Locating and Extracting Human Face in a Cluttered Scene by using Genetic Search

Non-member Hertog Nugroho (Keio University)
Non-member Shigeyoshi Takahashi (INES Corp.)
Non-member Yoshiteru Ooi (INES Corp.)
Member Shinji Ozawa (Keio University)

This work investigates a new approach to detect human face in a cluttered scene. Our method consists of two main search procedures, both using genetic algorithm. The first one is to locate a head inside the scene and the second one is to identify the existence of face within the extracted head area. For this purpose, we have developed two models to be used as a tool to calculate the fitness for each observation in the search procedure: One is a head model which is approached by an ellipse and the other one is a face template which size is adjustable. To find a face, we exploited face salient features such as eyes and nose/mouth on binary image where the features look darker. The procedures work sequentially. The head search is activated first, and after the head area is found, the face detection is activated. The experiment demonstrates the effectiveness of the method. Since our ultimate aim is to provide a good quality of face image for recognition, we also made comparison between face image magnification by linear interpolation with that obtained by zooming camera. The results gives our preference toward that obtained by zooming camera.

Key words: genetic algorithms, ellipse model, template model, facial features, zooming.

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 such as criminal identification, surveillance, security system, video conferencing and so on.

Development of a fully automatic human face recognition requires two steps. The first step is to detect and find the location of faces in a given unknown picture. The next step is to extract features from the faces and compare them with known database face models. Under less restrictive condition, considering the situation where the person exists but he/she is not facing the camera, the first step should be broken again into two sub-steps. The first one is to locate a head and the second one is to detect, within the extracted head area, whether we have a face or not. However, since both the location and the size of a face in a scene are unknown, the problem is very difficult.

Recently, the problem of automatic face location had been addressed. In reference a hierarchical knowledge-based method consisting of three levels was utilized to locate unknown human faces. While the approach seems promising, the run time required to detect a face was still long. In reference a motion information is used to separate the face region from the background and then the color information is used to find eye, eyebrow, and mouth regions of the face. The run time for this approach is impressive, but as it is necessary to extract facial features in detail, it lacks size dimension of the object. In reference a face is regarded as a textual object. By using several textual features, the detection of face is done by scanning the whole image using a predetermined size window. The approach is capable to detect face under some degrees of rotation and view. But, since the size of the face can not be determined beforehand, the search procedure becomes exhaustive and consequently, the run time is long.

In this paper, we attempt to develop an automatic face extraction method by at first, isolating a person under surveillance from background, and by using templates to locate head and to detect face. Our goal is to extract a face from the moving person scene and to
provide a face image recognizable enough for further identification. Fig. 1 illustrates the overall procedure of our proposed method. Assuming one moving person with normal posture behaviour, by applying simple image subtraction method, we can isolate the overall posture area of the person and then concentrate on the area to find his/her head. Locating the head is done by using an ellipse model as a template under genetic search. Our consideration to use genetic algorithm is based on the argumentation that our problem can be categorized into one of search or optimization problems.

After the head area is located, its size is determined and then, concentrating on the extracted area inside the ellipse, we try to detect a face. The main property of human head (including face) which we want to exploit is its color. In grey scale image, the human hair is usually darker than the skin of his face. Also some salient features of human face such as eyes, nose and mouth look darker than the skin. By thresholding, we can count the dark pixels portion. In case of a rather frontal view face (dark pixels portion is less than some threshold), the second search procedure to locate facial features, also based on genetic algorithms, is activated. For this purpose, we have designed a facial feature template and counting the edges along the template and dark pixels within the template to calculate its fitness. Our approach is similar with that proposed in [13], but we use different strategy for calculating the objective function and also our approach is more flexible in terms of size and capable to handle persons wearing glasses. To get a good quality of face image, we have made a study to compare two face image magnification methods, i.e.: interpolation and by using zooming camera. The result gives our preference to the second one.

This paper is organized as follows. The main procedures, i.e: object extraction, head localization and face detection will be discussed subsequently following this section, which are presented in Section 2, 3, and 4 respectively. In Section 5, an implementation of the method by using zooming camera will be discussed followed by a brief discussion on the computational overhead which is presented in Section 6. In Section 7 the conclusion is discussed.

