Evolutionary Object Detection and Information Extraction for Image Understanding

March 2006

Takuya Akashi

Evolutionary Object Detection and Information Extraction for Image Understanding

A DISSERTATION
submitted in partial fulfillment
of the requirements for the degree

Doctor of Engineering

in Information Science and Systems Engineering

by

Takuya Akashi

2006

The dissertation of Takuya Akashi is approved.

Professor Shunichiro Oe

Professor Minoru Fukumi

Professor Norio Akamatsu, Committee Chair

The University of Tokushima, Japan

2006

To my parents . . .
who—among so many other things—
saw to it that I learned to play the piano
while I was still in elementary school

Acknowledgements

Before presenting this thesis, I would like to express my special thanks and appreciation for the help, advice and blessings that I received throughout the time I worked on this project.

I wish to sincerely thank my supervisors and advisers Professor Norio Akamatsu and Professor Minoru Fukumi for their kind support, invaluable advice, comments and guidance throughout my stay in the A3 laboratory. I also thank Professor Shunichiro Oe and the professors of the department of Information Science and Intelligent Systems, the University of Tokushima, for their advice and provision of some of the reference material used in this work. Moreover, I appreciate the advise which I received from: Professor Yoshisuke Kurozumi, Professor Tsutomu Takeuchi, and the professors of the department of Information Communication Engineering, Kyoto Sangyo University. I thank Professor Kanya Tanaka and Associate Professor Yuji Wakasa, who gave thoughtful consideration to my study. I cannot forget all students and staff of the A3 research group for all kinds of help and assistance extended to me even sometimes without my knowledge.

I am also very greatful to Dr. Stephen Karungaru for his help in my studies adn social life.

God bless all of you.

Special thanks goes to my wife Kazue for sacrificing herself to be with me throughout this work. I cannot forget to thank my parents and brother for always being there to cheer me up. Thank you.

May the almighty God bless you all for your kindness. Thank you very much.

Takuya Akashi.

March 2006.

Vita

1978 Born, in Tokyo, Japan

2001 B.S. (Engineering), Kyoto Sangyo University.

2003 M.S. (Engineering), The University of Tokushima.

2004-2005 Research Assistant, Information Science and Intelligent Systems, Faculty and School of Engineering, The University of Tokushima.

2005-present Assistant Professor Associate, Department of Electrical and Electronic Engineering, Faculty of Engineering, Yamaguchi University.

Publications

Main Papers (Journal papers)

1. Genetic Lips Extraction Method for Varying Shape, Takuya Akashi, Yasue Mitsukura, Minoru Fukumi, and Norio Akamatsu, IEEJ Transaction on Electronics, Information and Systems, Vol. 124, No.1, pp.128-137, January 2004, in Japanese.

2. High Speed Genetic Lips Detection by Dynamic Search Domain Control, Takuya Akashi, Stephen Karungaru, Minoru Fukumi, Norio Akamatsu, Yuji Wakasa, and Kanya Tanaka, IEICE Transactions on Information and Systems, (Submitted).

Secondary (International Conferences)

1. Lips Extraction with Template Matching by Genetic Algorithm, Takuya Akashi, Minoru Fukumi, and Norio Akamatsu, The Sixth International Conference on Knowledge-Based Intelligent Information & Engineering Systems, pp.343-347, Crema, Italy, 2002.

2. Genetic Lips Extraction Method for Varying Shape, Takuya Akashi, Yasue Mitsukura, Minoru Fukumi, and Norio Akamatsu, IEEE International Symposium on Computational Intelligence in Robotics and Automation, pp.982-987, Kobe, Japan, July 2003.

3. Improvement of Lips Region Extraction Method for Lipreading, Takuya Akashi,

Minoru Fukumi, and Norio Akamatsu, The Fifth IASTED International Conference on Signal and Image Processing, pp.127-132, Hawaii, USA, August 2003.

4. Tracking of Moving Object Using Deformable Template, Takuya Akashi, Minoru Fukumi, and Norio Akamatsu, The Seventh International Conference on Knowledge-Based Intelligent Information & Engineering Systems, pp.1162-1168, Oxford, United Kingdome (2003).

5. Accuracy and Speed Improvement in Lips Region Extraction, Takuya Akashi, Minoru Fukumi, and Norio Akamatsu, The Eighth Australian and New Zealand Conference on Intelligent Information Systems, pp.495-500, Sydney, Australia, December 2003.

6. Invariant Lips Extraction for Variation of Horizontal Direction, Takuya Akashi, Minoru Fukumi, and Norio Akamatsu, The Sixth IASTED International Conference on Signal and Image Processing, pp.58-63, Hawaii, USA, August 2004.

7. Genetic Lips Extraction Method with Flexible Search Domain Control, Takuya Akashi, Minoru Fukumi, and Norio Akamatsu, The Eighth Knowledge-Based Intelligent Information & Engineering Systems, pp.779-806, Wellington, New Zealand, September 2004.

8. Real-Time Genetic Lips Region Detection and Tracking in Natural Video Scenes, Takuya Akashi, Minoru Fukumi, and Norio Akamatsu, 2004 IEEE Conference on Cybernetics and Intelligent Systems, pp.682-687, Singapore, December 2004.

9. Three-dimensional Geometric Information Acquisition of Lips for Natural Scene, Takuya Akashi, Minoru Fukumi, and Norio Akamatsu, 2005 RISP International workshop on Nonlinear Circuits and Signal Processing, pp.123-126,

Hawaii, USA, March 2005.

10. Estimation of Face Direction Using Near-Infrared Camera, Takuya Akashi, Hironori Nagayama, Minoru Fukumi, and Norio Akamatsu, 2005 RISP International Workshop on Nonlinear Circuits and Signal Processing, pp.231-234, Hawaii, USA, March 2005.

Preface

Image Understanding is the process to understand the content of images in order to automate visual tasks by computers. A visual task is some activity which relies on vision. Usually the "input" to this activity is a single image (picture) or video sequence, and usually the "output" is some decision, description, action, or report.

Why do these tasks need to be automated?

Because there is something to be gained from having a computer do the tasks rather than a human. There are several reasons that computers are more suitable than humans for visual tasks. The reasons cited below cover many applications, perhaps not all, and often there is more than one reason in an application.

Dangerous situation Tasks which is too dangerous for human.

Examples are: robots in nuclear power stations, robotic planetary exploration.

Sensitive situation Tasks which suffer if the human fatigues, and which are prone to this problem.

Examples are: industrial inspection, video-based security systems.

Economical situation Tasks which require specialized training, resulting in human resources that are rare and costly.

Example are: Medical screening for tumors, intelligence gathering from satellite imagery.

Strict situation Tasks which humans do poorly because visual items need to be measured accurately.

Example are: progress of disease, efficacy of medication, growth of cracks in weldments, number of specific cells in a microscope slide.

Humanly impossible situation Tasks which have too much data for effective application of humans.

Examples are: counting the potholes in highways, inspection of every bottle in a bottling plant, keeping up with intelligence data during wartime.

