DOCTORAL THESIS

Photographic Environment Independent Multiview Face Detection and Tracking Using Template Generation by Genetic Algorithm

Graduate School of Engineering, Iwate University
Docter's Course, Design & Media Technology
Junya Sato

March 2017

Acknowledgements

In advancing this research, I received many beneficial advices for the research and supports from many people. I could not have finished this thesis without these many helps. I deeply express my gratitude for people who supported me.

Dr. Kouichi Konno and Dr. Takuya Akashi mainly supervised me and they gave me many useful advices and comments while I belonged to smart computer vision laboratory (SMART CV LAB.). They also supported my life. I really appreciate them. Dr. Kouichi Konno, Dr. Takuya Akashi, and Dr. Tadahiro Fujimoto responsibly undertook the chief examiner and vice-chairman for this thesis. Their comments were very helpful to write this paper. I am grateful for your supports.

I met many people who entered SMART CV LAB. and graduated until I completed this thesis. I had beautiful days with them and I cannot forget memories. Especially, I would like to appreciate Mr. Chao Zhang, who is my classmate in SMART CV LAB. He gave me much valuable information about researches. His information were very useful for this research. I hope to work hard together in the future.

Finally, I am very grateful for my parents, sister, grandparents, and relatives because they helped and supported me long time. I am also grateful to Dr. Norishige Chiba, the honorary professor of Iwate University, for his help. Thank you very much.

Junya Sato

March, 2017

Published Contents and Contributions

Journal Papers

1. Junya Sato and Takuya Akashi, High-speed Multiview Face Localization and Tracking with a Minimum Bounding Box Using Genetic Algorithm, IEEJ Transactions on Electrical and Electronic Engineering, vol.12, no.5, 2017 (in press).

International Conferences

1. Daichi Oikawa, Junya Sato, and Takuya Akashi, Face Tracking with Protection of the Privacy using Color Histogram and Evolutionary Video Processing, The 2nd International Conference on Intelligent Systems and Image Processing , pp.229-232, Fukuoka, Japan, 2014.

2. Daichi Oikawa, Junya Sato, and Takuya Akashi, Improved Face Tracking with Privacy Protection using Half Ellipse and Additional Region Matching, Joint Conference of IWAIT and IFMIA, CD-ROM, Tainan, Taipei, 2015.

3. Junya Sato and Takuya Akashi, Investigation of Face Tracking Accuracy by Obscuration Filters for Privacy Protection, Irish Machine Vision and Image Processing Conference, pp.121-124, Dublin, Ireland, 2015.

4. Junya Sato and Takuya Akashi, Evolutionary Multi-view Face Tracking on Pixel Replaced Image in Video Sequence, International Conference on Soft Computing and Pattern Recognition, pp.322-327, Fukuoka, Japan, 2015.

Contents

Chapter 1

Introduction

In this thesis, photographic environment independent multiview face detection and tracking is addressed. The photographic environment means a variety of illumination conditions, backgrounds, appearances of objects, and qualities of images in this thesis. The multiview face detection and tracking in these practical environments is a difficult problem. In order to solve this problem, novel methods using template generation and template matching with genetic algorithm (GA) are proposed.

This research mainly focuses on two researches. In the first research, high-speed multiview face detection and tracking in various photographic environments is addressed as a basic research. The proposed method adopts template matching. In this method, a template is created in advance as the first step. Then, a region whose pattern is similar to the template is searched by the sliding window, etc. This is the basic concept of template matching. However, one template is generally able to detect only one pattern. Hence, many patterns of templates are necessary to detect multiview faces. Since generating many templates and calculating all the matching scores are inefficient, a novel method is proposed.

First of all, one 2D face model, whose shape is rectangle, is created. Next, three

parameters are defined to generate templates from the model for multiview faces. By optimizing these parameters using GA, the optimal template can be found efficiently. The GA is a method to find a global optimum within a given search resource. There exist population as search points in a search space and each individual consists of chromosomes. The chromosomes are parameters and they are encoded as bit strings. After the population is evaluated using a fitness function, the chromosomes are updated by iterating genetic operations, such as selection, crossover, and mutation. Since the population evolves to acquire high fitness, a global optimum or its approximate solution can be obtained finally. By introducing GA to the template generation, the optimal templates can be acquired efficiently.

The GA is also applied to target object search instead of sliding window. This method scans the whole target image using a search window with a variety of scales. This is equal to that all the combinations of the coordinate and the scale parameters are checked. Since this approach is inefficient and in-plane rotation is not considered, GA is applied. By optimizing geometric transformation parameters, such as parallel translation, scale, and in-plane rotation, all the parameters are simultaneously optimized and the target object can be localized efficiently.

For experiments, a challenging 60 video dataset was created. Subjects were recorded under various photographic environments. The proposed method is compared to a machine learning-based method and a face tracking method on the dataset. As a result, high accuracy and fast processing of the proposed method are confirmed.

In the second research, multiview face tracking on privacy protected videos is addressed. Recently, many surveillance cameras have been set at public and private spaces. Along with this, many computer vision techniques, such as pedestrian detection, action recognition, and face recognition are also applied. This is necessary

for crime deterrence, accident detection, and so on. However, some people may suffer psychological pressure about recording their faces. If reducing the pressure is considered, the recoded videos must be preprocessed to protect the privacy. For this purpose, blur and pixelation can be applied and they are often used in news programs. However, the computer vision techniques cannot be used on the preprocessed videos since the original pixel information are changed. Since my proposed multiview face detection and tracking method in the first research of this thesis is also unable to be applied completely, a new method is proposed.

First of all, a new preprocessing method is proposed to generate a privacy protected image and preserve the pixel value. In this method, a $n \times n$ filter is set in a target image and a neighborhood number is determined randomly. Then, the pixel values between the centered pixel of the filter and randomly selected neighborhood are replaced. By applying this filter to the whole image, privacy protected images are generated while the original pixel value is preserved.

Since the color information is preserved, the proposed method uses color histograms as a template to track a multiview face. In this research, Cr and Cb histograms are used because they are robust to illumination changes. The histograms are obtained from a frontal target face. By optimizing geometric transformation parameters using GA and localizing a region whose color histograms are similar to the template in target images, the multiview face tracking can be achieved.

For experiments, a challenging video dataset was recoded under a variety of photographic environments, and the proposed method is compared to related works. As a result, the effectiveness of the proposed method is confirmed.

As described, this thesis contributes photographic environment independent multiview face detection and tracking using by GA.

Chapter 2

Theoretical Background

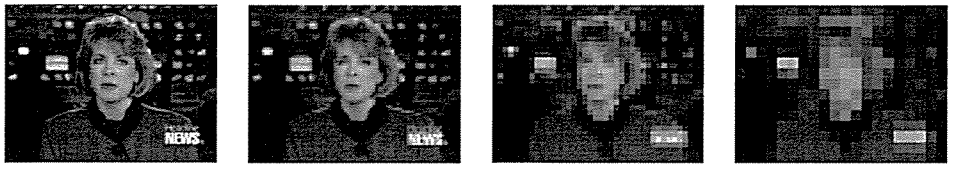
In this chapter, theoretical backgrounds for this research are explained.

2.1 Related Works

As described in the previous chapter, the objective in this research is to achieve multiview face detection and tracking using template matching with genetic algorithm (GA). In this approach, a user must design essential features for the detection. So far, many approaches have been proposed and Yang et al. [1] surveyed them. According to [1], approaches for the detection of faces in a single image are catregorized as below.

- Knowledge-based methods
- Feature invariant approaches
- Template matching methods
- Appearance-based methods

I briefly introduce some methods of each category here. Note that all the figures in this section are from [1].



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

Figure 2.1: Low resolution images, which are used in [2]: a) target image; b) low resolution image (cell size is 4×4 pixels); c) 8×8 pixels; d) 16×16 pixels.

2.1.1 Knowledge-Based Methods

Knowledge-based method is an easy and basic approach. In this approach, researchers design rules based on their knowledge about faces. For example, Yang et al. [2] generate multiple low resolution images (Fig. 2.1) and face candidate regions are searched using rules they set. In the lowest resolution image, the following rules are checked: 1) The center four regions of the face (the dark regions in Fig. 2.2) has a basically uniform intensity; 2) The upper round regions (the light gray regions in Fig. 2.2) has a basically uniform intensity; 3) The difference between the average gray values of the center regions and the upper round regions is significant. Based on these rules, the face candidate regions are found by sliding window. Then, the found candidate regions are further processed in higher resolution images (Fig. 2.1(b) and (c)).

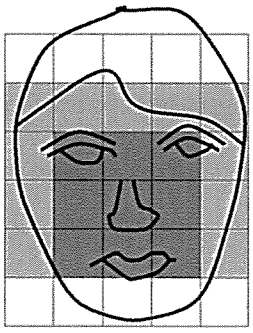


Figure 2.2: Rules using intensity distribution.

2.1.2 Feature Invariant Approaches

In order to find more invariant and effective features than the knowledge-based rules, feature-based methods have been proposed. For example, Sirohey [3] tries to extract faces from complex backgrounds for face identification using canny edge [4]. Graf et al. use band pass filtering and morphological operations to extract high intensity regions, which are eyes, etc [5]. Leung et al. propose a probabilistic method, which is based on local feature detectors and random graph matching [6]. They define a face localization problem as a search problem and try to find face patterns. Augusteijn et al. use face texture information and a nueral network [7]. Chai et al. extract face regions using skin colors which are Cr and Cb components [8]. There are researches using combination of multiple features. For instance, Chen et al. propose a method, which models distributions of skin and hair color in CIE XYZ space [9].

2.1.3 Template Matching Methods

Next is about template matching methods. In these methods, one or some templates that represent standard face patterns are created in advance, and regions whose patterns are similar to the template are searched in a target image using sliding window, etc [10]. According to [1], an early challenge to detect frontal faces on images using template matching-based approach is Sakai et al [11]. In this method, some subtemplates, which represent eyes, nose, mouth, and face contour, are firstly created. Secondly, lines in a target image is extracted using gradient change, and face candidate regions are acquired by computing the correlations between the preprocessed target image and the subtemplates. Craw et al. create a shape template and try to detect a face on an edge image [12]. The image is generated by a sobel filter. Govindaraju also generates an edge-based face model and proposes the

two stage face detection method [13]. Samal et al. [14] adopt a set of face silhouettes as the templates and they are generated using principal component analysis (PCA) [15]. In this method, hough transform [16] is used for the face detection.

Different from edge-based methods, Sinha adopts relationships of average intensities as a feature to cope with illumination change [17]. Figure 2.3 is the used ratio template. This template consists of some regions and average intensities of each region are firstly obtained. Secondly, ratios of the obtained average intensities are calculated. The ratios are calculated based on arrows as shown in Fig. 2.3. The average intensity of the indicated region by the head of arrow is denominator and the tail is numerator. If the calculated ratio is over a pre-defined threshold, one relationship is judged as satisfied. Based on this procedure, all the relationships are checked. As indicated in Fig. 2.3, there are two types of the relationships (black and gray arrows). They indicate essential relationships (black arrows) and confirming relationships (gray arrows). If the number of satisfied essential and confirming relations exceeds a threshold, the focused region in a target image is output as the face detection result. This method is also extended to detect a face and eyes [18].

Since described methods use predefined templates, they are not robust to scale, pose, shape, etc. In order to solve these problems, deformable templates are also proposed [19, 20, 21].

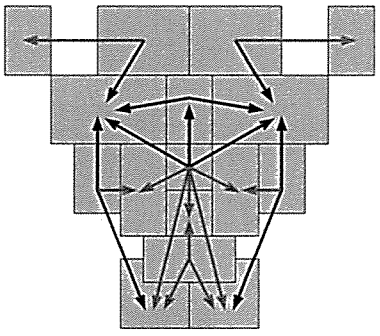


Figure 2.3: Ratio template (14 × 16 pixels).

2.1.4 Appearance-Based Methods

Generally, templates of template matching-based approaches are manually created by researchers in advance. In appearance-based methods, templates are generated by learning using an image dataset. For example, eigen face is generated by applying PCA to the dataset and able to detect and recognize faces [22]. Sung et al. try to face detection by analyzing distributions of image patterns of each class and using a multilayer perceptron classifier [23]. Besides, a probabilistic visual learning method [24], fisherface method [25] are proposed.

2.2 Template Matching

Next, template matching, which is mainly used in this research, is explained. Template matching is a basic object detection method in computer vision. Figure 2.4 represents the outline of a basic template matching by sliding window. The left image is the template, and the right image is the target image. For the simple explanation, the template is the clipped region from the target image. First of all, a search window, which is a candidate region, is set and the similarity between the template and the candidate region is calculated. If the similarity is over a predefined threshold, the region is stored as a detection result. Otherwise, the similarity

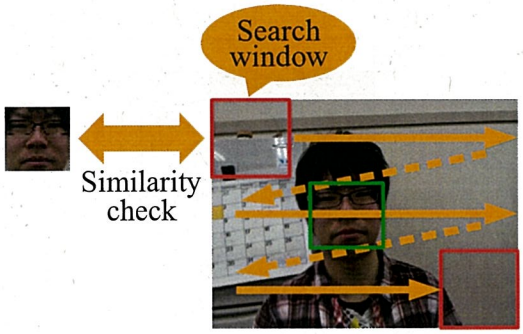


Figure 2.4: Outline of a basic template matching by sliding window.

is calculated again after the search window shifts to the right some pixels. When the window arrives at the right edge of the target image, the window starts scanning again from the left edge after the window is set some pixels below. This processing is iterated until the search window arrives at the right bottom of the target image. Since it is necessary to address scales and in-plane rotations of a face, the window with a variety of sizes and rotation angles iteratively scans. By this procedure, the face region can be localized finally. This is a basic template matching by sliding window.

The template in Fig. 2.4 is the clipped region of the target image for the simple explanation. Hence, this template can detect only specific person's face. In order to detect various faces, it is necessary to create a template, which has common features or patterns in the various faces. In this research, different templates are designed and used for multiview face detection and tracking, and multiview face tracking on privacy protected videos. These details are described in each chapter.

2.3 Search method

As represented in Fig. 2.4, sliding window is often used when a face in a target image is searched. Nevertheless, iteratively scanning the whole image using a variety of scales is ineffective. In order to solve this problem, some effective search approaches are proposed. For instance, Barnea et al. propose sequential similarity detection algorithm (SSDA) in 1972 [26]. This method accumulates the difference of each pixel value when the dissimilarity between a template and a candidate region is calculated using sum of absolute difference (SAD), etc. If the accumulated value exceeds a predefined threshold, the current candidate region is rejected. Since it is not necessary to check all the pixels, this approach can reduce the calculation cost. Tanimoto proposes an effective search method by pyramids search [27]. This

method firstly generates some kinds of low resolution images. Next, candidate regions are obtained by applying sliding window in the lowest resolution image. After that, neighborhoods of the obtained regions are searched in higher resolution image. By repeating this procedure, the face region can be acquired effectively. Since the sparse and dense searches are performed in the low and high resolution images, this approach is also called as coarse-to-fine. In this method, some parameters for the number of low resolution images and the search ranges in each resolution image must be set in advance.

Murase et al. point out that the coarse-to-fine method sacrifices accuracy for processing speed because the solution candidates in parameter spaces are not checked well [28]. In order to solve this problem, they propose active search. This method is able to automatically determine whether neighborhood should be checked or not using a similarity of the already checked region. I explain the detail with Fig. 2.5. The template and already checked candidate region are indicated as M and A. The similarity between M and A is S_{AM}. In order to judge whether unchecked region B should be checked or not, an upper limit is calculated using the below equation.

$$S_{UP} = \frac{min(S_{AM}|A|, |A \cap B|) + |B - A|}{|B|} \tag{2.1}$$

The $|\cdot|$ means the number of pixels, and the $|A \cap B|$ represents the logical conjunction between the A and B. If S_{UP} is larger than the currently acquired maximum

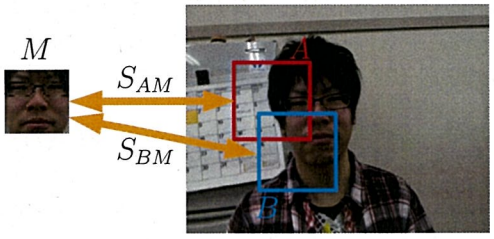


Figure 2.5: The calculation of an upper limit.

similarity, S_{BM} is calculated. Otherwise, it is omitted. By introducing this approach, fast search can be achieved.