2. Object Extraction

Since in real world people always moves, it is reasonable to exploit motion information to get the overall posture of the subject. Two things can be benefited from this approach: the first one is that the search area can be reduced significantly and thus, makes it easier for the search procedure to converge to its optimum value; and the second one is that the processing time is fast, as has been mentioned in [13], because it is not necessary to scan the whole image.

We used three frames image subtraction method which is simple and yet powerful enough to discriminate between moving object and another non-moving ones. First, the sensing system inputs a series of image I from a camera sensor. Let $S(t_n - \Delta t)$, $S(t_n)$, and $S(t_n + \Delta t)$ be edge images extracted by Sobel operation from $I(t_n - \Delta t)$, $I(t_n)$, and $I(t_n + \Delta t)$, respectively. $\Delta t$ is the temporal difference in the images. Then, the difference image $D_-$ can be obtained by subtracting two successive edge images as follows:

$$D_- = pos(S(t) - S(t - \Delta t))$$

where

$$pos(x) = \begin{cases} x & \text{for } x > 0 \\ 0 & \text{otherwise} \end{cases}$$

With the same procedure, the difference image $D_+$ can be obtained by subtracting $S(t)$ with $S(t + \Delta t)$. Please notice that the subtrahend in both procedures is similar, i.e: $S(t)$. It is necessary when we need to get moving object portion in current state. By multiplying $D_-$ and $D_+$ and thresholding, the edges of moving person at current frame $\Omega_t$ is obtained, while the remaining background noise is greatly reduced.

3. Head Localization

After the overall posture of a moving person has been extracted, the next procedure is to locate the head of
the person. For this purpose, we use an ellipse model representing the shape of human head. Strictly speaking, the shape of human head can not be approximated by ellipse and in real world, the shape is affected by several factors such as the model of hair, the view posture, gender and so on. But in most cases it consists of several convex contours in almost closed formation. This information, together with the use of genetic search, will be sufficient for head localization.

Fig. 2. Procedure to search a head.

The skeleton of the procedure is illustrated in fig. 2. Here we have an edge map of a moving person as an input.

Fig. 3. Determining area of interest and starting point.

To reduce the searching area, at first the boundary of the object and the starting point is determined as is illustrated in fig. 3. The boundary of the object is determined by recording the outermost points in left-, right-, upper-, and lower-side. We use a priori knowledge to set the starting point of the search at the location where the head likely can be found. Assuming normal motion activity, the human head usually can be found in the upper-side of human body. With this assumption, the starting point is determined at the middle of the area of interest (i.e. \( w_s \) is half of the width of the area \( w_a \)) and at about the center of the head (i.e. \( h_s \) is half of \( w_s \)).

Fig. 4. A Head Model.

The ellipse parameter set used to interpret the genetic string consists of: center of ellipse \((x_s, y_s)\), two principal axis \((A, B)\), and inclination \((\theta)\) as was shown in Fig.4 with the following constraints:

- The center of ellipse should be in the vicinity of the starting point.
- Major axis \( A \) should have a length almost equal to \( h_s \), more or less.
- Inclination \( \theta \) should be within the range of tolerable angle.

With the above constraints, the search area can be limited while at the same time we can save bit requirement for the string without eliminating their flexibility.

The objective function \( F_{head}(x_s, y_s) \) used to evaluate the parameter set generated by genetic string for this search is as follows:

\[
F_{head}(x_s, y_s) = \frac{\left( \sum_{i=1}^{n} h(x, y) \right)}{l_e} \quad \text{(3)}
\]

\[
(x, y) \in F_{ellipse}(x_s, y_s, A, B, \theta)
\]

where

\[
F_{ellipse}(x_s, y_s, A, B, \theta) = \frac{((x - x_s) \cos \theta + (y - y_s) \sin \theta)^2}{A^2} + \frac{(-(x - x_s) \sin \theta + (y - y_s) \cos \theta)^2}{B^2} = 1\quad \text{(4)}
\]

and \( l_e \) is the contour length of the ellipse, \( \Omega \) is a smoothed version of edge map \( \Omega \), and \( h(x, y) \) is pixel intensity at location \((x, y)\).