Image Understanding systems start by processing images to remove noise and irrelevant information and to enhance the relevant information, then they analyze the image with feature extraction techniques. The technical challenge is to make the computer understand the contents of the images. In other words, the most difficult problem is to automatically produce a reasonable description from an image. It is clear that the nature of images and descriptions have a big distance. In the fields of Artificial Intelligence, Scene Analysis, Image Analysis, Image Processing, and Computer Vision, the many researchers work on reducing this distance in the last twenty years.

However, there are few Image Understanding systems which are suitable for practical use. The reason is that it is difficult to extract the relevant information to represent the object stability, supporting real-world. The object has complex changes by various causes. New object detection and information extraction approach, which can be applied to these various changes, is necessary for Image Understanding.

This dissertation is structured into five parts. The first, Chapters 1 and 2, deal with the introduction of this dissertation and theoretical background. Chapter 1 states the problems about object detection and extraction of information to represent the object. Chapter 2 shows important theoretical background for this study; color space, geometric transformation group and Genetic Algorithm. The second part (Chapter 3) is very important, and describes a basis technique for the following chapters. These techniques are a proposed genetic object detection and the information extraction system. The third part (Chapters 4 and 5) deals with three-dimensional object detection and an information extraction. The fourth part (Chapters 6 and 7) is the main part of this work. This part overcomes a trade-off between the speed and the accuracy. This trade-off is the problem in the previous chapters. Chapter 6 introduces the downsized GA to overcome the trade-off for a single image processing. In Chapter 7, the evolutionary video processing for Image Understanding is proposed. The last part is Chapters 8 and 9. Chapter 8 outlines future work and research projects. These works are working in real time, three-dimensional image understanding, and robotics. A final summary of this work is in Chapter 9.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Purpose

The purpose of this work is object detection in an active scene for Image Understanding. As part of the objectives, we also try to acquire numerical information to represent the object. In this dissertation, the object to be treated is lips region of a talking person. The camera is free to move and independent with the human. There are two reasons of this active situation. The first, the lips have various deformations by speech, and an appearance of lips changes by free camera motion. The second, the lips are very useful for many important applications, as described hereinbelow.

Speech recognition is one of the most useful interfaces that uses little space and without physical contact between the device and the human. However, for the mobile devices and the ubiquitous computing interface, it has some problems: performance limitation and background noise by various situations, such as public spaces, stores, offices, and home. In order to overcome these problems, the lips information should be effectively used. Because, in human speech perception, audio-visual integration is very useful [1, 2, 3]. Therefore, using the lips image is very important for many applications, such as interface in mobile devices, and many studies of audio-visual speech recognition by lips image have been reported [4, 5, 6, 7, 8, 9].

On the other hand, an image-based surveillance system with mobile robots, such as wheeled robot [10] and micro aerial robot [11, 12], has been proposed. Moreover, speaker identification by using lips information is performed in [13, 14]. If these two technologies are collaborated, a human surveillance system with an mobile robot

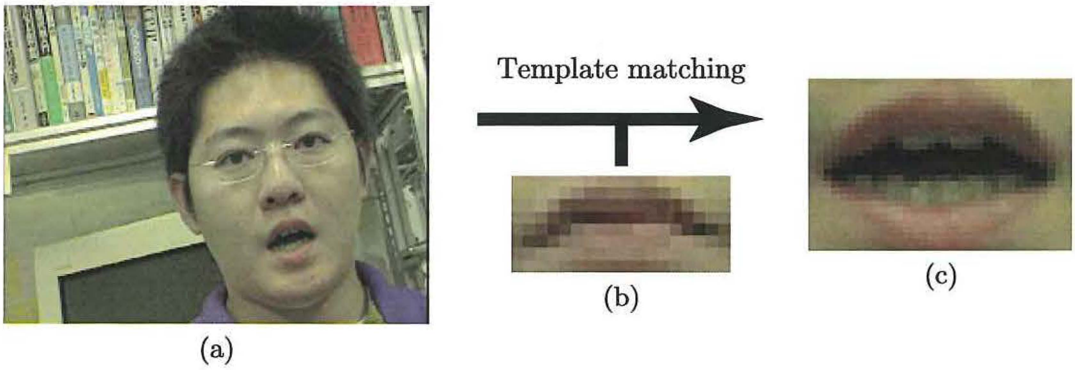


Figure 1.1 Examples of image: a) target ; b) template ; c) detected lips as steady-state.

will be achieved, as mentioned in [15]. Robots such as a pet and a humanoid robot are expected to be peers of human. For a nonverbal communication between these robots and human, a function of tracking speakers is important because the robots must locate people, activate for their voice and look at them to identify visually, and associate voice and visual images. Therefore the lips is very important for interfaces of the personal mobile device.

Moreover, the lips redness is common to the entire human race. The reason is that the redness is called the "vermillion border" and composed of nonkeratinized squamous epithelium that covers numerous capillaries, which give the lips characteristic color [16, 17, 18, 19]. For these reasons, we focus on the lips redness as main feature during detection.

In this dissertation, we describe the lips detection and lips information acquisition as the image-based front-end of audio-visual speech recognition and speaker identification with a personal mobile device, as exemplified above. In order to make these speech recognition and speaker identification process simple and easy, it is preferred that lips used in the process is steady-state (see Figure 1.1). Considering the mobile devices, a camera and human move independently. Therefore, geometric change, such as parallel translation, scaling, and rotation, must be corrected. For a real-time application, the system should process both detection of lips and acquisition of lips information at the same time.

In our previous report [20], we used a single template matching with a GA(Genetic

Algorithm) [21] for one frame obtained from a video sequence. GA's chromosome in our system specified geometric information of lips. Therefore it can detect lips region even in case that lips region has some significant geometric changes by free camera motion.

The Genetic and Evolutionary Computation (GEC) is the generic name for GA, Genetic programming (GP) that is the extension of GA, Evolutionary Strategy (ES), and Evolutionary Programming (EP) [22]. Recently, GECs have gained a growing popularity and a fairly great number of attempts to use GECs to solve complex problems in various application fields [22]. Therefore, GECs can be applied to complex problem as mentioned above, which are object detection into an active scene for Image Understanding.

However, there is a trade-off between accuracy and a processing time in the previous system, which cannot be come closer to the real-time processing. Consequently, we describe a new method, search domain control (SD-Control) and evolutionary video processing, which can make the system apply to a real-time processing. This is the main subject in this paper.

1.2 Statement of the Problem

In this section, in order to clarify the problem which is treated in this study, some related works and images, which is input in the proposed system, are described.

The purpose of our study is detection of a lips region, extraction of lips information, as an interface and front-end of the audio-visual speech recognition on personal mobile devices. Hence a camera and human move independently, and the lips region can have some significant geometric changes, such as parallel translation, scaling, and rotation. Moreover, the lips shape changes by speech. In this paper, we address three issues for lips detection as follows.