Lampert et al. propose a novel method applying branch-and-bound scheme [29]. Generally, a set of candidate regions whose appearances are similar has almost the similar matching scores. Therefore, if one candidate region is judged as that this region does not contain a target object, a set of the candidate regions can be discarded. If one candidate region contain a target object, the set is splitted into small sets and they are checked again. By iterating this procedure, effective search is possible. In experiment, high accuracy and fast speed of the Lampert's method are quantitatively represented.

These related works are high-speed. However, in-plane rotation is not considered. Hence, rotation angle cannot be obtained. In order to acquire the angle, it is necessary to scan iteratively while gradually rotating a target image. Nevertheless, this approach is not effective. In order to solve this problem, this research focuses on the optimization of parameters by metaheuristic algorithms. More specifically, this research addresses more effective search by simultaneously optimizing geometric parameters such as a location, scale, and in-plane rotation.

2.4 Metaheuristic algorithm

As explained in Sect. 2.3, most related works consider the location and scale, however in-plane rotation is not considered. Since rotation angle is necessary depending on applications, this research considers in-plane rotation in addition to the location and scale parameters. Nevertheless, finding optimal parameters which can localize a target object in a target image is a hard problem. This is because there exist enormous solution candidates in each parameter space, and the sum of solution candidates exponentially increases when the combination of these parameters is

considered. Also, since it is necessary to find the optimal solution within a reasonable time limit if practical use is considered, applying a full search method which guarantees an exact solution is impossible. In order to solve this problem, metaheuristic algorithms are often used. They are designed to solve approximately a wide range of hard optimization problems. Also, almost all them have the following characteristics [30].

- Nature-inspired (for example, from physics, biology, and ethology)
- Use of stochastic components (involving random variables)
- No use of the gradient or hessian matrix of the objective function
- Several parameters that need to be fitted to problems at hand

As introduced in [30], a variety of metaheuristic algorithms have been proposed. Among them, this research focuses on population-based metaheuristic algorithms. In this category, there are genetic algorithm (GA) [31], particle swarm optimization (PSO) [32], differential evolution (DE) [33]. I briefly explain these methods.

2.4.1 Genetic Algorithm (GA)

Algorithm 1 Genetic algorithm

$g = 1$
Initialize population $P = \{p_{1,g}, p_{2,g}, ..., p_{N,g}\}$
while (Not end of generation iteration) **do**
  for $i = 1$ to N **do**
    Calculate fitness f_i
  end for
  Selection
  Crossover
  Mutation
  $g = g + 1$
end while

Genetic algorithm is proposed by John H. Holland [31]. The procedure of this method is represented in Algorithm 1. Let g be the generation number. The P is the population, the N is the population size, the i is the individual number, and the f is the fitness. First of all, population P is initialized. Since an individual in simple GA consists of a binary string as chromosomes, the population are initialized by randomly determining the binaries. Next, after the chromosomes of each individual are decoded from the binary string to real values, fitness f is calculated using the decoded real values. After that, selection scheme is performed for the next generation.

There are some selection schemes such as roulette wheel selection, tournament selection, and ranking selection, etc. A comparison of each selection scheme is reported in [34, 35]. This research mainly adopts roulette wheel selection. The reason is that this selection scheme achieves high performance on template matching with GA [36, 37]. In the roulette wheel selection, individuals whose fitness is high are selected with high probability. Therefore, this scheme can be represented using the following equation.

$$s_i = \frac{f_i}{\sum_{k=1}^{N} f_k} \tag{2.2}$$

Let s be selection probability.

After the selection, children are generated by crossover. There are also some crossover methods, such as one point crossover, two point crossover, and uniform crossover. In this research, the uniform crossover is used and this reason is the same to the selection scheme. Figure 2.6 shows the uniform crossover. In this crossover, randomly chosen genes are changed with a selected individual pair.

Finally, mutation, which inverts randomly chosen genes, is performed with a low probability. This processing is applied to prevent premature convergence of the population. By iterating these procedures until a termination condition is satisfied,

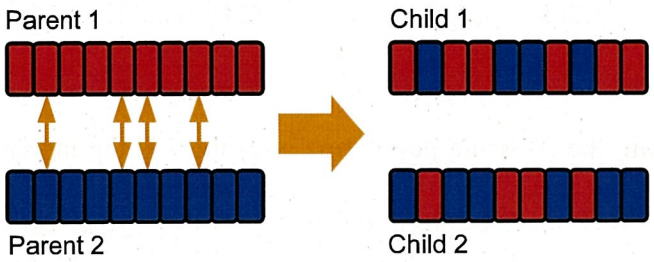


Figure 2.6: Uniform crossover.

the optimal solution can be obtained finally. This is a simple GA algorithm.

2.4.2 Particle Swarm Optimization (PSO)

Particle swarm optimization (PSO) is a global optimization method, which uses the metaphor of the flocking behavior of birds, etc [32]. The procedure is described in Algorithm 2. Each particle P has position vectors X, velocity vectors V, and $pbest$ which is the best position vectors acquired in the search. First of all, P is initialized by randomly determining X and V. Next, fitness f of each particle is calculated. If the f is higher than f^{pbest}, which is the highest fitness obtained in the search, the x^{pbest} is updated. Then, V is calculated to update X of each particle. For the calculation of V, inertial mass ω, is firstly calculated. The ω_{max} and ω_{min} are the maximum and minimum limits of the inertial mass, and the T is the upper limit of the number of iterations of search. These values must be defined by a user in advance. Next, V is calculated. For this calculation, acceleration coefficient c_1 and c_2, uniform random numbers ([0.0,1.0]) ϕ_1 and ϕ_2, and x^{gbest} are required. The x^{gbest} is the position vectors that one particle with the highest fitness has. There are cases where the calculated V is too large or small. In order to prevent this, velocity clamping is applied. This is to restrict the calculated V to a reasonable range $[-V_{max}, V_{max}]$. Generally, [10,20] % of the size of each search space is set as the $[-V_{max}, V_{max}]$ [38]. After the clamping check, the X is updated using the V.

Algorithm 2 Particle swarm optimization

$g = 1$
Initialize position vectors $X = \{x_{1,g}, ..., x_{N,g}\}$ and velocity vectors $V = \{v_{1,g}, ..., v_{N,g}\}$ of particles $P = \{p_{1,g}, ..., p_{N,g}\}$
while (Not end of iteration) **do**
  for $i = 1$ to N **do**
    Calculate fitness f_i
    if $f_i^{pbest} < f_i$ **then**
      $x_{i,g}^{pbest} = x_{i,g}$
      $f_i^{pbest} = f_i$
    end if
  end for
  Store x_g^{gbest}
  $\omega = \omega_{max} - \frac{(\omega_{max} - \omega_{min}) \times g}{T}$
  for $i = 1$ to N **do**
    $v_{i,g} = \omega v_{i,g} + c_1\phi_1(x_{i,g}^{pbest} - x_{i,g}) + c_2\phi_2(x_g^{gbest} - x_{i,g})$
    if $v_{i,g} < -V_{max}$ **then**
      $v_{i,g} = -V_{max}$
    end if
    if $V_{max} < v_{i,g}$ **then**
      $v_{i,g} = V_{max}$
    end if
    $x_{i,g+1} = x_{i,g} + v_{i,g}$
  end for
  $g = g + 1$
end while

This is the PSO algorithm.

2.4.3 Differential Evolution (DE)

Recently, differential evolution (DE) [33] is researched very well since this method achieves high performances on optimization competitions. The procedure is described in Algorithm 3. First of all, population P is initialized by randomly determining vectors X. The X is a real value. Next, fitness f is calculated. After that, mutant vectors x^u are calculated by mutation. By using the calculated mutant vectors and crossover, a child v is generated. Then, the fitness of the child f^v is calculated, and parent's vectors are replaced with the child's vector x^v if $f < f^v$. By iterating this procedure, a global optimum can be acquired finally.

Algorithm 3 Differential evolution

$g = 1$
Initialize vectors $X = \{x_{1,g}, ..., x_{N,g}\}$ of population $P = \{p_{1,g}, ..., p_{N,g}\}$
while (Not end of iteration) **do**
  for $i = 1$ to N **do**
    Calculate fitness f_i
  end for
  for $i = 1$ to N **do**
    Generate a mutant vector $x^u_{i,g}$ by mutation
    Generate a child $v_{i,g}$ by crossover
    Acquire fitness of the child f^v_i
    if $f_i < f^v_i$ **then**
      $x_{i,g+1} = x^v_{i,g}$
    else
      $x_{i,g+1} = x_{i,g}$
    end if
  end for
  $g = g + 1$
end while

Here, the details of the mutation and crossover are explained. There are some types of mutation and crossover methods and they are represented using the following notation, $DE/A/B/C$. The A means the selection method of a target vector, the B represents the number of difference vectors, and C is the crossover method. Hence, $DE/rand/1/bin$ means that a target vector is randomly selected ($x_{r_1,g}$), one difference vector is calculated ($x_{r_2,g} - x_{r_3,g}$), and binomial crossover is applied. The binomial crossover is explained later. The r means the randomly chosen individual numbers and $r_1 \neq r_2 \neq r_3$. Since a scale $F_{i,g}$ is usually applied to the differencee vector, the equation to generate a mutant vector ($u_{i,g}$) with the $DE/rand/1/bin$ is represented as below.

$$u_{i,g} = x_{r_1,g} + F_{i,g}(x_{r_2,g} - x_{r_3,g}) \tag{2.3}$$

In the mutation strategy, there are also $DE/rand/2$, $DE/best/1$, $DE/best/2$, $DE/current\text{-}to\text{-}rand/1$, $DE/current\text{-}to\text{-}best/1$, etc.

Algorithm 4 Binomial crossover

$j_{rand} = rand\_int[1, D]$
for $j = 1$ to D **do**
  if $rand\_double[0.0, 1.0] < C_{i,g}$ or $j == j_{rand}$ **then**
    $x_{i,g}^{v,j} = x_{i,g}^{u,j}$
  else
    $x_{i,g}^{v,j} = x_{i,g}^{j}$
  end if
end for

After the mutant vector is calculated, the child v is generated by the crossover. There are some crossover methods, such as binomial crossover and exponential crossover, etc. Here, the binomial crossover is described in Algorithm 4 as an example. Firstly, a variable number j_{rand} is randomly selected. The D is the number of considered variables. Next, a random number of [0.0,1.0] is generated, and if the value is smaller than $C_{i,g}$ or $j == j_{rand}$, the mutant vector $x_{i,g}^{u,j}$ is used as the child's vector $x_{i,g}^{v,j}$.

2.5 Template Matching with a Metaheuristic Algorithm

As described in Sect. 2.3, some effective object search methods are proposed. They consider location and scale of a target object, however in-plane rotation is not considered. This means that the rotation angle of the target object in an image cannot be obtained. In order to acquire the angle, it is necessary to adopt other algorithms or sliding window while gradually rotating the target image. Nevertheless, this approach is ineffective. For more effective search, in-plane rotation parameter should be simultaneously optimized in addition to the location and scale. Of course, the optimization of these geometric transformation parameters is not easy task. The reason is that each parameter has a search space and there exist enormous solution

candidates. When the combination of these parameters is considered, the number of solution candidates exponentially increases. Moreover, when the practical use is considered, this combinatorial optimization problem must be solved within reasonable time limit.

In order to address this problem, this research uses a metaheuristic algorithm. This algorithm is able to simultaneously optimize multiple parameters. Hence, by optimizing the geometric transformation parameters to localize a target object in an image, the target object can be searched effectively. The below equation represents the geometric transformation when the location, scale, and in-plane rotation of a target object are considered.

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{2.4}$$

Each matrix from the left side represents the transformation matrices of the parallel transformation, in-plane rotation, and scale. By simultaneously optimizing the five parameters, t_x, t_y, θ, s_x, and s_y, in each matrix using a metaheuristic algorithm, a target object can be localized. This is the template matching with a metaheuristic algorithm.

2.5.1 Performance Comparison between GA and PSO on Template Matching

As explained in Sect. 2.4, there are a variety of metaheuristic algorithms. It is necessary to investigate which method can achieve good performance on template matching. We have already investigated this in [39], and its content is described here.

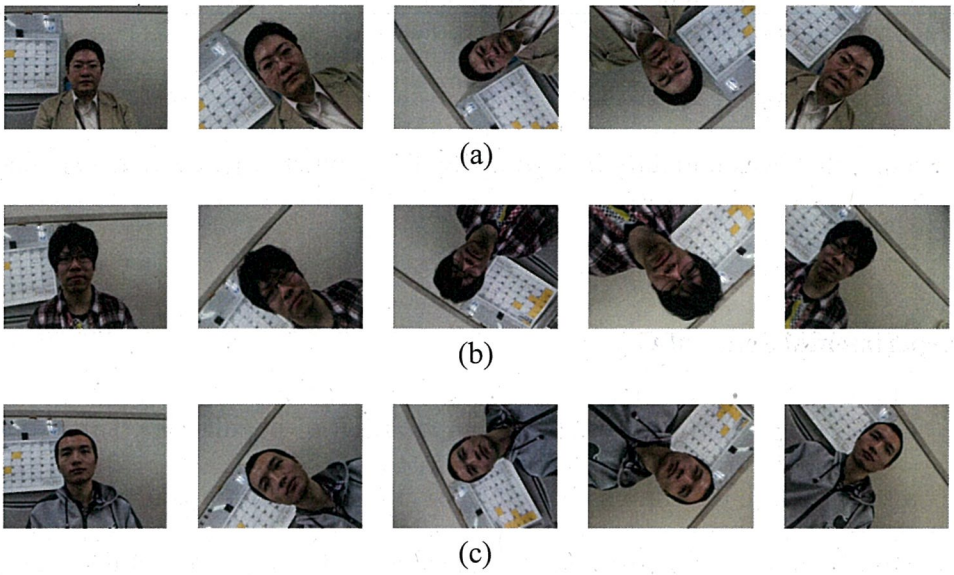


Figure 2.7: Examples of a target sequence: a) sequence 1; b) sequence 2; c) sequence 3.

For the comparative experiments, three target video sequences were created (Fig. 2.7). All the sizes are 320 × 240 pixels. A webcamera was used to record. The distance between the camera and a face was approximately 50 cm when the recording started. The camera was brought close to the face while rotating manually with constant speed. When the distance between the camera and the face was approximately 30 cm, the camera was moved away from the face to 50 cm. This movement was repeated two times by one rotation.

Templates are acquired from the initial frame in each sequence (Fig. 2.7). All the templates include eyebrows, palpebral fissures, nose, and mouth completely. For the reduction of the calculation cost, the templates were scaled down to 1/4.

Figure 2.8: Templates: a) 29 × 29 pixels; b) 33 × 33 pixels; c) 30 × 30 pixels.

The interpolation method is bicubic interpolation.

The performances of GA and PSO are evaluated based on whether they can keep detecting the faces correctly by optimizing the geometric parameters explained in Sect. 2.5.

Experimental Setup of GA

Genetic algorithm firstly requires initialization of population. This is always necessary on an initial frame. Nevertheless, the initialization is unnecessary from the next frames. This is because the appearance of a target object does not drastically change on consecutive frames, and the local search is enough. Hence, by inheriting the converged population at final generation in a current frame to the next frame as the initial population, the local search can be achieved. This idea is proposed as evolutionary video processing [40] and good performances are reported [37, 36]. Therefore, GA introduces this approach in this comparison.