By using eq. 4, each set of ellipse parameters, derived from the generated string, is evaluated. In the cycle of
genetic algorithm, those with higher fitness will replace the weaker one. The process is halted when the terminate condition is satisfied or the iteration count exceeds a certain value. The terminate condition is considered satisfied if the difference between the highest fitness of the population and the average of sum of fitness of the upper half of the population is less than a certain threshold. In this case, the head is considered found and located and the ellipse parameter set translated from the highest fitness string among the population at the last generation is recorded. Generating an ellipse by using this parameter set onto source image will result an area where there is most likely that we can find a face.

If the iteration count exceeds a certain termination count, the system is forced to halt and try another chance at the next frame. In this case we assume that the system can not detect a face.

To demonstrate the effectiveness of our proposed method, the preliminary experiments was conducted on several persons walking arbitrary. The scene were recorded and converted into 256 × 240 sized image frames and then fed consecutively into the system. Termination count were not restricted to see how long does the system requires to reach the terminate condition.

Figure 5-a shows some examples of edge maps Ω and 5-b shows the search result represented by ellipse which is superimposed onto source images. We consider here that the head is accurately located if most of the edges of head outline is located on the ellipse and the whole face area is within the ellipse with as few background as possible, justified by visual observation. Examining the edge images we can see some cracks around the head and at some other location we didn’t have edge at all. But the procedure has demonstrated that even with this ill-condition, it can still accurately locate the head and estimate its size. Similarly, we conducted 485 trials and we found out that the procedure gave 446 correct results (approx. 91 % effectiveness).

Figure 6 illustrates the relation between the termination count and the head localization probability. We see that if we put the count to a higher value, the probability to localize the head is higher, but at the cost of computing time. As a real figure, at the termination count of 800, the procedure requires about 16 seconds under Sun S4/20A with 125 MHz hyperSPARC CPU.

4. Face Detection

After the assumed head area has been located, the next step is to analyze whether, within the area, we have a face or not. In the following section a template used to detect facial features is described, followed by the objective function used in the evaluation. Then, the result of experiment is demonstrated in the subsequent section.

4.1 Facial Feature Template. We assume that human face, under ambient lighting exhibits darker appearance around its hair and its facial features, provided that we are working with grey-level image. In a frontal face, the facial features are considered to have a certain geometrical relation property while hair region doesn’t have any, apart from its location which is usually in the upper section.

Based on this assumption, we consider to use facial feature templates as were illustrated in fig. 7. It is based mostly on 3 salient facial features: eyes and mouth. Since we also plan to use genetic search to find the features, from the template, we derive 10 parameters to be used in interpretation of the genetic string. They are: displacement from center of template \((x_r, y_r)\) with respect to center of ellipse \((x_e, y_e)\), distance between eyes(ew),
distance between eyes axis and mouth \((em)\), eyes and mouth widths \((w_k, k = 1, 2, 3)\), and heights \((h_k)\), all normalized with respect to ellipse size, and template inclination \((\alpha)\). The following constraints are applied to determine the hit requirement for the string:

- Template displacement should not exceed a certain distance from the center of ellipse.
- Distance between eyes \((eew)\) and distance between eyes axis and mouth \((em)\) are initially set to be two thirds of the width of the head \((i.e.: 2 \times B)\). The bit of the string for this parameter is interpreted as a variance around this initial distance.
- Template inclination \((\alpha)\) is independent from ellipse inclination \((\theta)\) but with similar constraint.

Another constraints are described below. With the above constraint, the template can be adapted to various size of face up to a certain degree.

A face is considered detected if the template is located exactly over the top of the darker portion of facial features. Since we are considering to use genetic algorithm to evaluate the parameter set provided by the string, the objective function should reflect this condition and should be defined to have maximum value when the condition is encountered.

To meet this requirement, we utilize dark pixels and edge information from the extracted head area. The dark pixels are obtained by thresholding the area using automatic thresholding method proposed by Otsu \(^{14}\). The method has advantage that the threshold value is calculated automatically and hence the result is invariant against lighting effect. Meanwhile, the edge information is obtained by gradient operator.