1. Active scene by free camera motion
2. High accuracy in detection of lips region and extraction of lips geometric information

3. High speed processing

1.2.1 Related Work

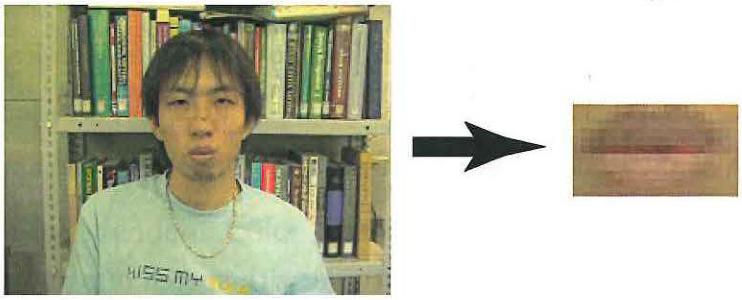
Many studies of audio-visual speech recognition by lips image have been reported [4, 5, 6, 7, 8, 9]. As far as we know, the most of these studies set a precondition for lips image. For example, in [4] the subject wears a helmet with camera, and in [5, 6, 7], extracted lips images in a database are used from the start. Moreover, in [8, 9], the lips information is acquired through the lips detection after detection for static face. For the real-time process, lips detection and lips information acquisition must be performed in only one phase.

Major methods of image processing for face images are divided into three, an image-based approach, a model-based approach, a both image- and model-based approach. In the image-based approach, eigenlips methods [7, 9] have been proposed. In these methods, a set of training lips images is generated by the principal component analysis. The training data must be chosen carefully to include all possible lips configurations. Other proposed image-based approaches are rule-based approaches by features of a face [23], pixel-based approaches by red exclusion [5], optical-flow approaches to measure lips movement [24], etc. These approaches cannot adapt to considerable geometric changes in every frame of lips. In the model-based approach, Active Shape Model [6], and Genetic Snakes [8, 25] which is an improved version of Snakes [26] have been proposed. The optical flow is used to track facial expression as motion of the craniofacial muscle [27, 28], however, these approaches cannot be applicable for the free camera motion. These approaches have some constraints, such that a target image is only a face region, human pose is fixed, and a subject wears a camera to obtain a mouth image, and so on. Because, the initial setting of problems is needed and the number of nodes and parameters should be skillfully determined. Therefore, these approaches are difficult to be applied to our purpose. As for both image- and model-based approaches, a high speed face tracking method [29] was proposed. In this method, many facial feature patch templates must be prepared as a training set. These templates are regions surrounding the feature, such as eye and mouth, then the information of object cannot be acquired at the detection. Therefore, after the detection, the information is acquired by another step. From this reason, these methods by using whole face are also difficult to be applied to our purpose, the real-time processing.

On the other hand, the lips has not only the shape feature but also the anatomical color feature, which is described in Section 1.1. Hence, the lips vermilion color must be used as [5].

In order to solve three problems, which are listed above, we use the single template matching with a genetic algorithm as a matching process.

1.2.2 Input Images

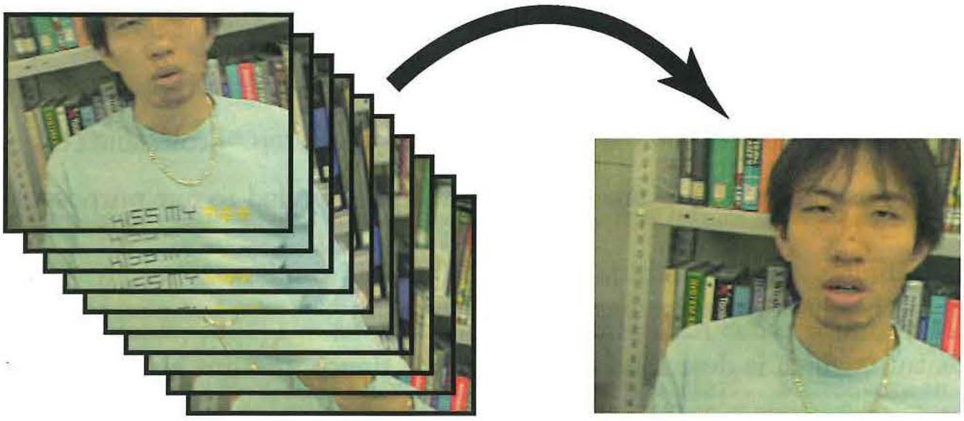


(a) (b)

Figure 1.2 Template acquisition: a) source of template image (240 × 180 pixels) ; b) template image (20 × 11 pixels).

At first, input images, a template image and target images, are prepared. Creating the template, the changes of subjects, scene, and lips must be considered. However, in this study, only the lips changes are focused, because this system will be applied to personal devices.

Generally speaking, it is difficult and time-consuming to create many templates of the various changes of lips. Therefore, only one template is created in this system. An example of the template is illustrated in Figure 1.2. The template image is acquired from a captured face image just before video sequence capturing for target images. In



(a) (b)

Figure 1.3 Target image: a) source video sequence ; b) target image

this proposed system, the basic shape of the template is a square so that both of the skin and the lips are contained, as shown in Figure 1.2. Because, a color difference between skin and lips is a very important feature from an anatomical viewpoint [16, 17, 18, 19], moreover this shape makes it easy to calculate. The template image is a closed mouth of the same subject as target images, because the application of this system is for personal devices.

Next, the source video sequence is captured with speech of a subject and the free camera motion, and target images are acquired from the source video sequence (refer to Figure 1.3). The lips in the target image has various changes as shown in Figure 1.3. Types of these changes are described in Section 3.2.

Chapter 2

Theoretical Background

In this chapter, theoretical background of this study, color space, feature of lips color, geometry, and genetic algorithm, are described.

2.1 Color Space

In this section, color spaces for the color data that is used in the proposed method are described. In our study, a modified Yxy color space that is based upon Yxy color space is used.

2.1.1 Device-Independent Color Spaces

Some color spaces can express color in a device-independent way. Whereas RGB colors vary with display, scanner and digital camera characteristics, device-independent colors are not dependent on any particular device and are meant to be true representations of colors as perceived by the human eye. These color representations, called device-independent color spaces, result from work carried out by the Commission Internationale de l'Éclairage (CIE) established in 1931. and for that reason are also called CIE-based color spaces [30].

The most common method of identifying color within a color space is a three-dimensional geometry. The three color attributes, hue, value, and chroma, are measured, assigned numeric values, and plotted within the color space.

For example, conversion from an RGB color space to a CMYK color space involves a number of variables. The type of printer or printing press, the paper stock, and the inks used all influence the balance between cyan, magenta, yellow, and black. In addition, different devices have different gamuts, or ranges of colors that they can produce. Because the colors produced by RGB and CMYK specifications are specific to a device, they're called device-dependent color spaces. Device color spaces enable the specification of color values that are directly related to their representation on a particular device.

Device-independent color spaces can be used as interchange color spaces to convert color data from the native color space of one device to the native color space of another device.