Genetic algorithm also introduces elitism. The elitism saves an elite individual, whose fitness is the highest in each generation, and inherits to the next generation. Since there are cases where elite fitness decreases by generation iteration, the elitism is introduced to prevent it.

In comparative experiments, GA optimizes parameters of parallel translation (x, y), scales (s_x, s_y), and in-plane rotation (θ). Since each parameter is coded as eight bits based on my previous work [36], the chromosome length of individuals is 40 bits. The set crossover and mutation probabilities are 0.7 and 0.05. The sizes of population and generation iteration are described later.

Experimental Setup of PSO

As explained in Sect. 2.4.2, it is necessary to set values of parameters for PSO in advance. Each set value is represented below.

- Acceleration coefficient (c_1, c_2): 2.0
- Uniform random number (ϕ_1, ϕ_2): [0.0,1.0]
- Upper and lower limit for inertia weight (ω_{max}, ω_{min}): 0.9 and 0.4
- The number of iterations (T): 100

Different values have an influence on the search performance. Nevertheless, finding optimal parameters is not main objective in this experiments, hence the parameter investigation is future work.

Similar to GA, particle swarm of PSO at final iteration in a current frame is inherited to the next frame as an initial particle swarm. However, PSO does not have an algorithm such as mutation of GA to prevent premature convergence. This causes detection failures and they are confirmed in preliminary experiments. In order to solve this problem, Algorithm 5 is introduced.

Let N be the swarm size. The D is the number of variables, and $D = 5$ in this comparative experiments since the five geometric transformation parameters are optimized. Each x, v, x^{pbest} is the position vector, velocity vector, and $pbest$.

Algorithm 5 Mutation for PSO.

```
for i = 1 to N do
  if i != gbest then
    for j = 1 to D do
      if rand_double[0.0, 1.0] <= 0.07 then
        Initialize x_i^j, v_i^j, and x_i^{pbest,j}
      end if
    end for
  end if
end for
```

The threshold 0.07 was empirically determined. By generating uniform random numbers and using them, the vectors are initialized.

In order to suppress generation of too small or large velocity vector, velocity clamping is introduced. Generally, [10,20] % of the size of each search space is set [38]. In the experiments, three different ranges, 10%, 15%, and 20%, are adopted and compared.

Other Experimental Setup

Since the both of GA and PSO depend on a random number, different ten random seeds were tried. The average value of all the results was used for the evaluation. The number of used swarm sizes was 10, 30, and 50. The population sizes of GA were 10, 31, and 52. The reason why these sizes were set is that the difference of the detection accuracy is compared by adjusting processing time as much as possible. Each search range of the optimized geometric transformation parameter is represented below.

- Parallel translation (x,y): [0,319] and [0,239]
- Scales for x and y axes: Different scales depending on target sequences
- In-plane rotation (θ): [0,360]

The adopted fitness function for GA and PSO is based on sum of absolute difference (SAD). Equation (2.5) represents the fitness function.

$$f = 1.0 - \frac{\sum_{y=1}^{h}\sum_{x=1}^{w}|p'_{x,y} - p_{x,y}|}{255 \times w \times h} \tag{2.5}$$

Let f be the fitness. The x and y are the coordinate, and the w and h are the width and height of the template. The p' and p mean that pixel value of a template and a

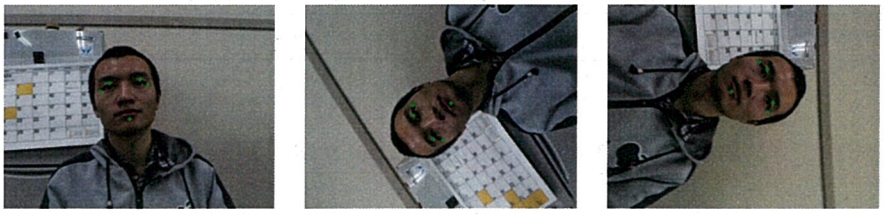


Figure 2.9: Manually set evaluation points.

target image.

In order to judge the result, we set two criteria. The first criterion is that when a detection bounding box, whose size is increased to 110 %, includes all the five evaluation points, the result is judged as success. Figure 2.9 shows the examples of the evaluation points. They are manually set. Each point indicates the upper eyelids, inner corners of eyes, and bottom of mouth. When it is difficult to set the point due to occlusion of hair, the point is predictively set. The second criterion is the judgment of the in-plane rotation. There is a possibility that the result is judged as success even though the rotation angle is not correct. In order to remove such case, the distances between the upper left and right vertexes of the detection bounding box, and the evaluation point of the bottom of mouth are calculated. Also, the distances between the upper left and right vertexes of the detection bounding box, and the evaluation points of the both upper eyelids are calculated. If the distances to the upper eyelids are shorter than the distance to the bottom of mouth, the result is judged as success.

The computer which was used has 3.2 GHz CPU and 4 GB physical memory.

Experimental Results and Consideration

From Table 2.1 to 2.4 show the detection accuracy and processing time per one frame of GA and PSO. When all the accuracy of PSO with different velocity limitations are compared, 20 % is the best. Therefore, this percentage is the most

Table 2.1: Results of GA.

The number of individuals	10	31	52
Accuracy (%)	94.5	96.4	97.7
Processing time (millisecond)	36.8	111.0	185.0

Table 2.2: Results of PSO. The velocity limitation is 10 %.

The number of particles	10	30	50
Accuracy (%)	88.2	94.7	96.5
Processing time (millisecond)	38.7	114.4	187.1

Table 2.3: Results of PSO. The velocity limitation is 15 %.

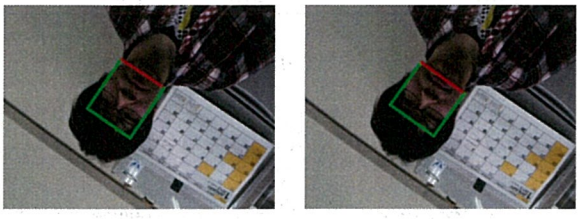
The number of particles	10	30	50
Accuracy (%)	88.5	97.0	97.4
Processing time (millisecond)	37.9	111.8	187.4

Table 2.4: Results of PSO. The velocity limitation is 20 %.

The number of particles	10	30	50
Accuracy (%)	91.1	97.2	97.9
Processing time (millisecond)	37.4	111.0	186.2

suitable percentage for template matching. When the population and swarm sizes are 10, the accuracy of GA is higher while the accuracy of PSO with 30 and 50 is higher. This reason is that different from GA, PSO is better at addressing continuous problems such as video processing since the particles search based on directional vectors. On the other hand, GA does not consider the directional vectors since individuals are encoded as binary. This algorithm difference causes the accuracy difference.

By obtaining optimized geometric transformation parameters, sensing of the target object in images is possible. Figure 2.10 shows the detection results by GA and PSO. The values of the optimized parameters to acquire the detection bounding box are represented in Table 2.5. This result indicates that the both methods obtain the similar sensing information. Different from the search methods enumerated in Sect. 2.3, metaheuristic algorithms are able to acquire the in-plane rotation easily in addition to the location and scale. Therefore, this approach is more effective.



(a) (b)

Figure 2.10: Used frames to obtain sensing information: a) GA; b) PSO.

Table 2.5: Sensing result of Fig. 2.10.

Information		GA	PSO
Coordinate	x	145	144
	y	111	111
Scale factor	x	1.8	1.9
	y	2.1	1.9
Angle of rotation		213	215

Finally, result examples by GA and PSO are shown in Fig. 2.11 and 2.12. The both methods accurately detect the face. This is the technique, which is mainly adopted to all the researches in this thesis. By applying this technique, a variety of tasks are addressed. The following each chapter describes the details.

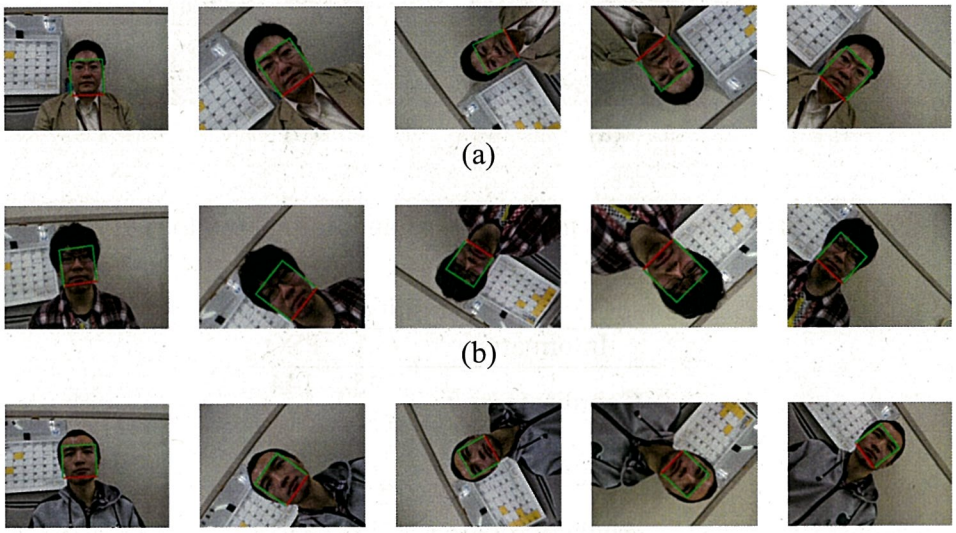


Figure 2.11: Result examples by GA.

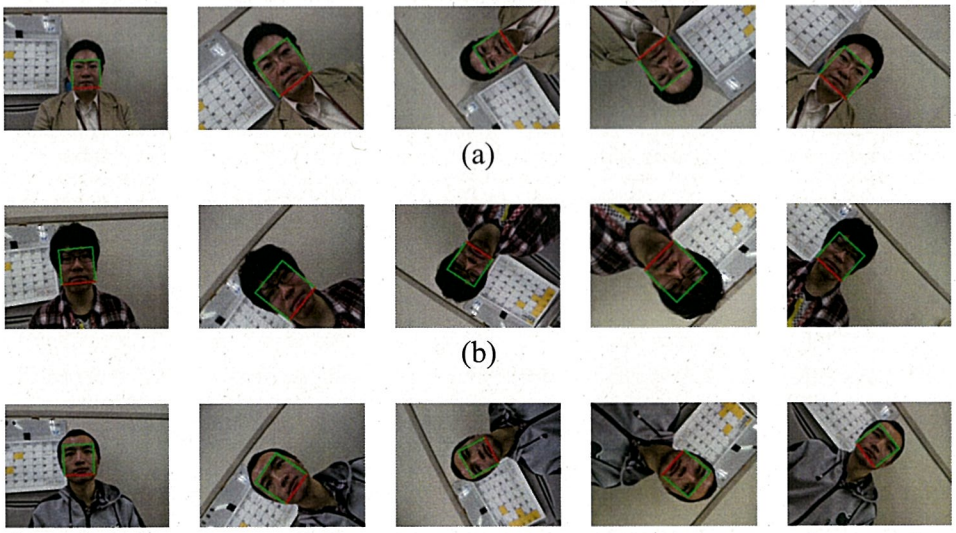


Figure 2.12: Result examples by PSO.

Chapter 3

High-speed Multiview Face Detection and Tracking Using Effective Template Generation and Search by GA

In this chapter, I explain the high-speed multiview face detection and tracking using effective template generation and search by GA. This research is published as "high-speed multiview face localization and tracking with a minimum bounding box using genetic algorithm" in [36].

In order to detect and track a multiview face, the following ideas are introduced. For example, a head is treated as a cylinder and the multiview face can be represented using the development of lateral surface (2D model). The face can be localized by a template, which is generated from the model and corresponds to the target face direction. Processing is very fast because parameters for both template generation and affine transformation are simultaneously optimized by GA. In the experiment, challenging 60 video sequences are created in a situation where sub-

jects drastically move their faces in a room using a standard computer and web camera. Then, the proposed method is applied to the sequences and the performance is investigated. As a result, the proposed method achieves fast and accurate multiview face localization and tracking.

3.1 Introduction

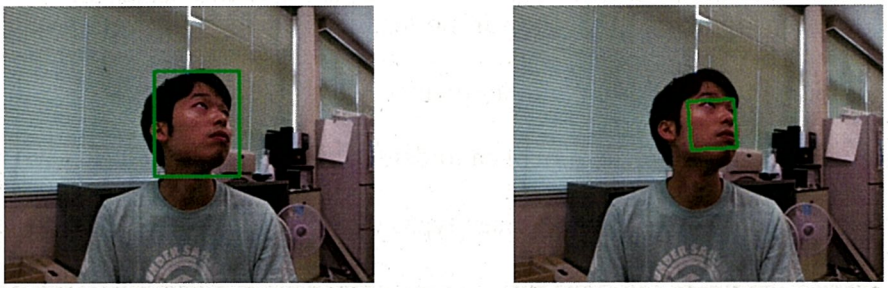
A human face has much information that is useful for many systems employing cameras. For example, there is an eye detection method for detection of sleeping at the wheel [41] and lip detection and reading methods for an interface and communication systems [40, 42]. Generally, these methods firstly localize a face, and set a ROI to obtain eyes and lip information. Therefore, accuracy and localization speed are important for practical use. However, there are some difficulties: 1) Each photographic environment is different, 2) Parameters of cameras are different in the real world, and 3) Face direction is not always frontal. In order to overcome these problems, many methods have been proposed. In particular, many methods treat the face detection problem as a binary decision one, with Viola's method [43] being one of the most famous to so. In recent years, many methods using deep learning have been proposed [44, 45]. These methods are based on prior machine learning and this approach is currently the standard one. However, there are some problems: 1) Collecting and creating datasets are hard work [46], 2) Learning does not always converge [47]- therefore, empirical parameter tuning is necessary, and 3) The learning time is long [48].

In order to avoid these problems, we treat face localization not as a binary decision problem but as an optimization problem in this research. Because our research considers applying to the systems described above in the future, one precondition, where the number of users is one, can be set. In other words, one face always ap-

pears in each frame and the face can be localized using optimization methods by setting the affine transformation parameters, which localize the face, as the global optimum. Nevertheless, the sum of candidate solutions, which exist in each parameter space, is enormous. Because seven types of parameters are used in this research, the sum of candidate solutions further increases. This is a combinatorial optimization problem, which makes it difficult to search for the global optimum in real-time. However, evolutionary computation methods including GA [49] can solve the problem efficiently. By focusing on this advantage and applying it to template matching, we deal with multiview face localization and tracking. Because candidate regions are generated by chromosomes of each individual, the scale and rotation of regions can be adaptively adjusted by generation iteration, unlike sliding window. In other words, since candidate regions are invariant with the scale and rotation, the labor for feature design can be reduced.

Our considering applications are based on video processing while also proposed method must be fast. Because sliding window, which is often used in many methods, iteratively scans the whole image, this method is not suitable for video processing. After the face is globally searched and localized in the initial frame, locally searching and tracking it from the second frame is more efficient. Of course, the sliding window can achieve this. However, if scale and rotation are considered, the computational complexity increases, so this approach is still inefficient. On the other hand, GA can achieve what is needed. Besides, the population can automatically and globally search when tracking is lost. Hence, template matching with GA has some advantages. Based on this approach, accomplishing multiview face localization and tracking with high-speed is our main objective in this research.

The size of the detection bounding box according to recent methods [50], which are based on prior machine learning, tends to be large (Fig. 3.1(a)). However, the



(a) (b)

Figure 3.1: Difference in the size of the bounding box: (a) a recent face detection method [50]; (b) proposed method.

size should be as small as possible because a large bounding box has noises, such as background. Hence, we also address localization and tracking problems with a minimum bounding box (Fig. 3.1(b)) in this research. Note that the meaning of the minimum bounding box is the minimum region surrounding face parts such as palpebral fissures, nares, and a mouth, but, as much as possible, has no noises such as hair and background.