As an evaluator, we propose an objective function \(F_{face}(x, y)\) as follows:

\[
F_{face}(x, y) = a_1 \times \left( \prod_{k=1}^{3} e_k \right) + a_2 \times \left( \prod_{k=1}^{3} d_k \right) - a_3 \times eew/\omega \tag{5}
\]

under the following constraints:

\[
h_k \leq w_k \quad k = 1, 2, 3.
\]

\[
w_e \leq eew
\]

where

\[
e_k = \frac{r_{e+}+w_e/2}{g(x, y)} + \frac{y_{e+}+h_e/2}{g(x, y)}
\]

\[
g(x, y) \in G(x, y) \tag{6}
\]

\[
d_k = \sum_{y=y_{e+}-h_k/2}^{y_{e+}+h_k/2} \sum_{x=x_{e+}-w_k/2}^{x_{e+}+w_k/2} b(x, y)
\]

\[
b(x, y) \in B(x, y) \tag{7}
\]

\(B(x, y)\) denotes binary map of the head region within ellipse after thresholding and \(G(x, y)\) denotes its smoothed edge version, while * could be 1 or 2. Function \(e_k\) is a summation of edge pixels intensity along the template compared with its maximum value. The idea is similar with that of head localization, such that it gives maximum value where the edges are located exactly along the template. The product operation on the 3 mini-templates ensures that a match condition should occur on all mini-templates. It guarantees that the geometrical relation between facial features is examined and satisfied.

Function \(d_k\) is a dark pixels counter inside each mini-template compared with the area of the template. It gives maximum value where the whole area within the template all consists of dark pixels, meaning that the facial features are located precisely within the template.

Please notice that using function \(d_k\) alone in the overall objective function can lead to a false conclusion, since a binarized facial image contains dark pixels not only on its facial features, but also on several areas, especially in the hair section. The case is also true with edges, since the structure of edges in facial image is unpredictable.

The third term on the right hand side of the objective function (eq. 5) behaves as a pulling string between eye templates. It will contribute to a maximum overall estimation value when it encounters two similar blobs located next to each other. We exploit a fact here that
eyes have similar shapes and there is no significant feature between them (i.e. the parameter $eew$ should be small). Hence, if there are three linearly located blobs around the eyes area, two nearly located ones are preferred. Eyes width $w_e$ is expected to be large, but should not exceed $eew$ to prevent overlapping of the eyes templates.

By using this objective function and implementing the similar strategy with that of head localization, the facial features can be detected at the location where the edges of facial features fit along the template while at the same time dark pixels portion of facial features are entirely covered by the same template.

Our approach to this problem is similar with that in [1] in that both of us exploited the geometrical relation between face salient features and therefore, used face template to find them. But, the strategy to obtain our goal is different. They used only edge map of the face as an input and the objective function to measure the search was developed accordingly. Furthermore, the size of the face was set to its normalized size (which is almost covering the whole scene) and the search was done in a "brute-force" style. As a consequence, it takes much of computer time.

In our strategy, we used not only edge map, but also color map (realized as binary map). This has advantage over previous method in that the inherence of face salient feature can be exploited optimally, while both edge map and color map can define blobs more accurately. In proposing objective function, we are not only considering the above features but also the geometrical relation between salient features as well. In the search procedure, we utilized genetic algorithm with some constraints to optimize the search and hence, as will be demonstrated in the experiment, the search time is much faster.

4.2 Results of Face Detection. The experiment was conducted indoor. A person was required to walk arbitrarily within the scope of camera view. The scene was recorded into a video format and then converted into 256 x 240 sized image frames. Parameter $a_1$ is set larger than $a_2$ to give preference on detecting edges over dark pixels. In our experiment these parameters are set 1.4 and 0.4 respectively.

Figure 8 illustrates one example of face detection procedure. Figure 8-a shows a source image with the head area covered by an ellipse as a result of head localization procedure. Figure 8-b is a blow version of the head area after thresholding $B(x, y)$. It is apparent that our claim in the previous section that the facial features look darker as well as the hair section is likely. Figure 8-c is a smoothed edge map of the head area $G(x, y)$. Again we can see that edges appear around facial features, but also somewhere else which is why we need also dark pixel information to help detecting the facial features. Figure 8-d demonstrates the detection result. Figure 8-b, -c, -d are magnified version (for visualization purpose). The procedure were run in the actual original size.

Fig. 8. An example of face detection procedure. (a) Source image with head found and localized. (b) Thresholding the head area $B(x, y)$ (magnified). (c) Smoothed edge map of the head area $G(x, y)$ (magnified). (d) Detection result (magnified).

Fig. 9. Some examples of face detection results under various size and orientation.
Table 1. The effect of size on the detection performance.