The CIE created a set of color spaces that specify color in terms of human perception. It then developed algorithms to derive three imaginary primary constituents of color–X, Y, and Z–that can be combined at different levels to produce all the color the human eye can perceive. The resulting color model, CIE XYZ, and other CIE color models form the basis for all color management systems. Although the RGB and CMYK values differ from device to device, human perception of color remains consistent across devices. Colors can be specified in the CIE-based color spaces in a way that is independent of the characteristics of any particular display or reproduction device. The goal of this standard is for a given CIE-based color specification to produce consistent results on different devices, up to the limitations of each device.

2.1.2 XYZ Color Space

There are several CIE-based color spaces, but all are derived from the fundamental XYZ space. In 1931 CIE defined a human "Standard Observer", based on measurements of the color-matching abilities of the average human eye. The XYZ space allows colors to be expressed as a mixture of the three tristimulus values X, Y, and Z. The term tristimulus comes from the fact that color perception results from the retina of the eye responding to three types of stimuli. After experimentation, the CIE set up a hypothetical set of primaries, XYZ, that correspond to the way the eye's retina behaves.

The Y primary is identical to Luminance, X and Z give color (chroma) information. This forms the basis of the CIE 1931 XYZ color space, which is fundamental to all colorimetry. Values are normally assumed to lie in the range $[0, 1]$. Colors are rarely specified in XYZ terms, it is far more common to use "chromaticity coordinates" which are independent of the Luminance (Y). Other device-independent color spaces based on XYZ space are used primarily to relate some particular aspect of color or some perceptual color difference to XYZ values [30, 31].

2.1.3 Yxy Color Space

Yxy space expresses the XYZ values in terms of x and y chromaticity coordinates, somewhat analogous to the hue and saturation coordinates of HSV space. The coordinates are shown in the following formulas, used to convert XYZ into Yxy [30, 31]:

$$Y = Y \tag{2.1}$$

$$x = \frac{X}{X+Y+Z} \tag{2.2}$$

$$y = \frac{Y}{X+Y+Z} \tag{2.3}$$

$$z = \frac{Z}{X+Y+Z} \tag{2.4}$$

Note that the Z tristimulus value is incorporated into the new coordinates and does not appear by itself. Since Y still correlates to the lightness of a color, the other aspects of the color are found in a combination of three axes: x, y, and z, with,

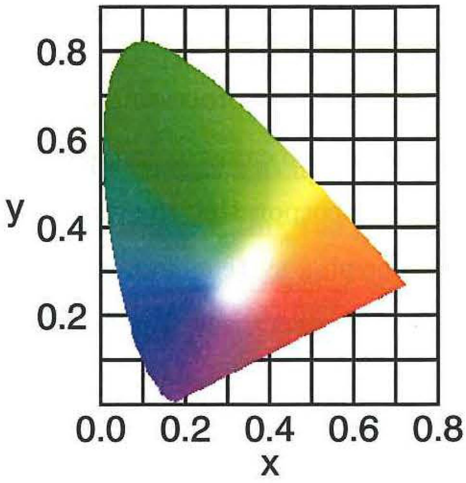


Figure 2.1 The chromaticity diagram

in broad terms, x representing the amount of redness in a color, y the amount of greenness and lightness (bright-to-dark), and z the amount of blueness. And this x and y is the chromaticity coordinates. This allows color variation in Yxy space to be plotted on a two-dimensional diagram. Figure 2.1 shows the layout of colors in the x and y plane of Yxy space.

2.1.4 Gamma Correction

In image processing, computer graphics, digital video and photography, the symbol γ represents a numerical parameter which describes the nonlinearity of the intensity reproduction. The cathode-ray tube (CRT) employed in modern computing system is nonlinear in the sense that the intensity of light reproduced at the screen of a CRT monitor is a nonlinear function of the voltage input. A CRT has a power law response to applied voltage. The light intensity produced on the display is proportional to the applied voltage raised to a power denoted by γ. Thus, the produced intensity by the CRT and the voltage applied on the CRT have the following relationship:

$$I_{int} = (v')^{\gamma} \tag{2.5}$$

The actual value of γ for a particular CRT may range from about 2.3 to 2.6 although most practitioners frequency claim values lower than 2.2 for video monitors [31].

The process of pre-computing for the nonlinearity by computing a voltage signal from an intensity value is called gamma correction. The function required is approximately a 0.45 power function. In image processing application, gamma correction is accomplished by analog circuits at the camera. In computer graphics, gamma correction is usually accomplished by incorporating the function into a frame buffer lookup table. Although in image processing systems gamma was originally used to refer to the nonlinearity of the CRT, it is generalized to refer to the nonlinearity of an entire image processing system. The γ value of an image or an image processing system can be calculated by multiplying the γ's of its individual components from the image capture stage to the display.

To compensate for the nonlinearity of the display (CRT), gamma correction with a power of $\left(\frac{1}{\gamma}\right)$ can be used so that the overall system γ is approximately 1.

In a video system, the gamma correction is applied to the camera for pre-computing the nonlinearity of the display. The gamma correction perform the following transfer function:

$$valtage' = (valtage)^{\frac{1}{\gamma}} \tag{2.6}$$

where *voltage* is the voltage generated by the camera sensors.

For color images, the linear values R, G and B values should be converted into nonlinear voltages R', G' and B' through the application of the gamma correction process. The color CRT will then convert R', G' and B' into linear red, green, blue light to reproduce the original color.

Gamma correction is usually performed in cameras, and thus, pixel values are in most cases nonlinear voltage. Thus, intensity values stored in the frame buffer of the computing device are gamma corrected on-the-fly by hardware look up tables on their way to the computer monitor display. Modern image processing systems utilize a wide variety of sources of color images, such as images captured by digital cameras, scanned images, digitized video frames and computer generated images. Digitized video frames usually have a gamma correction value between 0.5 and 0.45. Digital scanners assume an out put gamma in the range of 1.4 to 2.2 and they perform their gamma correction accordingly. For computer generated images the gamma correction value is usually unknown. In the absence of the actual gamma value the recommended gamma correction is 0.45 [31].

In summary, pixel values alone cannot specify the actual color. The gamma correction value used for capturing or generating the color image is needed. Thus, two images which have been captured with two cameras operating under different gamma correction values will represent colors differently even if the same primaries and the same white reference point are used.

2.1.5 Linear and Non-linear RGB Color Space

The image processing literature rarely discriminates between linear RGB and non-linear (R′G′B′) gamma corrected values. For example, in the JPEG and MPEG standards and in image filtering, non-linear RGB(R′G′B′) color values are implicit. Unacceptable results are obtained when JPEG or MPEG schemes are applied to linear RGB image data. On the other hand, in computer graphics, linear RGB values are implicitly used. Therefore, it is very important to understand the difference between linear and non-linear RGB values and be aware of which values are used in an image processing application. Hereafter, the notation R′G′B′ will be used for non-linear RGB values so that they can be clearly distinguished from the linear RGB values.

2.1.6 Linear RGB Color Space

As mentioned earlier, intensity is a measure, over some interval of the electromagnetic spectrum of the flow of power that is radiated from an object. Intensity is often called a linear light measure. The linear R value is proportional to the intensity of the physical power that is radiated from an object around the 700 [nm] band of the visible spectrum. Similarly, a linear G value corresponds to the 546.1 [nm] band and a linear B value corresponds to the 435.8 [nm] band. As a result the linear RGB space is device independent and used in some color management system to achieve color consistency across diverse devices.