In the experiment, a challenging 60 video dataset in which subjects drastically move their faces with various backgrounds was created. In each frame, only one face appears. In addition, any illumination changes and occlusions are not included since the proposed method will be applied to the systems, which are used in rooms. The proposed method and comparative methods are applied to the created dataset, and the performances are compared.

Particle filter (PF) [51] can be applied to this research since the structure is similar to GA. However, PF has more pre-determined parameters. For example, the ranges of velocity and noise vectors for all the state vectors must be determined in advance if the linear and non-Gaussian model is used. Because this is hard work and GA does not require these settings, this research focuses on GA.

3.2 Proposed Method

3.2.1 Basic Idea

A human head can be approximated using an ellipsoid or a cylinder [52, 53]. Development of cylinder's lateral surface can be represented from a profile to frontal face using only one 2D model. Hence, the multiview face can be localized using a template, which is generated from the model and corresponds to the face direction with yawing. The rotated (in-plane) face can be localized by rolling the model. Also, appearance change with pitching can be represented by changing distance of facial parts of the model. For example, the distance between palpebral fissures and nares is short when face direction is upward. The distance between nares and mouth is short when face direction is downward. This change can be represented easily using the 2D model. Thus, there is an advantage as calculation cost is small because the 3D appearance change can be represented by the 2D model.

There are some problems about the above ideas. For instance, after all patterns of the templates are generated, all matching scores must be calculated in a whole target image, since prediction of face direction in advance is difficult. This approach takes a long time. Other problems are generalities of the 2D model and features. They must be considered for all user's faces. In this section, we explain a solution for these problems. Note that the proposed method simulates the 2D appearance changes by the 3D head pose changes but the method does not estimate the 3D head pose. The outline and the pseudo code are described in Fig. 3.2 and Algorithm 6. The detail is explained in the following sections.

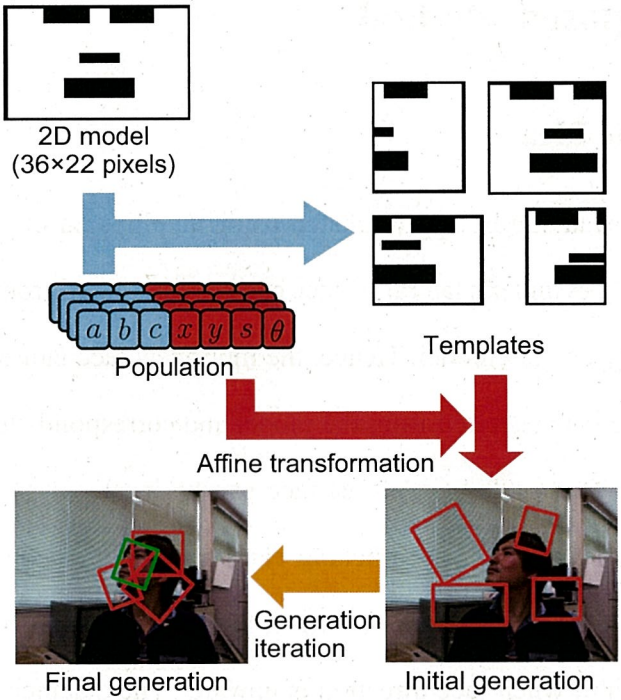


Figure 3.2: Outline of the proposed method. Red bounding box indicates the face candidate region. Green bounding box indicates the elite individual and localization result.

Algorithm 6 Proposed method

```
Input 2D face model
Generate population for GA
while (Not end of a video frame) do
  Input one video frame
  Create binary and skin region extracted images
  while (Not end of a generation iteration) do
    Create templates from the model using the population
    Calculate matching scores (fitness)
    Selection, crossover, and mutation
  end while
  Obtain the elite individual
  Display the detection result of the elite individual
end while
```

3.2.2 2D Model

As described in the previous section, a human head can be approximated using a cylinder. Development of lateral surface is a rectangle and the rectangle model is created as shown in Fig. 3.2. The model is a binary image. The black regions

indicate palpebral fissures, nares, and a mouth. The eyebrows are not used because there are cases where they are occluded by frontal hair. Since this research focuses on high-speed, the created model is simple and small (36×22 pixels). Of course, many models with varied appearances can be created by changing the shape, width, and height of regions, which represent the model and the facial parts. We should consider these models, however we do not discuss them because this paper focuses on a localization and tracking possibility of the multivew face by optimization.

In order to detect a face using the 2D model, parameters for template generation and for face candidate regions are necessary. The proposed method localizes the face by obtaining these optimized parameters. The details of these and the optimization method are explained in the following sections.

3.2.3 Template Matching with Genetic Algorithm (GA)

For multiview face localization, a variety of methods can be applied. In particular, detector based approaches are popular; however, generating a trained model is hard work and the calculation cost of sliding window is large. Therefore, an optimization based approach is adopted in this research.

This approach can localize the face region by setting the affine transformation parameters that indicate the face region as the global optimum and the function to calculate the matching score as the objective function. Nevertheless, optimizing the seven parameters, which are for affine transformation (x, y, s, and θ) and template generation (a, b, and c) as shown in Fig. 3.2, is difficult. This is because the number of candidate solutions in each parameter is large and the sum of combined candidate solutions is even larger. Evolutionary computation methods including GA can solve this combinatorial optimization problem efficiently. Hence, we address this problem by constructing an algorithm, which can localize the multiview face at high-speed,

by applying GA in this research. The applied GA is the binary coded GA [49]. The reason why this GA is used is that the speed of the crossover and the mutation is faster than the real coded GA.

We explain how to localize the face using GA and model. Templates are generated from the model using chromosomes, a, b, and c of the individuals. Figure 3.3 represents the procedure. First of all, a pixel range $[a, b]$ is determined by the chromosome. If $c \leq 0$, the region under the horizontal line of the upper nares in the template shifts up $|c|$ pixels. Otherwise, the region over the horizontal line of the lower nares shifts down c pixels. By processing in this way, the template is generated. Figure 3.3 is an example of $(a, b, c) = (11, 33, -4)$. The number of generated templates is the same to the population size because the templates are generated using chromosomes of each individual. Next, rectangles, with a size the same as each generated template, are transformed by the parameters of x, y, s, and θ. The located rectangles in a target image are called candidate regions and each matching score is calculated using the fitness function. The detail of the function is described in Sect. 3.2.3. The calculated matching score is used as the fitness in template matching with GA.

After all the fitness scores are acquired, genetic operations, selection, crossover, and mutation are performed to generate next population. In this research, the roulette

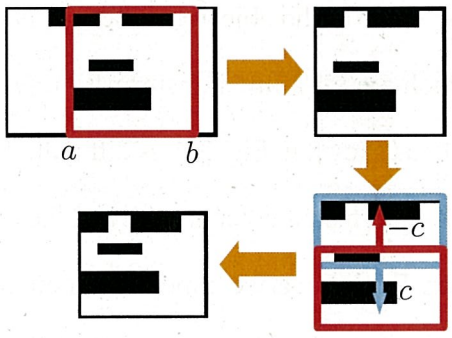


Figure 3.3: Example of template generation.

wheel selection and the uniform mutation are used. The roulette wheel selection is a process to select the individuals based on the following equation.

$$P_i = \frac{f_i}{\sum_{k=1}^{N} f_k} \tag{3.1}$$

Let P and i be the selection probability and the individual number. The f and N are the fitness and the population size. In this selection method, the probability of individuals with high fitness being selected is high. Next step is crossover. Uniform crossover is adopted and assigns the genes of the parents to the offspring. The parent's gene to be copied at each crossover point is randomly determined. Uniform crossover can generate various types of chromosomes and diversity can be maintained [54]. Last operation is mutation. The probability of mutation randomly inverting the genes is low. This operation is also applied to maintain diversity. By iterating these genetic operations, the population for the next generation is produced and the parameters are updated. This GA has already been applied to an eye tracking method [41] and a lip detection method [40], and high detection accuracy is reported. Therefore, these operations are also effective in this research.

The proposed method introduces evolutionary video processing [40]. This processing initializes the population just once at the first generation in an initial frame, and inherits them from the final generation to the next frame as the initial population. The reason why this processing is used is that the search is effective because the face appearance does not drastically change in consecutive frames. Also, this processing can reduce calculation cost for initialization.

Fitness Function

In this research, the fitness is calculated in each candidate region by counting the number of pixels, which represent the black in the binarized image and skin

region in the skin region extracted image. The binarized image is used because face parts, such as palpebral fissures, nares, and a mouth, can be extracted as black pixels. The reason why the skin region is focused on is that most backgrounds can be removed. Since the binarized image and the skin region extracted image can be treated as binary, calculation cost is reduced, which is an advantage. By combining these features, the multiview face can be localized. However, we noticed the generated template size tends to be small in the preliminary experiment. For instance, the template representing the profile face localizes half of the frontal face. In order to suppress this tendency, information about width and height of the generated templates is introduced. The designed fitness function is represented in Eq. (3.2).

$$f = r_{skin} + r_{parts} + 0.5p_{skin} + 0.5(r_{width} + r_{height}) \tag{3.2}$$

$$r_{skin} = m/s^i_{skin} \tag{3.3}$$

$$r_{parts} = e/s^i_{parts} \tag{3.4}$$

$$p_{skin} = 1.0 - e/s^i_{skin} \tag{3.5}$$

$$r_{width} = w^i/W_{model} \tag{3.6}$$

$$r_{height} = h^i/H_{model} \tag{3.7}$$

The fitness function consists of a reward r and a penalty p, and they are normalized to [0.0,1.0]. The r_{skin} represents the proportion of the number of white pixels in the skin region of the template. The m and the s^i_{skin} in Eq. 3 mean the number of white pixels in the skin region of the template and the area of the skin region of the template, which is generated by the ith individual. The r_{parts} is the proportion of the number of black pixels in the face parts regions of the template. The e and s^i_{parts} in Eq. 4 are the number of black pixels in the face parts regions of the template and the area of the face parts regions of the template, which is generated by the ith indi-

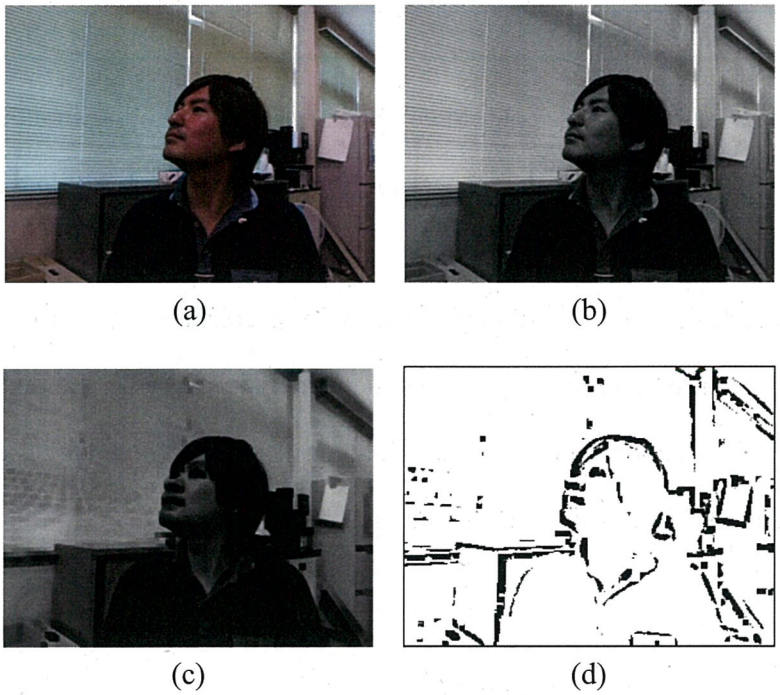


Figure 3.4: Preprocessing for a binarization: (a) color; (b) gray scale; (c) twice grayscale erosions; (d) binarization using an adaptive threshold.

vidual. The p_{skin} represents the proportion of the number of black pixels in the skin region of the template (Eq. 5). The r_{width} and the r_{height} are the proportion of the width and the height of the template to the width and the height of the model. The w^i and the W_{model} in Eq. 6 represent the width of the generated template by the ith individual and the width of the model. The h^i and the H_{model} in Eq. 7 represent the height of the generated template by the ith individual and the height of the model. The constant values, 0.5, are the weight, which are determined empirically. They are necessary since no weights cause a localization error in uniform background due to the influence of the p_{skin}. Also, too large a template tends to be generated by the influence of the r_{width} and the r_{height}. The weights can reduce these failures.

3.2.4 Preprocessing

After a video frame is input, binary image and skin region extracted image are created. Figure 3.4 shows the generation process for the binary image. Firstly, a target color image (Fig. 3.4(a)) is converted to a gray scale image (Fig. 3.4(b)). Secondly, grayscale erosions are performed twice to enhance face parts (Fig. 3.4(c)). The 3×3 kernel is used and replaces the centered pixel with the minimum pixel value in the neighborhood. After that, the binary image is generated using an adaptive threshold (Fig. 3.4(d)). Thresholding is performed based on the following equations.

$$B(x,y) = \begin{cases} 0 & \text{if } p(x,y) \leq t \\ 255 & \text{otherwise} \end{cases} \tag{3.8}$$

$$t = \left\{ \frac{1}{11 \times 11} \sum_{j=-5}^{5} \sum_{i=-5}^{5} p(x+i, y+j) \right\} - v \tag{3.9}$$

Let B be the binarized pixel and the target pixel at (x, y) is denoted as $p(x, y)$. The t is the threshold, which is calculated in a neighborhood of 11×11 pixels. The neighborhood size affects the line width of the edges after the binarization. The smaller size is the thinner line width. Because the area ratio between face parts and skin after binarization should be the same to the model, 11×11 is set based on the preliminary experiment. The v is a constant value. The smaller v is, because more black pixels appear as noise. Since much noise affects the accuracy, $v = 10$ is used based on the preliminary experiment. These fixed parameters are easily affected by large illumination change. However, these parameters can be applied because this research supposes that there is no illumination change, as mentioned in Sect. 3.1. The same values are used in all the experiments.