<table>
<thead>
<tr>
<th>Face Size (in pixels)</th>
<th>Without glasses</th>
<th>With glasses</th>
</tr>
</thead>
<tbody>
<tr>
<td>18 x 24</td>
<td>1/11 (9)</td>
<td>1/5 (20)</td>
</tr>
<tr>
<td>22 x 29</td>
<td>36/43 (86)</td>
<td>39/54 (72)</td>
</tr>
<tr>
<td>28 x 38</td>
<td>72/70 (94)</td>
<td>43/48 (86)</td>
</tr>
<tr>
<td>34 x 46</td>
<td>77/63 (90)</td>
<td>65/62 (77)</td>
</tr>
<tr>
<td>40 x 53</td>
<td>13/16 (81)</td>
<td>12/13 (90)</td>
</tr>
<tr>
<td>Total</td>
<td>199/231 (86)</td>
<td>143/182 (85)</td>
</tr>
</tbody>
</table>

Note: Numbers in each cell represent successful detection/total trial and (effectivity in %).

Figure 9 shows some examples of face detection results. Orientation angles were not precisely measured, but the subjects are allowed to turn until about 45 degrees from frontal view. Although difficult to be observed, subjects illustrated in the first row actually wore glasses.

Table 1 and table 2 show the detail of experiment conducted on 5 sequences of images, each of which consists of 70 - 100 image frames. Two of them were taken from the sequences where the subjects wore glasses. Table 1 demonstrates the effect of the size of face image on the detection, while table 2 demonstrates the effect of face orientation. Detection is considered successful if all facial features (both eyes and mouth) are located inside the template, justified by visual observation.

Table 2. The effect of face orientation on the detection performance.

<table>
<thead>
<tr>
<th>Condition of Subject</th>
<th>Left Orientation</th>
<th>Frontal view</th>
<th>Right Orientation</th>
</tr>
</thead>
<tbody>
<tr>
<td>w/o glasses</td>
<td>36/43 (87)</td>
<td>116/127 (91)</td>
<td>40/52 (88)</td>
</tr>
<tr>
<td>w glasses</td>
<td>57/75 (74)</td>
<td>61/70 (87)</td>
<td>25/32 (78)</td>
</tr>
</tbody>
</table>

Note: Numbers in each cell represent successful detection/total trial and (effectivity in %).

From table 1 we can see that, apart from the smallest size, the effectivity of our method is more than 85%, while the effect of glasses only degrades it a little. This is a little better than that obtained in [11]. For the cases of 18 x 24 sized face image, we observed that the facial features become very complex, touching each other, and sometimes one feature didn't just appear, and thus violating the condition for good detection. Meanwhile, glasses causes different appearance on the face due to reflection and shading and sometimes they even obscure facial features, especially eyes. But, even with this severe condition, our detector can still give a tolerable performance.

From table 2, the effect of orientation also clearly can be seen. Please note that the trials were conducted without including 18 x 24 sized face image. The degradation of performance on the orientation other than frontal view is expected. It is caused by the fact that the template is developed to be optimized against facial features in frontal view position. Indeed, it is allowed to vary to a certain degree, and also is given tolerance to inclination. But the mini-templates, not the whole template, are still in upright position. We observed that this fact contributes the mis-detection in several cases, especially in wide angle orientation, where the face tends to incline and so with the element of facial features. But, as has been demonstrated, the performance is not much degraded.

5. Face Image Extraction by using Tracking and Zooming Camera

In this section, we describe the possible implementation of our method in a real environment. We consider a scenario of locating a personal head in outdoor area, detecting a face and then magnifying only the face area before recording. Here, we propose to use tracking and zooming camera for some reason. Intuitively, the use of zooming camera with a right focus will provide an image magnification better than just purely image magnification (usually by interpolation). In some cases, image magnification even gives poor results because it tends to blur the object and hence destroys the important information. Considering the development of a fully automatic face recognition system where its performance is heavily relied on the quality of input image, the prominent face features should be easily discriminated. Therefore, it is reasonable to use zooming camera to obtain a better quality of face image.
illustrates the complete system. It consists of a Work-
station SUN SS2, a general purpose image processor
CORE, a CCD camera NEC TI-324A, a controllable
zoom-lens Canon J10xREA2 and a robot arm Move-
master RV-M1. The image processor is used to convert
video signal recorded by camera into a 512 x 432 sized
8-bit grey level image frame. The robot arm is used to
move the camera in terms of pan and tilt.

