmentation and estimation by constraint line clustering, *IEEE. Tr. on PAMI*, **11**, 1010–1027 (1989).
16. J. L. Barron, D. J. Fleet and S. S. Beauchemin, Performance of optical flow techniques, *Proc. CVPR* 236–242 (1992).
17. T. Uchiyama and M. A. Arbib, Color image segmentation using competitive learning, *IEEE. Tr. on PAMI*, **16**, 1197–1206 (December 1994).
18. Y. Ohta, T. Kanade and T. Sakai Color information for region segmentation, *Comput. Graph. Image Process.* **13**, 222–241 (1980).
19. Y. W. Lin and S. U. Lee, On the color image segmentation algorithm based on the thresholding and the fuzzy *c*-means techniques, *Pattern Recognition*, **23**, 935–952 (1990).
20. J. Liu and Y.-H. Yang, Multiresolution color image segmentation, *IEEE. Tr. on PAMI*, **16**, 689–700 (July 1994).
21. W. K. Pratt, *Digital Image Processing*, John Wiley & Sons, Inc. (1991).
22. M. Nadler and E. P. Smith, *Pattern Recognition Engineering*, John Wiley & Sons, Inc. (1993).
23. A. P. Witkin, Scale space filtering: a new approach to multi-scale description, *Image Understanding*, S. Ullman and W. Richards, (eds), pp. 79–95, Ablex Publishing, New Jersey (1984).
24. C. H. Lee, J. S. Kim, Y. K. Jung and K. H. Park, A real-time, parallel face recognition on task-level pipelined multicomputer, *Proc. IEEE Int. Conf. on SMC*, Vancouver, Canada, **3**, 2289–2294 (October 1995).
25. Irwin Miller and J. E. Freund, *Probability and Statistics for Engineers*, Prentice-Hall, New Jersey (1985).

About the Author—CHOONG-HWAN LEE was born in Cheon Ju, Korea on August 18, 1968. He received the B.S. degree in Electronic Engineering from Hanyang University, Korea in 1991, and M.S. degree from Korea Advanced Institute of Science and Technology (KAIST) in Electrical Engineering, Korea in 1993. He is now a Ph.D. candidate in Department of Electrical Engineering, KAIST. His current interests include computer graphics, graphics architecture, parallel processing, and computer vision.

About the Author—JUN-SUNG KIM was born in Pusan, Korea on April 11, 1969. He received the B.S. degree in Electronic Engineering from Pusan University, Korea in 1993, and M.S. degree from Korea Advanced Institute of Science and Technology (KAIST) in Electrical Engineering, Korea in 1995. He is now a Ph.D. candidate in Department of Electrical Engineering, KAIST. His current interests include computer graphics, graphics architecture, parallel processing, and computer vision.

About the Author—KYU HO PARK was born in Sang Ju, Korea on October 19, 1950. He received the B.S. degree from Seoul National University, Korea in 1973, and the M.S. degree from the Korea Advanced Institute of Science and Technology (KAIST), in 1975, and Dr Ing. Degree from the Universite de Paris, Paris, France, in 1983 all in Electrical Engineering. He worked for a development of EPABX at a private company from 1975 to 1978. He was awarded a France Government Scholarship during 1979–1983 and he studied at the Laboratoire des Signaux et Systemes, CNRS. He is now professor at the Department of Electrical Engineering, Korea Advanced Institute of Science and Technology. His major interests include computer vision, computer architecture and parallel processing. Dr Park is a member of KISS, KITE, IEEE.