Next, the procedure to create a skin region extracted image is explained. In this

research, the method of Chai et al. [8] is used. This method can extract the face region from a head-and-shoulders view even if background is cluttered. Also, this method does not use prior machine learning. Hence, it is suitable to our concept. We briefly explain the algorithm. There are five stages in this method. At the first stage, the candidates of the skin region are extracted. After an input image is converted to YCrCb color space, the Cr and Cb components are obtained. Next, candidate pixels are extracted based on the following equation.

$$C(x,y) = \begin{cases} 1 & \text{if } (Cr(x,y) \in R_{Cr}) \cap (Cb(x,y) \in R_{Cb}) \\ 0 & \text{otherwise} \end{cases} \tag{3.10}$$

The $C(x,y)$ indicates whether a pixel at (x,y) is a candidate or not. The $Cr(x,y)$ and $Cb(x,y)$ represent the pixel values in the Cr and Cb images. The R_{Cr} and R_{Cb} indicate the ranges of the pixel value, [133,173] and [77,127]. The extracted candidates are shown in Fig. 3.5(a). At the second stage, the pixels are classified into three classes ($D(x,y)$), non-facial region, intermediate, and facial region, using the following equations.

$$D(x,y) = \begin{cases} \text{facial region} & \text{if } d(x,y) = 16 \\ \text{intermediate} & \text{if } 0 < d(x,y) < 16 \\ \text{non-facial region} & \text{if } d(x,y) = 0 \end{cases} \tag{3.11}$$

$$d(x,y) = \sum_{j=0}^{3} \sum_{i=0}^{3} C(4x+i, 4y+j) \tag{3.12}$$

Note that where $x = 0, \ldots, W_{target}/4 - 1$ and $y = 0, \ldots, H_{target}/4 - 1$. The W_{target} and H_{target} are width and height of target image. Through this processing, the resolution is reduced to 1/16 (Fig. 3.5(b)). Next, the following three steps are applied. 1) All the $D(x,y)$, which correspond to the edge of the image, becomes zero

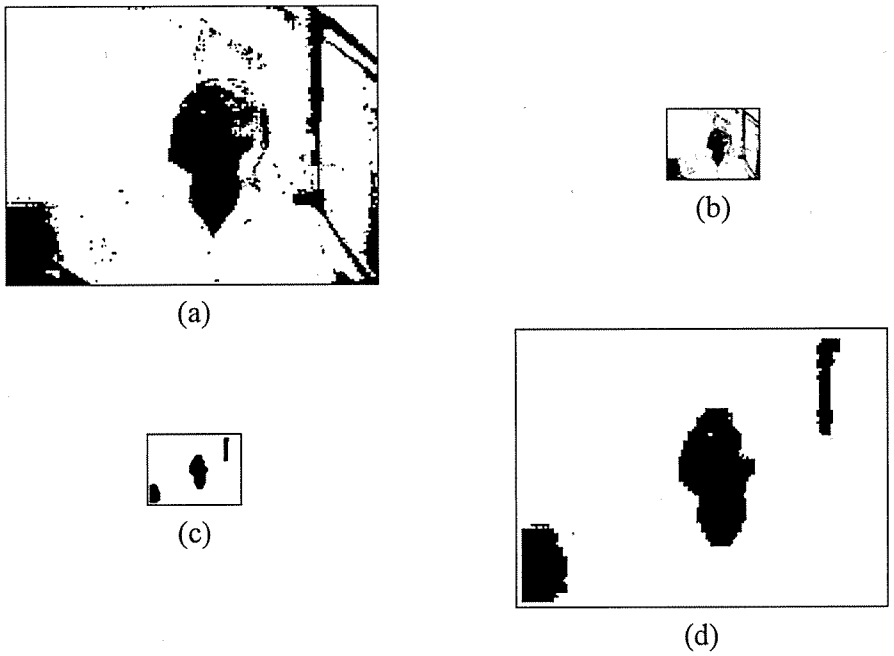


Figure 3.5: Preprocessing for skin region extraction: (a) extraction of the skin region candidates (stage one); (b) classification of the candidates (stage two). Facial region, intermediate, and non-facial region are represented by black, gray, and white pixels; (c) geometric correction (stage four); (d) reconstruction of the resolution (stage five).

(non-facial region). This means that $D(0, y) = D(W_{target}/4 - 1, y) = D(x, 0) = D(x, H_{target}/4 - 1) = 0$. 2) While the 3×3 neighborhood scans the whole image and $D(x, y)$ becomes zero when the number of $d = 16$ in the neighborhood is less than five. 3) The 3×3 neighborhood scans the whole image and $D(x, y)$ becomes 16 when the number of $d = 16$ in the neighborhood is greater than two. At the third stage, the pixels with uniform intensities are eliminated as background. However, this stage is not used in this research because results are detrimental. At the fourth stage, the number of connected pixels of horizontal and vertical directions is checked. If the number is lower than the threshold, the pixels are eliminated as noise (Fig. 3.5(c)). At the fifth stage, the resolution of the image in the fourth stage is reconstructed by using the image in the first stage (Fig. 3.5(d)).

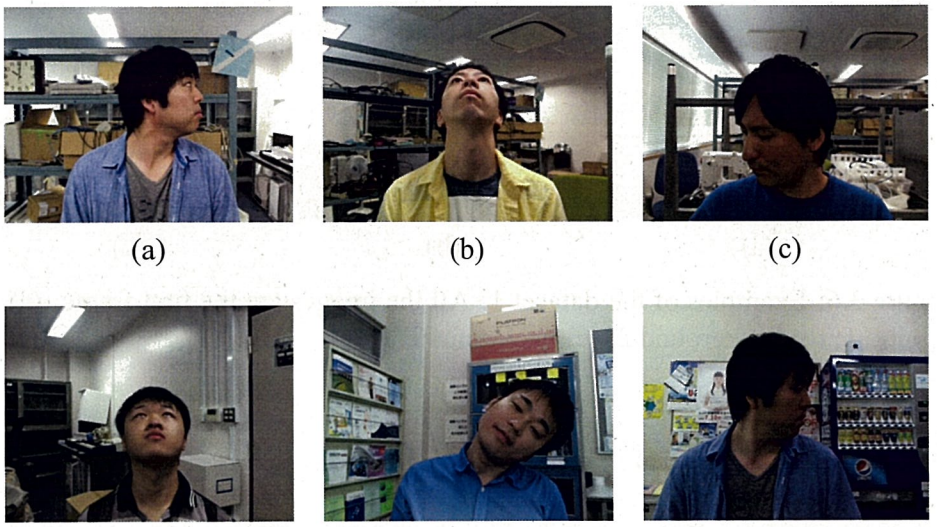
(a) (b) (c)

Figure 3.6: Examples of the dataset 1. All the images are the first frame: (a) background 1; (b) background 2; (c) background 3.

3.3 Experiment

3.3.1 Creation of a Dataset

This research considers applications to low cost systems, such as those used in rooms with feeding devices for orthopedically-impaired people [37]. Hence, video datasets in which chest-to-head area of one subject appears are necessary. Many face datasets exist; however, there is no suitable dataset for our objective, as far as we know. For example, AFLW [55] and FDDB [52] are famous datasets. However, they are not video data. Biwi kinect head pose database [56] is close to the considered environment since this is a video dataset with chest-to-head of one subject appearing. Nevertheless, the change of the face direction and the number of backgrounds are small. Thus, there is no applicable dataset and so we created a new dataset. We firstly took videos of ten subjects with three different backgrounds (dataset 1 and Fig. 3.6). Moreover, another dataset, which contains more various backgrounds, was created (dataset 2 and Fig. 3.7). The number of subjects in dataset 2 is five, with four of them being the same as dataset 1 and the other is a new subject. The number of background types is six. Because the applications to low cost systems are considered, a cheap web camera, HD pro webcam C910 of Logicool, was used with 30 fps. In both datasets, the first face direction is arbitrary and the subject freely moves face by the end. Speed of movement is not restricted,



(a) (b) (c)

(c) (d) (e)

Figure 3.7: Examples of the dataset 2. All the images are the first frame: (a) background 4; (b) background 5; (c) background 6; (d) background 7; (e) background 8; (f) background 9.

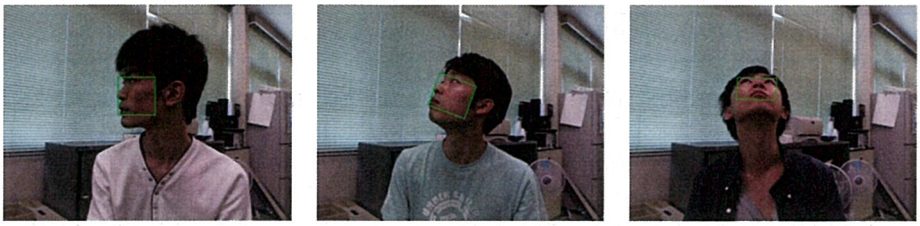
(a) (b) (c)

Figure 3.8: Example of a sequence: (a) initial frame; (b) 93th frame; (c) 186th frame.

however, to avoid blur, it is not too fast. The created dataset can be opened and downloaded at our project web page [57]. Figure 3.8 shows an example of face direction change in one sequence.

3.3.2 Creation of Ground Truth

Ground truth is created for evaluation and its data also can be opened at our web page [57]. The ground truth is created based on the following rules. If the face is only pitching or rolling, the upper side of the ground truth passes both upper sides



(a) (b) (c)

Figure 3.9: Examples of ground truth.

of the palpebral fissures. If the face is yawing, the upper side of the ground truth passes the upper side of one palpebral fissure, the one that has a smaller y coordinate. Because the lower side of the ground truth passes through the bottom of the mouth regardless of the face direction, the height of the ground truth is determined. Next, by determining the maximum width, which includes the palpebral fissures, nares, and the mouth, but, as much as possible, does not contain hair, sideburns, and background, the ground truth is created. If the palpebral fissures are occluded by hair, the ground truth is predicted and created. Examples of ground truth are shown in Fig. 3.9.

3.3.3 Evaluation Method

The objective of the proposed method is to localize the multiview face by optimizing parameters for template generation and affine transformation. Also, this research focuses on video processing while localization accuracy must be high in all video frames. In order to evaluate the localization accuracy, the average overlap ratio [58] between the localization result and the ground truth in each target sequence is used. As comparative methods, Orozco's detection method [50] and Saragih's tracking method [59] are applied. Since Orozco's method outputs an upright bounding box, the rotated ground truth cannot be used for the evaluation. Instead, an upright bounding box is created from the ground truth and used because

it almost represents the minimum face region. The Saragih's method also does not output the bounding box so bounding box must be created using landmarks for the evaluation. To create a bounding box similar to the proposed method for fair comparison, just some selected landmarks are used. The proposed method outputs the bounding box, which passes through the top of the palpebral fissures and the bottom of the mouth, and includes all the face parts. Based on this, the landmarks, which indicate the contours of the palpebral fissures and the mouth, and indicate the facial contour included in the range of y coordinate from the top of the palpebral fissures to the bottom of the mouth, are selected. The bounding box is created from these selected landmarks.

3.3.4 Settings

As shown in Fig. 3.2, seven parameters are optimized. Each search space is represented below.

- (a,b,c): ([0,17],[18,35],[-6,5])
- Parallel translation (x,y): ([0,319],[0,239])
- Scale (s): [2.0,5.0]
- Rotation (θ): [-50,50]

As indicated in Fig. 3.3, the a, b, and c are necessary to generate templates from the model. The combined range of a and b is [0,35] to represent from a profile to frontal face. The c represents the appearance change by pitching of the face. The range is [-6,5] and this is the conceivable maximum range in this research. The search ranges of the x and the y are [0,319] and [0,239] because the size of target images is 320 $\times$ 240 pixels. Ranges of scale and rotation are [2.0,5.0] and

[-50,50], which are also conceivable maximum ranges. Chromosome length per one parameter is 8 bits based on our related work [41]. Crossover and mutation probabilities affect convergence speed and maintenance of diversity of population. However, any determination method can be established and the values close to 1.0 and 0.0 are often used [60]. Therefore, 0.95 and 0.05 are set in this research based on the preliminary experiment. There is no established theory to determine population and generation sizes. Generally, population size is smaller than generation size since evolutionary computations acquire an optimum solution by evolving the population. Hence, 100 is set as generation size and five kinds of population sizes (N), 10, 20, 30, 40, and 50, are checked. Twenty random seeds are used and the average is shown as the whole result since GA depends on the random seed.

In the experiment, average processing time per one image is measured. CPU and RAM of the computer are comprised of Intel Core i7-3770S (3.1 GHz) and 16 GB.

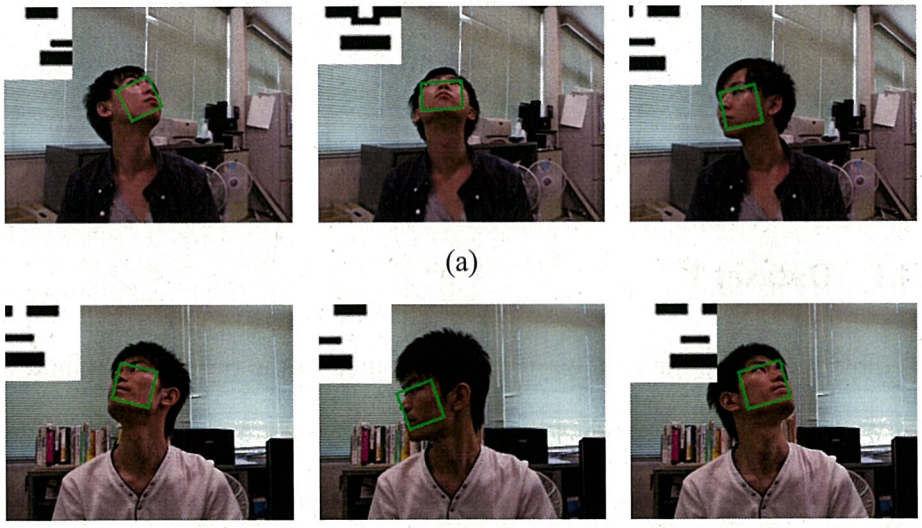
3.4 Result and Discussion

3.4.1 Dataset 1

Table 3.1 shows the results. The performance of the proposed method with all the N and background types is higher. Especially, performances in background 1 and 2 are high. Figure 3.10 shows the localization examples from background 1 and 2. Top left in the image indicates the generated template from the elite individual. A template similar to the face appearance is generated and adapted to face direction. Also, robustness of localization is shown regardless of any large face direction change. Figure 3.11 shows convergence of population in an initial frame of a sequence. The green and red bounding boxes represent the elite and other indi-

Table 3.1: Average overlap ratio in dataset 1.

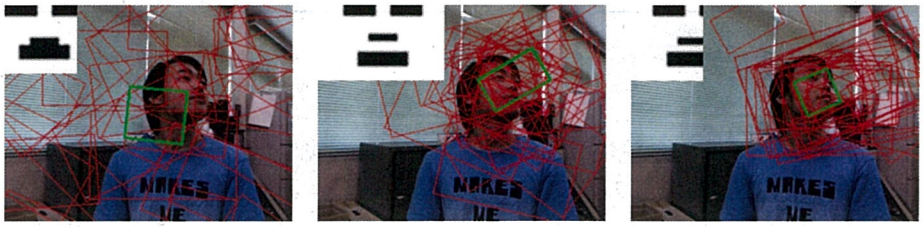
		Background type			
		1	2	3	3'
Our method	$N = 10$	0.61	0.58	0.31	0.45
	$N = 20$	0.63	0.59	0.25	0.45
	$N = 30$	0.63	0.58	0.18	0.45
	$N = 40$	0.64	0.59	0.17	0.45
	$N = 50$	0.64	0.58	0.15	0.44
Saragih et al.		0.54	0.40	0.24	-
Orozco et al.		0.30	0.27	0.25	-



(a)

(b)

Figure 3.10: Examples of the results: (a) from the background 1; (b) from the background 2.



(a) (b) (c)

Figure 3.11: Convergence of the population. Generation is: (a) first; (b) 40th; (c) final.

viduals in the first generation. Top left of image indicates the generated template from the elite. Parameters for affine transformation and template are optimized as generation iterates.

In background 3, the performance of proposed method (when $N = 10$) is higher than the comparative methods, however the difference is lower compared to backgrounds 1 and 2. One reason for this is the existence of a region that is similar to the face. Figure 3.12 shows a typical example and each image from left to right shows the result image with generated template, binary image, and skin region extracted image. The localization fails because the cardboard in the shelf has a similar pattern to that of the template and skin region pixels. This failure occurs in all the sequences of background 3. In order to investigate the failure result in greater depth, we experimented again after the cardboard region ((x,y,$width$,$height$)=(165,61,65,41)) is eliminated as shown in Fig. 3.13(a). The result is represented in Table 3.1 (background 3'). The performance does not improve very much and the average overlap

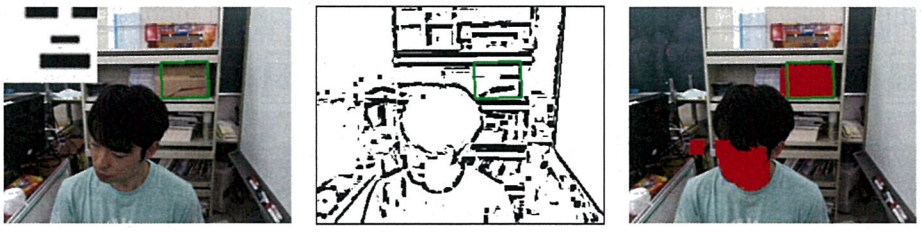
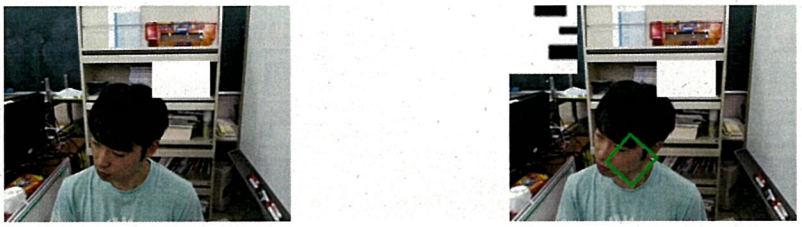


Figure 3.12: Typical failure on a cardboard in the background 3.