Fig. 11. Camera parameters.

The controller program developed by our collegues,
Hwang et al. [12] was used to control the pan, tilt, and
zoom of the camera (for the definition of pan, tilt and
zoom, see Fig. 11) and the following is a brief overview
of that method.

Let a moving object with position \( P_n = [x_n(t_n),
 y_n(t_n)]^T \) and velocity \( V_n \) in time \( t_n \) is subjected to
be followed by an observing camera. The estimated posi-
tion \( D_n = [x_n(t_n), y_n(t_n)]^T \) of that object is given by

\[
D_n = P_n + \Delta t V_n \tag{8}
\]

where \( \Delta t \) is usually the system sampling period. In our
case, the estimate position is simply the ellipse center
\([x_e, y_e]^T\). To follow the object trajectory, it is impera-
tive that the direction of optical axis of camera is con-
trolled to correspond with the centroid of the object,
which can be done by controlling pan and tilt of the
camera. Both pan and tilt angle \( \theta_{pan}(t_n), \theta_{tilt}(t_n) \) are
given by

\[
\theta_{pan}(t_n) = \tan^{-1} \left( \frac{d_x(t_n)}{\epsilon_x f(t_n)} \right) \tag{9}
\]

\[
\theta_{tilt}(t_n) = \tan^{-1} \left( \frac{d_y(t_n)}{\epsilon_y f(t_n)} \right) \tag{10}
\]

where \( \epsilon_x, \epsilon_y \) are the vision system offsets calibrated
at 80.11 and 78.41 pixel/mm at \( f(0) \) respectively, and
\( f(t_n) \) is the current focal length.

To obtain the desired size of object, the focal length
of zoom-lens can be controlled by giving a signal with a
certain period, which the relation is illustrated in Fig.
12. Assuming small changes, the relation can be ap-
proached by

\[
\frac{\Delta f(n)}{\Delta i_n} = \frac{df(t_n)}{di_n} \tag{11}
\]

Rearrange for zoom control signal \( \Delta i_n \), we have

\[
\Delta i_n = \left( \frac{df(t_n)}{di_n} \right)^{-1} \Delta f(n) \nonumber
\]

\[
= \left( \frac{df(t_n)}{di_n} \right)^{-1} f(t_n) \left( \frac{a}{a_n} - 1 \right) \tag{12}
\]

where \( a_n \) and \( a \) are current and desired object size re-
spectively. Initial focal length \( f(0) \) is given as 10.0 mm
and the period of zoom control should be less than
\( \Delta t_{max} \). In our case, zoom magnification depends on the
size of face image, the desired size, and the maximum
zooming capability of the camera.

Some examples of magnification results are illustrated
in Fig. 13. Fig. 13-a shows subjects in various distance
locations from camera. Three of the subjects wore glass-
es. Fig. 13-b shows the face detection results using the
procedure previously discussed, and magnified by us-
ing linear interpolation. Here we see that, due to poor
lighting condition, some mini-templates didn’t fit onto
the actual facial features as was shown for example in
the third row. This is due to the fact that glasses cause
different shading effect depending on where the light
source comes from. Since input image for our detector
depends on the distribution pattern of dark pixels, this
certainly affects the detection results. But, even under
this poor condition, the overall results demonstrate the
effectiveness of our detector.

From fig. 13-b, we can also see the effect of subjects’
location on magnification. A far-located one gives a
heavily blurred result, making it almost impossible to
be used for recognition, while a near-located one causes
only a slight blurring effect, but is still difficult to be
used for recognition.
Fig. 13. Some result examples on the experiment of image face detection by zooming camera. (a) The original image. (b) The magnification of the detected face image by interpolation. (c) The magnification of similar image by zooming camera.

In fig. 13-c, we can see the magnification on the same face images by using zooming camera. Here we see that the quality of images is far superior than that shown in fig. 13-b. We can easily identify the subject and the salient face features can also easily discriminated, making them feasible for automatic face recognition.

6. Computational Overhead

In this section we present a brief discussion of the computational time spent in each process of the head localization and face detection (See Table 3). In this analysis, we used Sun SS2 machine. Due to the load fluctuation and the dependency of the algorithm on the complexity of the image, the time taken to a given computation process can not be estimated exactly. However, the table gives a rough idea about the time required by each process in the algorithm. As can be seen from the table, most of the time are spent in the searching procedure, either in head localization or in the face detection. But the figures are still faster than those reported in [10]. Actually, special hardware can be developed to exploit the parallel characteristic of the genetic algorithm.