The linear RGB values in the range $[0, 1]$ can be converted to the corresponding CIE XYZ value in the range $[0, 1]$ using the following matrix transformation [31]:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.4125 & 0.3576 & 0.1804 \\ 0.2127 & 0.7152 & 0.0722 \\ 0.0193 & 0.1192 & 0.9502 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{2.7}$$

The transformation from CIE XYZ values in the range $[0, 1]$ to RGB values in the range $[0.1]$ is defined by [31]:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 3.2405 & -1.5372 & -0.4985 \\ -0.9693 & 1.8760 & 0.0416 \\ 0.0556 & -0.2040 & 1.0573 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{2.8}$$

Alternatively, tristimulus XYZ values can be obtained from the linear RGB values through the following matrix [31]:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.490 & 0.310 & 0.200 \\ 0.117 & 0.812 & 0.011 \\ 0.000 & 0.010 & 0.990 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{2.9}$$

The linear RGB values are a physical representation of the chromatic light radiated from an object. However, the perceptual response of the human visual system to radiate red, green, and blue intensities is non-linear and more complex. The linear RGB space is, perceptually, highly non-uniform and not suitable for numerical analysis of the perceptual attributes. Thus, the linear RGB values are very rarely used to represent an image. On the contrary, non-linear R′G′B′ values are traditionally used in image processing applications such as filtering.

2.1.7 Non-linear RGB Color Space

When an image acquisition system, e.g. a video camera, is used to capture the image of an object, the camera is exposed to the linear light radiated from the object. The linear RGB intensities incident on the camera are transformed to non-linear RGB signals using gamma correction. The transformation to non-linear R′G′B′values in the range $[0, 1]$ from linear RGB values in the range $[0, 1]$ is defined by [31]:

$$R' = \begin{cases} 4.5R, & \text{if } R \leq 0.018 \\ 1.099R^{\frac{1}{\gamma C}}, & \text{otherwise} \end{cases} \tag{2.10}$$

$$G' = \begin{cases} 4.5G, & \text{if } R \leq 0.018 \\ 1.099G^{\frac{1}{\gamma C}}, & \text{otherwise} \end{cases} \tag{2.11}$$

$$B' = \begin{cases} 4.5B, & \text{if } R \leq 0.018 \\ 1.099B^{\frac{1}{\gamma C}}, & \text{otherwise} \end{cases} \tag{2.12}$$

where γC is known as the gamma factor of the camera or the acquisition device. The above transformation is commonly used in video cameras with $\gamma C = \frac{1}{0.45}$ ($\simeq 2.22$). The linear segment near low intensities minimizes the effect of sensor noise in practical cameras and scanners.

Thus, the digital values of the image pixel acquired from the object and stored within a camera or a scanner are the R′G′B′ values usually converted to the range of

0 to 255. Three bytes are then required to represent the three components,R', $G'$and B' values that are stored as image components. It is these non-linear R′G′B′ values that are stored as image data files in computers and are used in image processing application. The RGB symbols used in image processing literature usually refers to the R′G′B′ values and, therefore, care must be taken in color or space conversions and other relevant calculations.

Suppose the acquired image of an object needs to be displayed in a display device such as a computer monitor. Ideally, a user would like to see (perceive) the exact reproduction of the object. As pointed out, the image data is in R′G′B′ values. Signals (usually voltage) proportional to the R′G′B′ values will be applied to the red, green, and blue guns of the CRT (Cathode Ray Tube) respectively. The intensity of the red, green, and blue lights generated by the CRT is a non-linear function of the applied signal. The non-linearity of the CRT is a function of the electrostatics of the cathode and the grid of the electron gun. In order to achieve correct reproduction of intensities, an ideal monitor should invert the transformation at the acquisition device (camera) so that the intensities generated are identical to the linear RGB intensities that were radiated from the object and incident in the acquisition device. Only then will the perception of the displayed image be identical to the perceived object.

A power-law response, which inverts the non-linear (R′G′B′) values in the range $[0,1]$ back to linear RGB values in the range $[0,1]$, is defined by the following power function [31]:

$$R = \begin{cases} \frac{R'}{4.5}, & \text{if } R' \leq 0.018 \\ \left(\frac{R'+0.099}{1.099}\right)^{\gamma D}, & \text{otherwise} \end{cases} \tag{2.13}$$

$$G = \begin{cases} \frac{G'}{4.5}, & \text{if } G' \leq 0.018 \\ \left(\frac{G'+0.099}{1.099}\right)^{\gamma D}, & \text{otherwise} \end{cases} \tag{2.14}$$

$$B = \begin{cases} \frac{B'}{4.5}, & \text{if } B' \leq 0.018 \\ \left(\frac{B'+0.099}{1.099}\right)^{\gamma D}, & \text{otherwise} \end{cases} \tag{2.15}$$

The value of the power function, γD, is known as the gamma factor of the display device or CRT. Normal display devices have γD in the range of 2.2 to 2.45. For exact reproduction of the intensities, gamma factor of the display device must be equal to the gamma factor of the acquisition device ($\gamma C = \gamma D$). Therefore, a CRT with a gamma factor of 2.2 should correctly reproduce the intensities.

The transformations that tale place throughout the process of image acquisition to image display and perception are illustrated in Figure 2.2.

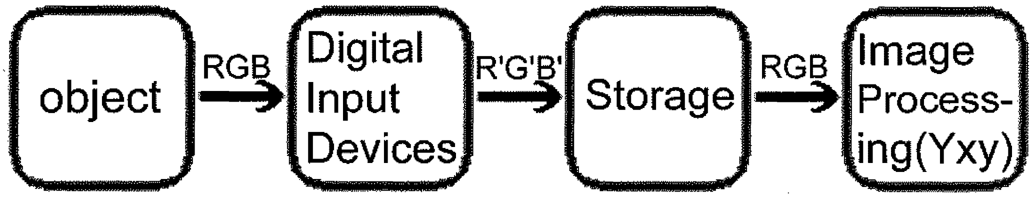


Figure 2.2 Transformation of intensities from image capture to image processing

In it obvious from the above discussion that the R′G′B′ space is a device dependent space. Suppose a color image, represented in the R′G′B′ space, is displayed on two computer monitors having different gamma factors. The red, green, and blue intensities produced by the monitors will not be identical and the displayed images might have different appearances. Device dependent spaces cannot be used if color consistency across various devices, such as display devices, printers, etc., is of primary concern. However. similar devices (e.g. two computer monitors) usually have similar gamma factors and in such cases device dependency might not be an important issue.

As mentioned before, the human visual system has a non-linear perceptual response to intensity, which is roughly logarithmic and is, approximately,the inverse of a conventional CRT's non-linearity. In other words, the perceived red, green, and blue intensities are approximately related to the R′G′B′ values. Due to this fact, computations involving R′G′B′ values have an approximate relation to the human color perception and the R′G′B′ space is less perceptually non-uniform relative to the CIE XYZ and linear RGB spaces. Hence, distance measures defined between the R′G′B′ values of two color vectors provide a computationally simple estimation of the error between them. This is very useful for real-time applications and systems in which computational resources are at premium.