(a) (b)

Figure 3.13: Elimination of the cardboard: (a) example of the elimination; (b) detection failure.

Figure 3.14: Examples of tracking failure by Saragih's method.

ratio is about 0.45. The reason for this is that there are many downward faces in background 3 (Fig. 3.13(b)) since the camera position is higher than the faces. The proposed method fails on a downward-looking face because face parts are self-occluded and this is common in all the sequences. Therefore, an improvement such as switching to online tracking, etc., is necessary. We will explore this in our future work.

Performances of comparative methods are lower than proposed method in all sequences. The average overlap ratio by Saragih's method [59] is higher than the Orozco's method [50]. Examples of tracking failure by Saragih's method are shown in Fig. 3.14. This method can track the multiview face with high speed. Nevertheless, fitting error occurs in the middle of tracking since the amount of face direction change is too large for this method. The cumulative error finally causes a localization failure, as shown in Fig. 3.14. Because this failure occurs in consecutive frames, the average overlap ratio decreases. Another reason is face detection fail-

ure. Saragih's method requires face detection to set ROI to fit the model, in order to obtain facial landmarks before tracking. However, there are cases where the multiview face cannot be detected in the initial frame and some consecutive frames due to large face direction. This causes some false negatives and they decrease the average overlap ratio. Thus, the performance of Saragih's method is lower than proposed method. The performance of Orozco's method is also low. This is because the bounding box for detection is too large as shown in Fig. 3.1(a). This method has higher performance on AFLW [55] and FDDB [52] datasets, which are often used as a benchmark. However, if multiview face detection with a minimum bounding box is considered, the performance decreases because the bounding box for detection is too large and it cannot obtain a high overlap ratio. Performance may be improved by simple scaling down of the large bounding box. However, it is not easy to decide the amount of scaling to acquire the minimum bounding box that will fit the face parts. Therefore, simple scaling is not suitable for our purpose. Also, general face detection methods output only upright bounding boxes. It means that the background is included in the bounding boxes when the face has out-of-plane rotation. On the other hand, the bounding box of the proposed method does not include much background since the parameter of in-plane rotation is optimized. Also, correction of in-plane rotation is easy when the optimized parameter is used. This is an important factor for applications, which obtain eyes and mouth information. Therefore, the proposed method is a practical method.

Table 3.2 shows average processing time of each method. The time by the proposed method is shown with each population size (N). Based on increase of N, the time also increases. This processing time is from template generation to generation iteration for the detection. The proposed method is very fast. However, Saragih's method is faster. This is because the amount of appearance change of the

face between the consecutive frames is small, and the local patches of the model can keep fitting and the face can be tracked for a small calculation cost. All the average processing times in Orozco's method are over one second. Since this method uses a deformable part model, some part filters must be applied to an image with different scales. It is necessary to iteratively scan the whole image, so calculation cost is large. Considering average processing time and localization performance, the proposed method is the most exceptional method.

3.4.2 Dataset 2

The average overlap ratio in the dataset 2 is shown in Table 3.3. From the result, performance of proposed method is the highest. Since performance in dataset 1 is also high, the proposed method is the best one. On the other hand, the performances of Sarahi's and the Orozco's methods are low, which is due to the same reason in dataset 1. Figure 3.15 shows result examples in each background of proposed method. Although each subject's face has a variety of locations, scales, and rotations, the proposed method detects all the faces. Similar to the result in the dataset 1, the proposed approach is robust to in dealing with affine transformation and out-of-plane rotation. Therefore, proposed method can be applied to systems, such as feeding devices for orthopedically-impaired people [37] and communica-

Table 3.2: Average processing time in the dataset 1 (ms).

		Background type			
		1	2	3	3'
Our method	$N = 10$	8.6	8.7	9.2	8.9
	$N = 20$	17.3	17.3	18.2	17.9
	$N = 30$	25.7	25.8	27.1	26.7
	$N = 40$	34.3	34.4	35.9	35.5
	$N = 50$	42.9	42.9	44.8	44.3
Saragih et al.		5.6	4.8	5.0	-
Orozco et al.		1233	1246	1241	-

Figure 3.15: Result examples of the proposed method in each background.

tion systems [42].

Nevertheless, proposed method fails in skin region extraction and localization and tracking of a downward-looking face. Figure 3.16 shows a failure example. Because the necessary features are lost by self-occlusion, detection fails. Also, false skin region extraction of cloth is another problem. Therefore, in the figure, it is necessary to propose a new method, which can solve these problems.

Table 3.4 shows the average processing time in dataset 2. This result is similar to dataset 1. The proposed method where $N = 10$ is fast, however Saragih's method is faster. The average processing time by Orozco's method is over one second in all

Table 3.3: Average overlap ratio in dataset 2.

		Background type					
		4	5	6	7	8	9
Our method	$N = 10$	0.48	0.53	0.55	0.62	0.50	0.63
	$N = 20$	0.54	0.53	0.55	0.63	0.57	0.64
	$N = 30$	0.53	0.54	0.55	0.63	0.53	0.64
	$N = 40$	0.53	0.54	0.54	0.63	0.55	0.64
	$N = 50$	0.48	0.53	0.55	0.62	0.50	0.63
Saragih et al.		0.46	0.36	0.51	0.40	0.50	0.47
Orozco et al.		0.31	0.28	0.30	0.30	0.26	0.28

Figure 3.16: Failure example of the proposed method.

Table 3.4: Average processing time in the dataset 2 (ms).

		Background type					
		4	5	6	7	8	9
Our method	$N = 10$	9.6	9.6	9.2	8.8	8.8	8.7
	$N = 20$	18.6	18.0	17.9	17.5	17.6	17.4
	$N = 30$	27.5	26.6	26.5	26.0	26.1	25.8
	$N = 40$	36.5	35.3	35.2	34.6	34.8	34.5
	$N = 50$	45.2	43.8	43.6	42.8	43.2	42.8
Saragih et al.		5.5	5.7	6.0	5.6	5.7	5.6
Orozco et al.		1254	1245	1235	1236	1238	1236

backgrounds. Saragih's method is the fastest, however the localization performance of the proposed method is the highest and it is practical to use.

3.5 Summary

In this research, we propose a novel high-speed multiview face localization and tracking method with a minimum bounding box by applying template matching with GA to video processing. A human head is approximated using a cylinder and from a profile to frontal face is represented using a 2D rectangle model, which is the development of the lateral surface. Then, multiview face localization can be achieved by adaptively generating a template, which corresponds to target face direction, from model. Also, high-speed processing is achieved by simultaneously optimizing parameters for template generation and affine transformation using GA.

For the experiment, different subjects with different backgrounds are recorded in a room using a standard web camera. Performance of proposed method is higher than comparative methods, which are based on prior machine learning. Also, proposed method has the advantage of being able to easily obtain sensing information and correct in-plane rotation because affine parameters can be obtained from an elite individual. However, proposed method often fails when there is a region similar to the face in a background and the face direction is downward. In the future, we will aim for higher performance by considering a new model, features, preprocessing, and method. Due to limitations of space, the estimation of face directions cannot be mentioned. We think the proposed method has substantial potential. The study of this will be our next task.

Chapter 4

Multiview Face Tracking on Privacy Protected Videos

In this chapter, I explain a face tracking method on privacy protected videos. Nowadays, many computer vision techniques are applied to practical applications, such as surveillance and facial recognition systems. Some of such applications focus on information extraction from the human beings. However, people may feel psychological stress about recording their personal information, such as a face, behavior, and cloth. Therefore, privacy protection of the images and videos is necessary. Specifically, detection and tracking methods should be used on the privacy protected images. For this purpose, there are some easy methods, such as blurring and pixelating, and they are often used in news programs, etc. Because such methods just average pixel values, no important feature for the detection and tracking is left. Hence, the preprocessed images are unuseful. In order to solve this problem, we have proposed shuffle filter and a multi-view face tracking method with genetic algorithm (GA). The filter protects the privacy by changing pixel locations, and the color information can be preserved. Since the color information is left, the tracking can be achieved by a basic template matching with histogram. Moreover,

by using GA instead of sliding window when the subject in the image is searched, it can search more efficiently. However, the tracking accuracy is still low and the preprocessing time is large. The proposed multiview face detection and tracking method, which is described in previous chapter, is also unable to achieve high performance. Therefore, a new method is proposed in this research. In the experiment, the improved method is compared with our previous work, CAMSHIFT, an online learning method, and a face detector. The results indicate that the accuracy of the proposed method is higher than the others.

4.1 Introduction

Until now, many cameras are set up everywhere and they are utilized for many purposes through computer vision techniques. One of the major examples is a surveillance system in public and private spaces. Although the surveillance system has been becoming popular, recording in the public and private spaces stresses people out because the recorded videos have much private information, such as a face and behavior. In order to solve this problem, protecting the privacy is necessary. Since a relationship between the privacy protection and the surveillance camera is strong, there are some researches [61, 62]. For example, there is a method that a processing such as background subtraction obscures the personal information. However, this approach is not always appropriate because the original images are used. For example, if an accident detection system in a bathroom is considered, images in which user's nudity appears must be used. Some people may suffer the psychological pressure about using the original color images that everyone can understand who is doing what. From this reason, detection and tracking methods should be used on the privacy protected images.

In order to protect the privacy, blur and pixelation are often used in TV pro-

grams, etc. These methods can protect the privacy by averaging pixel values. Although they are simple and easy to use, the processed images are unuseful for object detection and tracking since no important feature is left. Hence, a new method is necessary. In our previous method [63], we proposed shuffle filter, which replaces a centered pixel with a randomly selected neighborhood pixel. Because it just changes the pixel locations, some local features, such as edge and corner are lost and recognizing what or who is in the preprocessed images becomes difficult. On the other hand, the color information can be preserved. Therefore, the template matching with color histogram was used in our previous work.

Here, the algorithm is briefly explained. The detailed algorithm is introduced later. Before the tracking starts, a frontal face of a user is detected by a face detector using an initial raw image. Then, color histograms are created as the template from the detected region. This template creation step is performed only one time. After that, the tracking starts. The shuffle filter is applied to generate the privacy protected image when a new frame is input. Then, the region, whose histogram corresponds to the template in the privacy protected image, is searched. When the subject is searched, the sliding window is often used in many object detection methods. However, a search window must scan the whole image iteratively with a variety of sizes. Also, if in-plane rotation of the tracking target is considered, some rotated target images must be created and scanned again. This approach is inefficient. In order to solve this problem, we applied a genetic algorithm (GA) [49] instead of the sliding window. By optimizing affine transformation parameters and converging candidate regions, which are generated by individuals, more efficient search was achieved. From the experimental results, the previous method was superior to some related works.

Nevertheless, the preprocessing time of the filter is long because enormous ran-

dom numbers are generated in every input frame, and the tracking accuracy is still low. In order to solve these problems, the preprocessing and the creation of the template methods are improved in this research. In experiments, some related works, which are our previous work [63], CAMSHIFT [64], Viola and Jones face detector [43], and online learning [65], are compared. From the results, the research purpose is achieved.

4.2 Related works

In order to protect privacy, some obscuration methods, such as blurring and pixelating (mosaicing) [66, 67] have been used. These methods are easy to use, and we sometimes watch them in news programs, etc. However, a relationship between a loss of information and strength of the privacy protection is trade-off [67]. Therefore, it is difficult to determine the strength level. From this reason, these methods should not be used. In order to solve this problem, we have proposed shuffle filter that a centered pixel is replaced with a randomly selected neighborhood pixel [63]. Because this method only changes a location of the pixels, the pixel values can be preserved. Hence, using template matching with color histogram is one of valid methods. Also, the privacy can be protected since some local features such as corner and edge are lost when the size of the filter is large. Nevertheless, randomly selecting the neighborhood to replace in every frame is time-consuming since many random numbers must be generated. Therefore, we address this problem in this research.

For the face detection and tracking, many methods have been proposed. Viola and Jones face detector [43] is a famous and a practical method. Based on this method, a multi-view face detection method [68] is also proposed. This approach is applied to many applications since the performance is good. However, collecting

many datasets and checking their quality for the learning are hard work. Also, there is the privacy protection problem about collecting many face datasets. Moreover, it requires many face detectors, which are for each face direction. From these reasons, this method is hard to use.

CAMSHIFT [64] can track an object by manually selecting it before the tracking. The selected region is used to obtain the color histogram. After the tracking starts, a back projected image is created based on the frequency when a new frame is input. In the back projected image, high and low frequencies are represented as intensities from 255 to 0. Then, by searching the high probability region with the mean shift algorithm [69], this method can keep tracking even if the appearance is changed. Generally, if the color of the object drastically changes by occlusion and great appearance change, the tracking is difficult. In order to solve this problem, some improved methods are proposed [70, 71]. However, these methods cannot track a face if the background has the face color. Therefore, they fail to track in such case.

Oikawa et al. [63] propose a template matching using color histograms. Before the tracking starts, a frontal face in an initial raw image is automatically detected using a face detector. Then, only the face region is extracted and converted to YCrCb color space to obtain the histograms of Cr and Cb components. In addition, the histograms of Cr and Cb components are acquired from a frontal region of a head. The reason why four histograms are used is to improve the accuracy. After the tracking starts, only the face region can be tracked by evaluating the template with chi square distance. Although authors show results that the method is superior to CAMSHIFT, the accuracy is low. For a practical use, it is necessary to improve.

Online learning method [65] is a useful method since it can track an object even if the appearance changes and the occlusion occur. Because the method samples

positive and negative data from a target image, they are robust to the appearance changes. Nevertheless, it is necessary to investigate whether the method can keep tracking on a preprocessed image for the privacy protection. Therefore, we examined this.

4.3 Proposed Method

4.3.1 Preprocessing by replacing pixels for privacy protection

As described in Section 4.1 and 4.2, blurring and pixelating with strong level are good for privacy protection, however they cannot be used for any detection and tracking methods since the method just averages the pixel values and no important feature is left. Hence, a new method, which can protect the privacy and preserve the feature, is necessary. Because losing some local features, such as edge and corner is essential for obscuration, one of valid features is color. From this perspective, Oikawa et al. [63] propose shuffle filter that a centered pixel is replaced with a randomly selected neighborhood pixel. The pixel values can be preserved and the color information can be used since this filter just changes the pixel locations. However, this method always generates random numbers to determine the neighborhood pixel to replace in every frame, and it takes a long time. For fast processing, a solution is necessary. One of easy approaches is creating a replacement pattern before the tracking starts and using it when a new frame is input. This means the replacement pattern of every frame is always the same. However, it can protect the privacy since the local features are lost. The detailed algorithm is explained. Firstly, a map, whose size is the same to a target image and it is used to record the replacement pattern, is created. Secondly, $n \times n$ filter is set. Thirdly, a neighborhood pixel is randomly selected. After that, the information, which represents the centered pixel and the se-

(a) (b) (c)

Figure 4.1: Example of replaced pixels with different filter sizes: (a) 3×3; (b) 17×17; (c) 31×31.

lected neighborhood pixel are replaced, is recorded. Then, the filter moves to a next pixel. By iterating these steps, the map can be created. This method can reduce the preprocessing time since preparing this map is required before the tracking starts and generating many random numbers is only one time. Fig. 4.1 shows an example of preprocessed images with different filter sizes.