7. Conclusion

This paper has investigated a new approach to the issue of localization and detection of human face in a cluttered scenes. We have presented a new model-based searching method based on genetic algorithm which localizes a head and then, detects the face. Our contribution is the use of genetic search in combination of models and objective functions as a mean to find the optimal solution.

Preliminary experiment concerning the head localization and face detection demonstrates the acceptability of the detector giving the effectiveness of above 90 % and 80 % respectively. We also have demonstrated the use of zooming camera to get a better result which has been proved by the results of our experiment. Our ultimate aim, which is to provide face images for automatic recognition is accomplished by Justifying the result examples.

Some remaining problems still has to be addressed in order to improve the proposed method. The most crucial one is the searching method. It is necessary that the searching time should be faster in order for the system to be considered in real time application. The genetic search is good to provide optimal result, but we still need a faster method. Another one is lighting condition. Our method works well when there is enough light around the subject, especially in indoor environment. But under direct lighting, such as under the sun, the appearance on the face is much different. In order

<table>
<thead>
<tr>
<th>Process</th>
<th>Required times (in seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object Extraction</td>
<td>0.8 ~ 1.2</td>
</tr>
<tr>
<td>Head Localization</td>
<td>9 ~ 30</td>
</tr>
<tr>
<td>Extraction of Face within ellipse</td>
<td>0.5 ~ 0.8</td>
</tr>
<tr>
<td>Face detection</td>
<td>40 ~ 60</td>
</tr>
<tr>
<td>Zooming</td>
<td>3 ~ 6</td>
</tr>
<tr>
<td>Overall procedures</td>
<td>53.3 ~ 98</td>
</tr>
</tbody>
</table>
to develop a robust system, such condition should be taken into account.

In terms of hardware, we found another drawback in zooming mechanism. Since it is mechanical, a hardware change can’t speed up the process much. To overcome this problem, we are considering to use two cameras installed in a piggy-back construction, one on top of another. One camera is used for tracking and another one for close up. This scenario can eliminate the zooming time problem. We are considering this scenario in the future.

Acknowledgement. We would like to thank Mr. Hasegawa for discussion concerning the topics and all member of Ozawa laboratory for their support. We also appreciate the comments by anonymous reviewers, for without them, this paper would never materialize. The first author is supported by Hitachi Scholarship Foundation.

(Manuscript received June 26, 1997, revised December 12, 1997)

References


Hertog Nugroho (Non-member) was born in Jakarta, Indonesia on May 15th 1959. He received the B.E degree from Bandung Institute of Technology, Bandung, Indonesia, in 1984 and the M.E degree from Keio University, Yokohama, Japan in 1995. He is now pursuing a doctoral degree at Department of Electrical Engineering, Keio University. His research interest is in pattern recognition.

Shigeyoshi Takahashi (Non-member) was born in Nagano, Japan on January 24, 1968. He received the B.S degree from Nihon University, Tokyo, Japan in 1991. He joined the INES Corporation, Yokohama, Japan in 1991. He is currently a research engineer of the Advanced Research Center of INES. His research interest is in image processing.

Yoshiteru Ooi (Non-member) was born in Kagawa, Japan on March 7th, 1959. He received the B.S degree from Gakushuin University, Tokyo, Japan in 1984 and the M.S degree from Rikkyo University, Tokyo, Japan, in 1986. He joined the INES Corporation, Yokohama, Japan in 1986. He is currently a research engineer of the Advanced Research Center of INES. His research interest is in image processing.

Shinji Ozawa (Member) was born in 1943. He received B.E., M.E., and Ph.D degrees all in Electric Engineering from Keio University, Tokyo, Japan, in 1967, 1969, and 1972, respectively. He had been Research Associate, Assistant Professor, and Associate Professor from 1970 in the Department of Electric Engineering of Keio University. He is now Professor of same department. At 1984, he had been Visiting Professor of University of Maryland, USA. He has worked on Digital Communication and Signal Processing of Image and Voice. He is a member of IEICE(Japan), IPSJ(Japan), and IEEE(USA).