However, the R′G′B′ space is not adequately uniform, and it cannot be used for accurate perceptual computations. In such instances, perceptually uniform color spaces (e.g. Yxy, L*u*v*, and L*a*b*) that are derived based on the attributes of human color perception are more desirable than the R′G′B′ space.

2.1.8 The modified Yxy Color Space

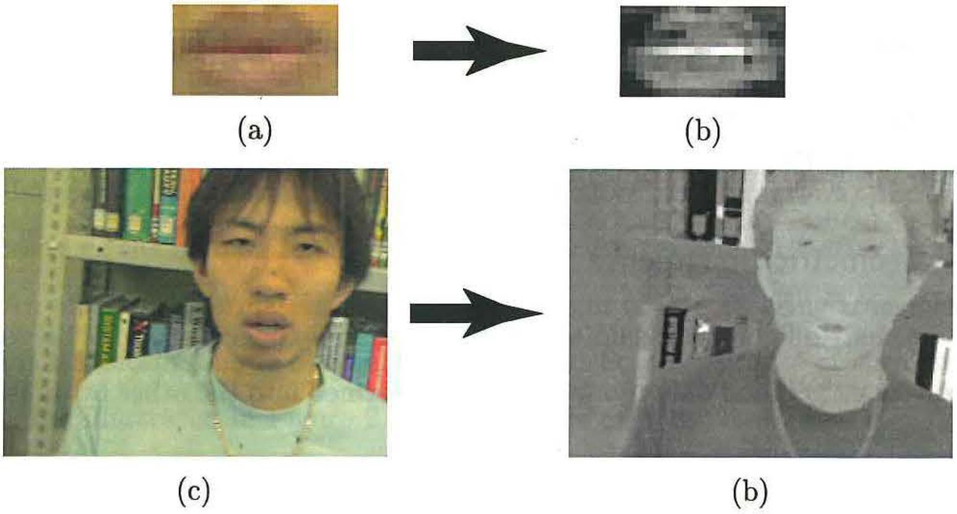
In our proposed method, we used a modified Yxy color space based on the x component of the Yxy color space. In a preliminary examination, we compared our methods using Gray scale and color data. Gray scale is easier to calculate and treat than color data. However, this comparison indicated that the use of color data yielded a better result. The color data we used is a modified Yxy color space based on the x component of Yxy color space. As mentioned earlier, Yxy color space is derived based on the attributes of human color perception, and the component x of Yxy color space represents the amount of redness in a color. Thus, the component x is suitable for lips extraction. Furthermore, for lips extraction, the blueness has little effect. Hence, we modify the transformation from linear RGB to CIE XYZ to eliminate blue component. In practice, the result was better in some auxiliary experiment.

The x component of the modified Yxy color space is obtained as described below. Image data which are captured by a digital video camera are sRGB values. These sRGB values are quantized in the range $[0, 255]$. Thus, at first, to obtain non-linear R′G′B′ values in the range $[0, 1]$, sRGB values are normalized. Next, the non-linear R′G′B′ values are transformed to linear RGB values by using formulas (2.10), (2.11) and (2.12). Then, our new XYZ values are obtained by transformation from the linear RGB values using the following our distinctive matrix transformation:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.4125 & 0.3576 & 0.0000 \\ 0.2127 & 0.7152 & 0.0000 \\ 0.0193 & 0.1192 & 0.0000 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{2.16}$$

This matrix transformation is based on formula 2.7. Finally, the component x value is obtained from these X, Y, Z. In the simulation, the component x value is multiplied by 1,000, because it is a small value (the range is $[0, 0.714]$ [32]).

2.2 Feature of Lips Color



(a) (b)

(c) (b)

Figure 2.3 Example of conversion to the x component: a) and c) source data, b) and d) the x component

The lips redness is common to the entire human race. The reason is that the redness is called the "vermillion border" and composed of nonkeratinized squamous epithelium that covers numerous capillaries, which give the lips characteristic color [16, 17, 18, 19].

For these reasons, we focus on the lips redness as main feature during detection. Therefore, the source color data (non-linear RGB data) is converted into the x component. This x component is one of the CIE-Yxy color space [31], and illustrates the chromaticity diagram (see Figure 2.1) with y component. The x component represents redness as shown in Figure 2.1. Moreover, lightness does not greatly influence on the x component, because it is independent of lightness [31]. For these merits, the x component is used in this study.

This Yxy color data is converted from the source data in the following order (refer to [31]): from non-linear RGB color space, linear RGB color space, XYZ color space, and finally the modified Yxy color space (refer to Section 2.1.8). Figure 2.3 shows the conversion from the source data to the x component data. The x component which is multiplied by 1000, is used as image data in our study. Comparing the modified x component with other color space values in our preliminary experiment, it achieves the relatively good result.

2.3 Geometry

The proposed method uses homogeneous coordinates. In this section, due to their importance, the advantages of using homogeneous coordinates is described.

2.3.1 Geometric Congruence

Two geometric figures are said to exhibit geometric congruence (or "be geometrically congruent") if and only if one can be transformed into the other by an isometry.

2.3.2 Similarity

A similarity is a transformation that preserves angles and changes all distances in the same ratio, called the ratio of magnification. A similarity can also be defined as a transformation that preserves ratios of distances. A similarity therefore transforms figures into similar figures. When written explicitly in terms of transformation matrices in three dimensions, similarities are commonly referred to as similarity transformations. Examples of similarities include the follows.

1. Central dilation: a transformation of lines to parallel lines that is not merely a translation.
2. Geometric Contraction: a transformation in which the scale is reduced.
3. Dilation: a transformation taking each line to a parallel line whose length is a fixed multiple of the length of the original line.
4. Expansion: a transformation in which the scale is increased.
5. Isometry: a transformation that preserves distances.
6. Reflection: a transformation in which all points are exchanged with their corresponding reflections in an infinite plane mirror.
7. Rotation: a transformation that preserves angles and distances.
8. Rotoinversion: reflection through the origin combined with a rotation.

9. Translation: a transformation consisting of a constant offset with no rotation or distortion.

2.3.3 Affine Transformation

An Affine Transformation is any transformation preserving collinearity (i.e., all points lying on a line initially still lie on a line after transformation) and ratios of distances (e.g., the midpoint of a line segment remains the midpoint after transformation). An affine transformation may also be thought of as a shearing transformation. An affine transformation is also called an affinity. Some examples of these transformations are illustrated in Figure 2.4. An example of linear non-affine is perspective projection.