4.3.2 Acquisition of a color histogram template

Some real problems, such as illumination and appearance changes, must be considered because we consider a practical application. In order to address them, selecting a robust color space for a template is important. In our previous study [63], Cr and Cb components were used since they were the best compared with the others in preliminary experiments. Because the YCrCb color space can extract intensity (Y), this color space is robust to illumination change. Also, the color distributions of the Cr and Cb do not drastically change even if the face appearance is changed. Fig. 4.2 shows the result of bhattacharya distances between the color histograms of a frontal face and the other face directions. The result of the distances is represented at the top left in each image. Except for the downward face, the distances are small. This result means that a multi-view face can be tracked with only one color histogram template, which is created from the frontal face. From this reason, these

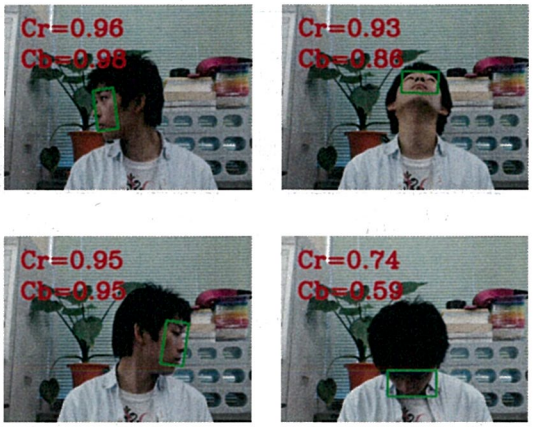


Figure 4.2: Bhattacharya distances between a frontal face and the other face directions.

color components are used in our previous research and this research.

Next, we explain how to acquire the color histograms for the template. In our previous study, they were obtained from the frontal face and the forehead using an initial raw frame before the tracking starts. However, these regions are not good for the robustness. The reason is that when the face pitches, the appearance of the forehead changes and the histogram distances become large. This causes low accuracy. Hence, a new region, which does not drastically change even if the face direction changes, must be used. Therefore, the new acquisition method is proposed in this study. Firstly, a frontal face is detected by [43] (Fig. 4.3(a)), and the detected region is extracted and resized to 1/9 (Fig. 4.3(b)). The reason why the region is resized is to decrease the calculation cost when the face in a target image is searched. The detail is explained in Section 4.3.3. Based on the resized size, a model is created (Fig. 4.3(c)). Each color region in the model represents the locations that the color histograms are obtained. The white color region represents a forehead and sideburns. The centered gray region indicates a frontal face. The bottom light gray region represents a chin. The black region is not used since it is background. Note that the model, which is created for another person, is not exactly the same

(a) (b) (c)

Figure 4.3: Acquisition of a color histogram template based on a model: (a) frontal face detection by [43]; (b) extraction of the detected region and resized result; (c) the model to create the template.

because it is created based on the size of the face detection result. Since Cr and Cb components are used, six histograms are acquired. They are registered as the template after they are normalized.

4.3.3 Template matching with GA

Sliding window is used in most object detection methods when an object in a target image is searched. Here, the procedure of the sliding window is explained. Firstly, a search window, which is an upright rectangle, is set on a target image. The window size is usually determined by a user. If a small object is a detection object, the small size such as 20×20 pixels is set. The initial position is top left of the target image. Secondly, a region in the window is input to a detector and evaluated. The region is displayed as a detection result if the detector outputs the high evaluation value because it means that the region includes the object. Thirdly, the search window moves to right and evaluate the region again. The amount of movement can be determined by the user. The window scans again from left to right when it reaches the right side of the target image. The search starts after the window is set at next row. The number of rows can be also determined by the user. The scan stops when the window reaches at bottom right of the target image. Although this approach is often used, the efficiency of the search is not good. This is because that the window

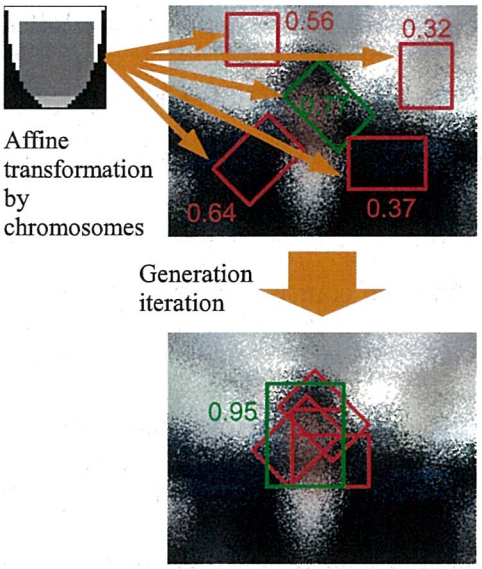


Figure 4.4: Template matching with a GA.

must scan the whole target image and the number of checks of the candidate regions is enormous. The number of checks increases if scaling and in-plane rotation of the object are considered. The reason is that the window with a variety of sizes must scan again on the many rotated target images. Consequently, it takes a long time to process one image. If the search window is flexible about the affine transformation, such as parallel translation, scale factor, and in-plane rotation, the search time can be reduced. In order to achieve this, each parameter must be optimized efficiently. Optimizing them is a combinatorial optimization problem. Hence, a GA is applied in our previous work and in this research. By optimizing the parameters, the object can be searched more efficiently. This is the template matching with the GA. Fig. 4.4 shows the outline of the template matching with the GA. First of all, each individual has the five affine transformation parameters, which are parallel translation of x and y axes, scale factor of x and y axes, and angle of rotation θ. They consist of eight bits. After they are decoded to real numbers, a rectangle, whose size is the same to the model and this is explained in previous section, is transformed and

located in the target image. The located regions are called candidate regions. After that, color histograms are obtained from each candidate region using the model. Note that the number of sampling points is constant even if the candidate region is scaled up and down. It means that no interpolation is used. This is because the interpolation takes a long time. Of course, if the size of the model is large, the sampling time becomes long even if the interpolation is not used because the sampling points increase. In order to avoid this, the detected and extracted face region is resized to 1/9 (Fig. 4.3(b)). This is the main reason the face region is scaled down. After the obtained histograms are normalized, an objective value is calculated by an objective function. It is explained in Section 4.3.4. Next, after a fitness value is calculated, genetic operations, such as selection, crossover, and mutation are performed. An elite individual can be obtained as a detection result by iterating these procedures. The application of GA to template matching has another advantage. By inheriting the population in the final generation to initial generation in the next frame, the initialization is not necessary. Therefore, the processing time can be reduced.

4.3.4 Objective function

The objective function is shown in equation (4.1).

$$\begin{aligned} O &= w_1(D_{H_{cr}} + D_{H_{cb}}) + (D_{F_{cr}} + D_{F_{cb}}) + \\ &\quad w_2(D_{C_{cr}} + D_{C_{cb}}), \qquad (4.1) \\ D &= \sum_{i=0}^{255}\{p(i) - q(i)\}^2. \qquad (4.2) \end{aligned}$$

Let O be an objective value. w is a weight, and D is a histogram distance between the template and an individual. H, F and C are the histograms, which are created from the forehead, face and chin regions (Fig. 4.3(c)). Equation (4.2) represents

chi square distance. The reason why this measurement is used is the performance is better than the other histogram measurements such as bhattacharya distance in preliminary experiments. p and q are the histograms of the template and a candidate region. i is a pixel value. In equation (4.1), the weights are multiplied to the histogram distances of the forehead and chin regions. This is because that these regions include some noise (background) when a facial appearance changes. For instance, when the face direction is upward, the forehead appearance changes, however the chin appearance does not drastically change. When the face direction is downward, this relationship is reversed. Therefore, the histogram distances should be adaptively weighted. Fortunately, since GA can adequately optimize the weights, w_1 and w_2 are optimized in addition to the affine transformation parameters. This idea is only proposed in this research. Each search range is determined empirically and they are [0.5,1.0] and [0.0,0.5]. A fitness value is calculated by a fitness function, which normalizes the objective value to [0.0,1.0].

4.4 Experiment

In order to investigate the effectiveness of the proposed method, we experimented. The test set consists of 15 videos and they were recorded by a web camera and used in our previous work [63]. The image size was 320 $\times$ 240 pixels. The number of appeared persons was five and backgrounds were three. Fig. 4.5 shows the three different backgrounds. Each background type was complex, a little complex and not complex. In the not complex background videos, cardboards appear since their color is similar to human skin. The reason why this condition is set is to investigate whether the used color spaces for the template are appropriate or not. The color spaces are judged as inappropriate if the detection fails on the cardboards. Only the person is in each video and his face drastically changes. Fig. 4.6 shows

(a) (b) (c)

Figure 4.5: Example of test sets: (a) complex background; (b) a little complex; (c) not complex but similar to skin color.

(a) (b) (c)

Figure 4.6: Target images in the test set of Fig. 4.5(b): (a) 54th frame; (b) 87th frame; (c) 140th frame.

the target images in the test set of Fig. 4.5(b). In order to create the template before the tracking starts, the face in all the initial frames are frontal. The number of filter sizes for shuffle was 15 (3×3 to 31×31). The number of individuals and generation iterations was 30 and 100. The probabilities of crossover and mutation were 0.95 and 0.05. Because GA depends on a random seed, we used 10 random seeds and all the results were averaged for evaluation. The coding method of a chromosome was binary, and eight bits were used for each parameter. Hence, the chromosome consists of 56 bits since five parameters for affine transformation and two weights for the objective function are optimized. The search ranges of parallel transformation of x and y axes are [0,320] and [0,240], scale factor of x and y axes are [1.5,3.0], angle of rotation θ is [-15.0,15.0], and weights for histogram distances of a forehead w_1 and a chin w_2 are [0.0,0.5] and [0.5,1.0].

In order to compare with the related works, four methods were selected. They

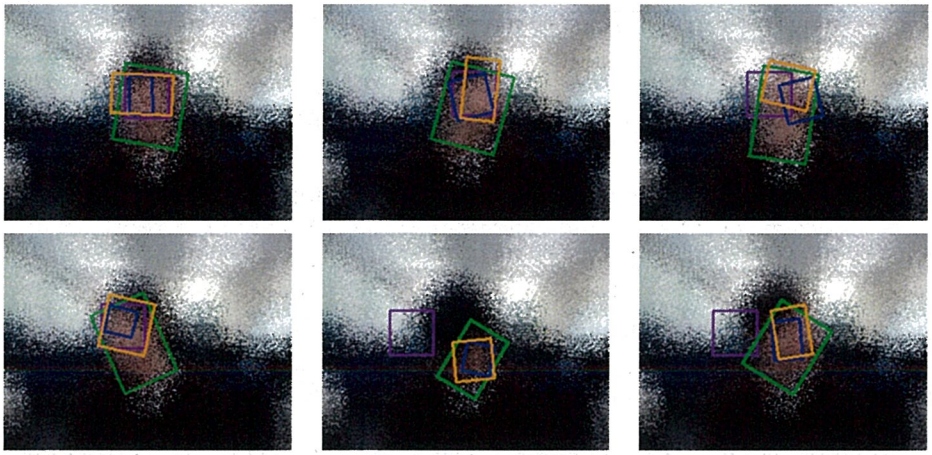


Figure 4.7: Example of detection results for some face directions. Orange is the proposed method, blue is Oikawa et al., green is CAMSHIFT, purple is CRT, and red is Viola and Jones detector, however it cannot detect.

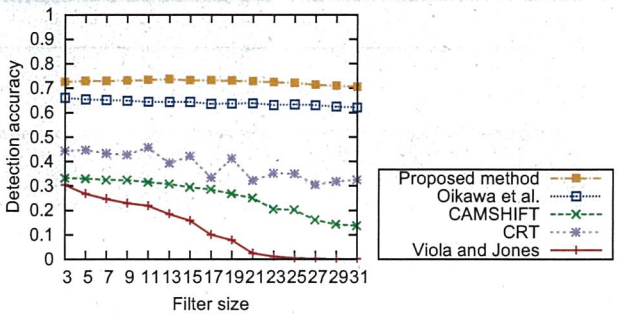
were our previous method, which is color histogram template matching using the color histograms of a frontal face and a forehead [63], Viola and Jones face detector [43], CAMSHIFT [64], and online learning method (CRT) [65]. The proper parameters for each method were set for fair comparison by author. An initial rectangle for the CRT was 70% of the frontal face detection result by [43] since the result includes the background and it causes low accuracy.

The ground truth for the test set was created by hand. It is a rectangle, which has in-plane rotation. The upper side is a top of the eyebrow, the bottom side is the bottom of a jaw, and the right and left side are near a sideburn. The output results were evaluated by bounding box evaluation [72]. In this evaluation, when an overlap ratio exceeds 0.5, the detection result is success. However, this threshold is tight for the experiments since the proposed method does not use the local features. This means that stabilizing the tracking results is difficult and it causes the low accuracy. Therefore, the two thresholds, 40% and 30%, were used. The computer, whose CPU was Intel Core i7-3770S (3.1 GHz) and RAM was 16 GB, was used.

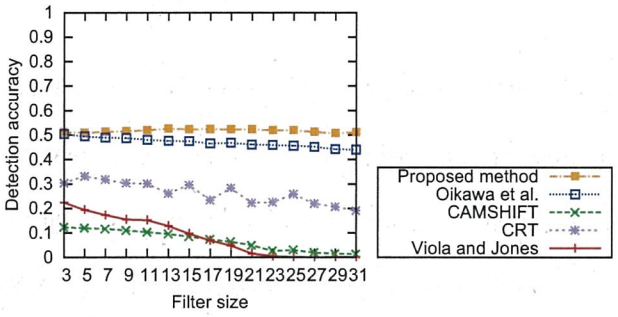
4.5 Results and consideration

Fig. 4.7 shows an example of the detection results by the proposed method and comparative methods in one test set. The face directions in each image are different. The size of the shuffle filter is 31 $\times$ 31 pixels and the background type is complex. The orange rectangle indicates the detection result by the proposed method. The blue, green, purple, and red rectangles represent the results by the Oikawa et al., CAMSHIFT, CRT, and Viola and Jones face detector. There is no detection result by the Viola and Jones face detector since it can detect nothing. The proposed method and CAMSHIFT can track the face. This is because that they use the color information as a feature to track. Therefore, they can track the face even if the large filter size is used. The middle and right images in the second row show the tracking failure of the CRT. The reason why the tracking fails is that the errors by the filter accumulate. This method uses a local feature to learn in every frame. However, the feature cannot be obtained since the shuffle filter loses it. This causes the tracking failure.

Fig. 4.8 shows the detection accuracy by each method. In all results, the detection accuracy of the proposed method is the highest. It indicates that the proposed method improves our previous method by using a model to create the color histograms and adaptive weights for the histogram distances. The accuracy of our previous research, which obtains the color histograms from a frontal face and a forehead region, is the second. The reason why this previous method is inferior to the proposed method is that additional information of the color histograms is insufficient. In the both of the previous and proposed methods, the histograms of the frontal face and the forehead region are used. However, the previous method does not use the histograms from the chin region. This causes a tracking failure when the face direction is upward. Also, the two weights for the histograms from the fore-



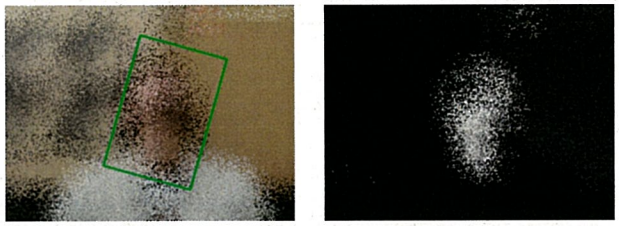
(a)



(b)

Figure 4.8: Detection result: (a) PASCAL=0.3; (b)PASCAL=0.4.

head and the chin regions are one of the causes. In the proposed method, they are optimized by the GA. Hence, it can tune them when the face direction changes. This improves the accuracy. There is one case that the proposed method fails to track. It is the downward face. This case is difficult because the facial parts are occluded and the illumination on the face becomes dark as shown in Fig. 4.2. Hence, this problem must be solved in the future. The detection accuracy of the online learning is low. The reason is that local features are lost by the filter for the protection privacy. Another reason is the parameter setting for decision trees. Generally, the deep and many trees increase the detection accuracy. However, the calculation cost becomes high. Because this is trade-off, setting the parameter is difficult. In this experiment, the parameters were set based on they can process in 100 ms per one frame. Hence, the accuracy is low. Viola and Jones face detector can detect nothing. The reason is that the local feature is lost by the filter. From this result, it indicates that



(a) (b)

Figure 4.9: Tracking results by CAMSHIFT: (a) detection results; (b) probability distribution image.

the shuffle filter can protect the privacy against the face detector, which is created with many clear images. Nevertheless, additional investigation is necessary since there is a possibility that the detector, which is created from the privacy protected images, can detect the face on the preprocessed images. CAMSHIFT can track the face accurately with high speed. However, the detection accuracy is low. Fig. 4.9 shows an example of the failure when the face direction is upward. Fig. 4.9(a) is the detection result and (b) is the probability distribution image. Because the face and neck regions have much skin color, the detection rectangle is too large. This method cannot track only the face when the background is similar to the skin color. Therefore, this method has the limitation.