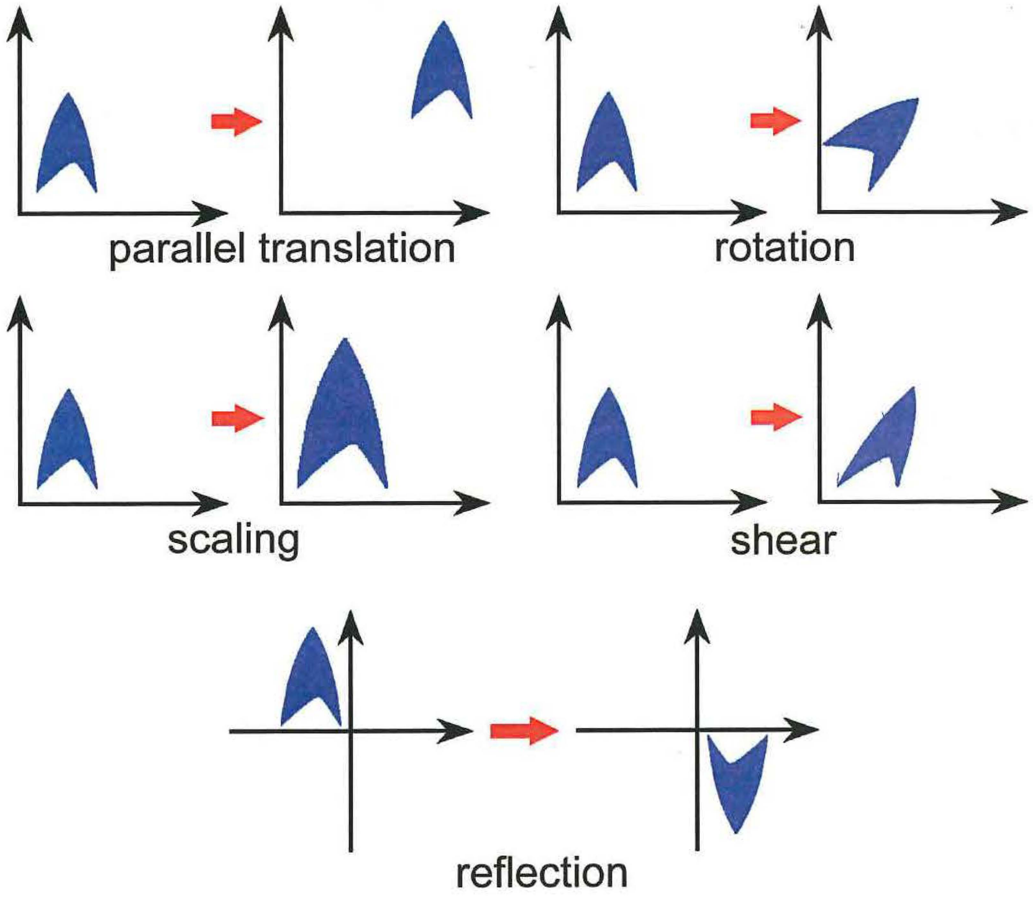


Figure 2.4 Examples of transformation

An affine transformation of $\boldsymbol{R^n}$ is a map $F : \boldsymbol{R^n} \to \boldsymbol{R^n}$ of the form

$$\boldsymbol{p}^* = F(\boldsymbol{p}) \tag{2.17}$$

$$F(\boldsymbol{p}) = A\boldsymbol{p} + \boldsymbol{q} \tag{2.18}$$

$$\begin{cases} x^* = ax + by + t_x \\ y^* = cx + dy + t_y \end{cases} \tag{2.19}$$

$$\begin{bmatrix} x^* \\ y^* \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x & t_y \end{bmatrix} \tag{2.20}$$

for all $\boldsymbol{p} \in \boldsymbol{R^n}$, where A is a linear transformation of $\boldsymbol{R^n}$. If $det(A) = 1$, the transformation is orientation-preserving; if $det(A) = -1$, it is orientation-reversing. The elements a, b, c, d, t_x and t_y define a linear transformation (more on this later).

Any transformation such as Figure 2.4 can be written as formula (2.20). Examples of the formulas which represent affine transformations are described below.

■Rotation To rotate a point counterclockwise by some angle θ:

$$\begin{cases} x^* = (\cos\theta)\, x - (\sin\theta)\, y \\ y^* = (\sin\theta)\, x + (\cos\theta)\, y \end{cases} \tag{2.21}$$

$$\begin{bmatrix} x^* \\ y^* \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{2.22}$$

■Scaling Set $b = c = 0$, and let a and d take on any value. This gives us some matrix:

$$\begin{cases} x^* = (s_x)\, x \\ y^* = (s_y)\, y \end{cases} \tag{2.23}$$

$$\begin{bmatrix} x^* \\ y^* \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{2.24}$$

■Shear Set $a = d = 1$, and let b and c take on any value as follows:

$$\begin{cases} x^* = x + (sh_x)\, y \\ y^* = (sh_y)\, x + y \end{cases} \tag{2.25}$$

$$\begin{bmatrix} x^* \\ y^* \end{bmatrix} = \begin{bmatrix} 1 & ah_x \\ xh_y & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{2.26}$$

This is called a shear.

■Reflection Still fix $b = c = 0$, and let a and d take on -1 values:

$$\begin{cases} x^* = -x \\ y^* = -y \end{cases} \tag{2.27}$$

$$\begin{bmatrix} x^* \\ y^* \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{2.28}$$

This is reflection with respect to the origin.

■Parallel Translation Set $a = d = 1$, $b = c = 0$ and let t_x and t_y in formula (2.19) and (2.20) take on any value as follows:

$$\begin{cases} x^* = x + t_x \\ y^* = y + t_y \end{cases} \tag{2.29}$$

$$\begin{bmatrix} x^* \\ y^* \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \tag{2.30}$$

Geometric contraction, expansion, dilation, reflection, similarity transformations, spiral similarities, rotation, and parallel translation are all affine transformations and can be combined by matrix multiplication as follow:

$$\begin{bmatrix} x^* \\ y^* \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} e & f \\ g & h \end{bmatrix} \begin{bmatrix} i & j \\ k & l \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{2.31}$$

2.3.4 Homogeneous Coordinates

In graphics systems, points are usually represented using homogeneous coordinates [33]. For example, a point (x, y) in 2 dimensional space is represented by the triple $(x, y, 1)$; a point (x, y, z) in 3-dimensional space is represented by the quadruple $(x, y, z, 1)$.

Linear transformations of a vector space can be represented by matrices (refer to above Section 2.3.3). A linear transformation can be applied to a point by multiplying the point (viewed as a column vector) by the matrix which represents the linear transformation. We would like to apply this to basic types of transformation such as translation, rotation, scaling, shearing, and reflection. However there's a problem. Linear transformations of a vector space always map the origin to the origin. We can see this easily by seeing what happens when we multiply a 2 × 2 matrix by the $(0,0)$ column vector. However, a translation by the vector (dx, dy) maps the origin to the point (dx, dy). Therefore, translation cannot be a linear transformation, and cannot be represented by the simple 2 × 2 matrix multiplication. Another non-linear transformations is projective transformation. Other transformations are linear transformations.