In order to reduce the preprocessing time of the shuffle filter, a new method is proposed in this research. In the experiments, the time was measured and compared. When the filter size was 3×3 and 31×31, the proposed method takes 1.2 and 1.0 ms and the previous method takes 4.0 and 90.6 ms. Obviously, the proposed method improves the preprocessing time. From the whole experimental results, the proposed method achieves the purposes in this research.

4.6 Summary

In this chapter, we propose an improvement method of our previous study, which can protect the privacy of images with fast speed and track a multi-view face on the preprocessed images. Because some local features, such as edge and corner are lost by the obscuration filter, the color feature is focused on. When a template is created, Cr and Cb color histograms from a frontal face, forehead, and chin regions are acquired by using a model. By using GA instead of sliding window when a face in the image is searched, the face can be detected more efficiently. Moreover, introducing the two weights for objective function and optimizing them, the tracking performance of the proposed method is improved. From the experiments, the improvement of the preprocessing time and the detection accuracy are confirmed.

Next task is increasing the test sets and improving the accuracy. Specifically, the tracking failure of a downward face must be solved. Also, proposing a new template creation method is necessary since the template is created from an initial raw image and the privacy is not protected in this step.

Chapter 5

Conclusion

In this thesis, I describe photographic environment independent multiview face detection and tracking using template generation by genetic algorithm. The photographic environment means an illumination condition, background, appearance of an object, and image quality in this thesis. In order to detect and track a multiview face in such photographic environments, two research tasks are mainly addressed.

In the first task, template matching with GA is focused and improved to develop a basic technique for the multiview face detection and tracking. By introducing a concept of automatic template generation to the basic technique, effective and fast template generation is achieved. Also, by optimizing geometric transformation parameters using GA for the effective target object search in a target image, high-speed multiview face detection and tracking are achieved. For experiments, a challenging video dataset was created, and recent machine learning-based method and tracking method are compared to the proposed method. As a result, the effectiveness of the proposed method is confirmed.

Next, multiview face tracking on privacy protected videos are addressed as an application research. In order to generate privacy protected images, a new method, which is different from conventional privacy protection filters, is proposed. This

method generates the privacy protected images by randomly replacing pixels in the filter. Since only the pixel positions are changed, original pixel values are preserved. In other word, color information can be used as a feature to track. Therefore, color histograms are used as templates. The templates are acquired from a frontal target face, and multiview face tracking is achieved by optimizing geometric transformation parameters. In experiments, a video dataset was created and the proposed method is compared to related works. The results show the effectiveness of the proposed method.

As describe, by applying a GA-based technique, photographic environment independent multiview face detection and tracking is achieved in this research. The developed algorithms can contribute to a variety of systems using face detection and tracking. In the future, the developed methods will be extended to detect multiple objects for practical use.

Bibliography

[1] M.-H. Yang, D. J. Kriegman, and N. Ahuja, "Detecting faces in images: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, pp. 34–58, 2002.

[2] G. Yang and T. S. Huang, "Human face detection in complex background," *Pattern Recognition*, vol. 27, no. 1, pp. 53–63, 1994.

[3] S. A. Sirohey, "Human face segmentation and identification," University of Maryland, Tech. Rep. CS-TR-3176, 1993.

[4] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, 1986.

[5] H. P. Graf, T. Chen, E. Petajan, and E. Cosatto, "Locating faces and facial parts," in *Proc. International Workshop on Automatic Face and Gesture Recognition*, 1995, pp. 41–46.

[6] T. K. Leung, M. C. Burl, and P. Perona, "Finding faces in cluttered scenes using random labeled graph matching," in *Proc. IEEE International Conference on Computer Vision*, 1995, pp. 637–644.

[7] M. F. Augusteijn and T. L. Skujca, "Identification of human faces through texture-based feature recognition and neural network," in *Proc. IEEE Conference on Neural Networks*, 1993, pp. 392–398.

[8] D. Chai and K. Ngan, "Face segmentation using skin-color map in videophone applications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 4, pp. 551–564, 1999.

[9] Q. Chen, H. Wu, and M. Yachida, "Face detection by fuzzy matching," in *Proc. IEEE International Conference on Computer Vision*, 1995, pp. 591–596.

[10] P. Sudowe and B. Leibe, "Efficient use of geometric constraints for sliding-window object detection in video," in *Proc. International Conference on Computer Vision Systems*, 2011, pp. 11–20.

[11] T. Sakai, M. Nagao, and S. Fujibayashi, "Line extraction and pattern detection in a photograph," *Pattern Recognition*, vol. 1, no. 3, pp. 233–248, 1969.

[12] I. .Craw, H. Ellis, and J. Lishman, "Automatic extraction of face features," *Pattern Recognition Letters*, vol. 5, pp. 183–187, 1987.

[13] V. Govindaraju, "Locating human faces in photographs," *International Journal of Computer Vision*, vol. 19, no. 2, pp. 129–146, 1996.

[14] A. Samal and P. A. Iyengar, "Human face detection using silhouettes," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 9, no. 6, pp. 845–867, 1995.

[15] I. T. Jolliffe, Ed., *Principal Component Analysis*. Springer-Verlag New York, 2002.

[16] R. O. Duda and P. E. Hart, "Use of the hough transformation to detect lines and curves in pictures," *Communications of the ACM*, vol. 15, no. 1, pp. 11–15, 1972.

[17] P. Sinha, "Object recognition via image invariants: A case study," *Investigative Ophthalmology and Visual Science*, vol. 35, no. 4, pp. 1735–1740, 1994.

[18] B. Scassellati, "Eye finding via face detection for a foevated, active vision system," in *Proc. National Conference on Artificial Intelligence*, 1998, pp. 1146–1151.

[19] Y. H. Kwon and N. da Vitoria Lobo, "Face detection using templates," in *Proc. International Conference on Pattern Recognition*, 1994, pp. 764–767.

[20] A. Lanitis, C. J. Taylor, and T. Cootes, "An automatic face identification system using flexible appearance models," *Image and Vision Computing*, vol. 13, no. 5, pp. 393–401, 1995.

[21] T. F. Cootes and C. J. Taylor, "Locating faces using statistical feature detectors," in *Proc. International Conference on Automatic Face and Gesture Recognition*, 1996, pp. 204–209.

[22] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1994.

[23] K. K. Sung and T. Poggio, "Example-based learning for view-based human face detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 39–51, 1998.

[24] B. Moghaddam and A. Pentland, "Probabilistic visual learning for object recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 696–710, 1997.

[25] P. Belhumeur, J. Hespanha, and D. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, 1997.

[26] D. I. Barnea and H. F. Silverman, "A class of algorithms for fast digital image registration," *IEEE Transactions on Computers*, vol. c-21, no. 2, pp. 179–186, 1972.

[27] S. L. Tanimoto, "Template matching in pyramids," *Computer Graphics and Image Processing*, vol. 16, no. 4, pp. 356–369, 1981.

[28] H. Murase and V. V. Vinod, "Fast visual search using focused color matching-active search," *Systems and Computers in Japan*, vol. 31, no. 9, pp. 81–88, 2000.

[29] C. H. Lampert, M. B. Blaschko, and T. Hofmann, "Beyond sliding windows: Object localization by efficient subwindow search," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[30] I. Boussaïd, J. Lepagnot, and P. Siarry, "A survey on optimization metaheuristics," *Information Sciences*, vol. 237, no. 10, pp. 82–117, 2013.

[31] J. H. Holland, Ed., *Adaptation in Natural and Artificial Systems*. MIT Press, 1975.

[32] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE International Conference on Neural Networks*, 1995, pp. 1942–1948.

[33] R. Storn and K. Price, "Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[34] T. Blickle and L. Thiele, "A comparison of selection schemes used in genetic algorithms," *Evolutionary Computation*, vol. 4, no. 4, pp. 361–394, 1996.

[35] D. E. Goldberg and K. Deb, "A comparative analysis of selection schemes used in genetic algorithms," in *Proc. Foundations of Genetic Algorithms*, 1991, pp. 69–93.

[36] J. Sato and T. Akashi, "High-speed multiview face localization and tracking with a minimum bounding box using genetic algorithm," *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 12, no. 5, 2017, (in press).

[37] T. Akashi, Y. Wakasa, K. Tanaka, S. Karungaru, and M. Fukumi, "Interactive interface with evolutionary eye sensing and physiological knowledge," *IEEJ Transactions on Electronics, Information and Systems*, vol. 129, no. 7, pp. 1288–1295, 2009.

[38] R. Xu, D. C. W. II, and R. L. Frank, "Inference of genetic regulatory networks with recurrent neural network models using particle swarm optimization," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 4, no. 4, pp. 681–692, 2007.

[39] J. Sato and T. Akashi, "Performance comparison between ga and pso in video processing," in *Proc. 11th International Conference on Quality Control by Artificial Vision*, 2013, pp. 156–161.

[40] T. Akashi, Y. Wakasa, K. Tanaka, S. Karungaru, and M. Fukumi, "High speed genetic lips detection by dynamic search domain control," *IEEJ Transactions on Electronics, Information and Systems*, vol. 127, no. 6, pp. 854–866, 2007.

[41] T. Akashi, H. Kubota, H. Tomita, and H. Konno, "Evolutionary driver's eye sensing method," in *Proc. International Symposium on Future Active Safety Technology toward zero-traffic-accident*, 2011, pp. CD–ROM.

[42] T. Saitoh, "Development of communication support system using lip reading," *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 8, no. 6, pp. 574–579, 2013.

[43] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, 2001, pp. 511–518.

[44] S. S. Farfade, M. Saberian, and F. Li, "Multi-view face detection using deep convolutional neural networks," in *Proc. International Conference on Multimedia Retrieval*, 2015, pp. 643–650.

[45] S. Yang, P. Luo, C. Loy, and X. Tang, "From facial parts responses to face detection: A deep learning approach," in *Proc. IEEE International Conference on Computer Vision*, 2015, pp. 3676–3684.

[46] M. George, N. Ghanem, and M. Ismail, "Learning-based incremental creation of web image databases," in *Proc. International Conference on Machine Learning and Applications*, 2013, pp. 424–429.

[47] M. Wiering, "Convergence and divergence in standard and averaging reinforcement learning," in *Proc. European Conference on Machine Learning*, 2004, pp. 477–488.

[48] R. Girshick, "Fast r-cnn," in *Proc. IEEE International Conference on Computer Vision*, 2015, pp. 1440–1448.

[49] D. E. Goldberg, Ed., *Genetic Algorithm in Search, Optimization and Machine Learning*. Addison Wesley Longman, 1989.

[50] J. Orozco, B. Martinez, and M. Pantic, "Empirical analysis of cascade deformable models for multi-view face detection," *Image and Vision Computing*, vol. 42, pp. 47–61, 2015.

[51] M. Isard and A. Blake, "Contour tracking by stochastic propagation of conditional density," in *Proc. European Conference on Computer Vision*, 1996, pp. 343–356.

[52] V. Jain and E. Learned-Miller, "Fddb: A benchmark for face detection in unconstrained settings," University of Massachusetts, Tech. Rep., 2010.

[53] M. Cascia, S. Sclaroff, and V. Athitsos, "Fast, reliable head tracking under varying illumination: An approach based on registration of texture-mapped 3d models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 4, pp. 322–336, 2000.

[54] X. Hu and E. Paolo, "An efficient genetic algorithm with uniform crossover for air traffic control," *Computers & Operations Research*, vol. 36, no. 1, pp. 245–259, 2009.

[55] M. Koestinger, P. Wohlhart, P. Roth, and H. Bischof, "Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization," in *Proc. IEEE International Workshop on Benchmarking Facial Image Analysis Technologies*, 2011, pp. 2144–2151.

[56] G. Fanelli, M. Dantone, J. Gall, A. Fossati, and L. Gool, "Random forests for real time 3d face analysis," *International Journal of Computer Vision*, vol. 101, no. 3, pp. 437–458, 2013.

[57] J. Sato and T. Akashi. A video dataset for multiview face localization and tracking. [Online]. Available: http://cvhost.scv.cis.iwate-u.ac.jp/research/projects/mvfd.html

[58] M. Everingham, S. M. A. Eslami, and L. V. Gool, "The pascal visual object classes challenge: A retrospective," *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, 2015.

[59] J. Saragih, S. Lucey, and J. Cohn, "Face alignment through subspace constrained mean-shifts," in *Proc. IEEE International Conference on Computer Vision*, 2009, pp. 1034–1041.

[60] A. Eiben, R. Hinterding, and Z. Michalewicz, "Parameter control in evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 124–141, 1999.

[61] F. Z. Qureshi, "Object-video streams for preserving privacy in video surveillance," in *Proc. IEEE International Conference on Advanced Video Signal Based Surveillance*, 2009, pp. 442–447.

[62] J. Wickramasuriya, M. Datt, S. Mehrotra, and N. Venkatasubramanian, "Privacy protecting data collection in media space," in *Proc. Annual ACM International Conference on Multimedia*, 2004, pp. 48–55.

[63] D. Oikawa, J. Sato, and T. Akashi, "Improved face tracking with privacy protection using half ellipse and additional region matching," in *Proc. Joint Conference of IWAIT and IFMIA*, 2015, pp. CD–ROM.

[64] G. R. Bradski and S. Clara, "Computer vision face tracking for use in a perceptual user interface," Tech. Rep. Q2 '98, 1998.

[65] C. Zhang, Y. Yamagata, and T. Akashi, "Robust visual tracking via coupled randomness," *IEICE Transactions on Information and Systems*, vol. E98.D, pp. 1080–1088, 2015.

[66] M. Boyle, C. Edwards, and S. Greenberg, "The effects of filtered video on awareness and privacy," in *Proc. ACM Conference on Computer Supported Cooperative Work*, 2000, pp. 1–10.

[67] R. Gross, E. Airoldi, B. Malin, and L. Sweeney, "Integrating utility into face de-identification," in *Proc. International Workshop on Privacy Enhancing Technologies*, 2006, pp. 227–242.

[68] C. Huang, H. Ai, Y. Li, and S. Lao, "High-perfomance rotation invariant multiview face detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, pp. 671–686, 2007.

[69] K. Fukunaga, *Introduction to Statistical Pattern Recognition, Second Edition*. Academic Press, 1990.

[70] D. Exner, E. Bruns, D. Kurz, and A. Grundhofer, "Fast and robust camshift tracking," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2010, pp. 9–16.

[71] J. G. Allen, R. Y. D. Xu, and J. S. Jin, "Object tracking using camshift algorithm and multiple quantized feature spaces," in *Proc. Pan-Sydney Area Workshop on Visual Information Processing*, 2005, pp. 3–7.

[72] M. Everingham, L. V. Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, pp. 303–338, 2010.