To solve this problem, homogeneous coordinates are used. Suppose we have a point (x, y) in the Euclidean plane. To represent this same point in the projective plane, we simply add a third coordinate of 1 at the end: $(x, y, 1)$. In general, a point in an n-dimensional Euclidean space is represented as a point in an $(n+1)$-dimensional projective space. Overall scaling is unimportant, so the point $(x, y, 1)$ is the same as the point, $(\alpha x, \alpha y, \alpha)$ for any non-zero α. In other words,

$$[X, Y, w]^T = [\alpha X, \alpha Y, \alpha w]^T \tag{2.32}$$

for any $\alpha \neq 0$ (Thus the point $(0, 0, 0)$ is disallowed). Because scaling is unimportant, the coordinates $[X, Y, w]^T$ are called the homogeneous coordinates of the point. If $w \neq 0$, (wx, wy, w) corresponds to the point $(x/w, y/w)$ in the original Euclidean plane; $(x, y, 0)$ corresponds to the point at infinity corresponding to the direction of the line passing through $(0, 0)$ and (x, y) (see Figure 2.5).

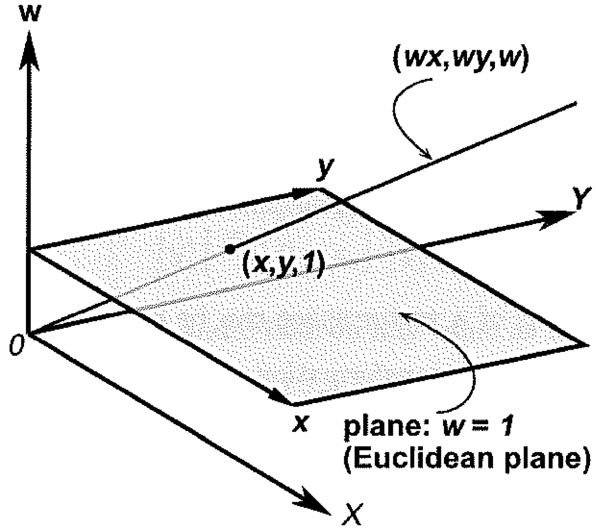


Figure 2.5 Image of homogeneous coordinates

The affine transformation formula (2.20) is simplified by homogeneous coordinates as follow:

$$\begin{bmatrix} X^* \\ Y^* \\ w^* \end{bmatrix} = \begin{bmatrix} a & b & t_x \\ c & d & t_y \\ p & q & s \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2.33}$$

where

$$\begin{cases} X^* = w^* x^* \\ Y^* = w^* y^* \end{cases} \tag{2.34}$$

or, this formula is generalize as follow:

$$\begin{bmatrix} X^* \\ Y^* \\ w^* \end{bmatrix} = \begin{bmatrix} a & b & t_x \\ c & d & t_y \\ p & q & s \end{bmatrix} \begin{bmatrix} X \\ Y \\ w \end{bmatrix} \tag{2.35}$$

where

$$\begin{cases} X = wx \\ Y = wy \end{cases} \tag{2.36}$$

The elements a, b, c, d, p, q, s, t_x and t_y define a homogeneous transformation. Any transformation such as Figure 2.4 can be written as formula (2.33) and (2.35).

Examples of the formulas which represent homogeneous transformations are described below.

■**Rotation** To rotate a point counterclockwise by some angle θ:

$$\begin{bmatrix} x^* \\ y^* \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2.37}$$

■**Scaling** Set $b = c = p = q = t_x = t_y = 0$, and let a and d take on any value. This gives us some matrix:

$$\begin{bmatrix} x^* \\ y^* \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2.38}$$

■**Shear** Set $p = q = t_x = t_y = 0$, $a = d = 1$, and let b and c take on any value as following:

$$\begin{bmatrix} x^* \\ y^* \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & sh_x & 0 \\ sh_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2.39}$$

This is called a shear.

■Reflection Still fix $b = c = p = q = t_x = t_y = 0$, and let a and d take on -1 values:

$$\begin{bmatrix} x^* \\ y^* \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2.40}$$

This is reflection with respect to the origin.

■Parallel Translation Set $a = d = 1$, $b = c = p = q = 0$ and let t_x and t_y in formula (2.33) and (2.35) take on any value as following:

$$\begin{bmatrix} x^* \\ y^* \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2.41}$$

Transformations can be combined by matrix multiplication.

$$\begin{bmatrix} x^* \\ y^* \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & t_{x1} \\ c & d & t_{y1} \\ p_1 & q_1 & s_1 \end{bmatrix} \begin{bmatrix} e & f & t_{x2} \\ g & h & t_{y2} \\ p_2 & q_2 & s_2 \end{bmatrix} \begin{bmatrix} i & j & t_{x3} \\ k & l & t_{y3} \\ p_3 & q_3 & s_3 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2.42}$$

Matrices are a convenient and efficient way to represent a sequence of transformations.

Any affine transformation can be expressed as a combination of these. We can combine homogeneous transforms by multiplication. Now any sequence of translate, scale, rotate and etc. operations can be collapsed into a single homogeneous matrix. For this example the matrix multiplication as follow:

$$\begin{bmatrix} x^* \\ y^* \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2.43}$$

2.3.5 Topology

The proposed method uses homeomorphism for varying lips shapes at the moment of speech. In this section, topology and homeomorphism which is a property of topology are described.

■The Basis of Topology Topology is the mathematical study of properties of objects which are preserved through deformations, twistings, and stretchings. (Tearing, however, is not allowed.) A circle is topologically equivalent to an ellipse (into which it can be deformed by stretching) and a sphere is equivalent to an ellipsoid. Continuing along these lines, the space of all positions of the minute hand on a clock is topologically equivalent to a circle (where space of all positions means "the collection of all positions"). Similarly, the space of all positions of the minute and hour hands is equivalent to a torus. The space of all positions of the hour, minute and second hands form a four-dimensional object that cannot be visualized quite as simply as the former objects since it cannot be placed in our three-dimensional world, although it can be visualized by other means.

There is more to topology, though. Topology began with the study of curves, surfaces, and other objects in the plane and three-space. One of the central ideas in topology is that spatial objects like circles and spheres can be treated as objects in their own right, and knowledge of objects is independent of how they are "represented" or "embedded" in space. For example, the statement "if you remove a point from a circle, you get a line segment" applies just as well to the circle as to an ellipse, and even to tangled or knotted circles, since the statement involves only topological properties.

Topology has to do with the study of spatial objects such as curves, surfaces, the space we call our universe, the space-time of general relativity, fractals, knots, manifolds (objects with some of the same basic spatial properties as our universe), phase spaces that are encountered in physics (such as the space of hand-positions of a clock), symmetry groups like the collection of ways of rotating a top, etc.

Topology can be used to abstract the inherent connectivity of objects while ignoring their detailed form. For example, the figures in Figure 2.6 illustrate the connectivity of a number of topologically distinct surfaces. In these figures, parallel edges drawn in solid join one another with the orientation indicated with arrows, so corners labeled with the same letter correspond to the same point, and dashed lines show edges that remain free. The above figures correspond to the disk (plane), Klein bottle, Möbius strip, real projective plane, sphere, torus, and tube. The labels are often omitted in such diagrams since they are implied by connection of parallel lines with the orientations indicated by the arrows.

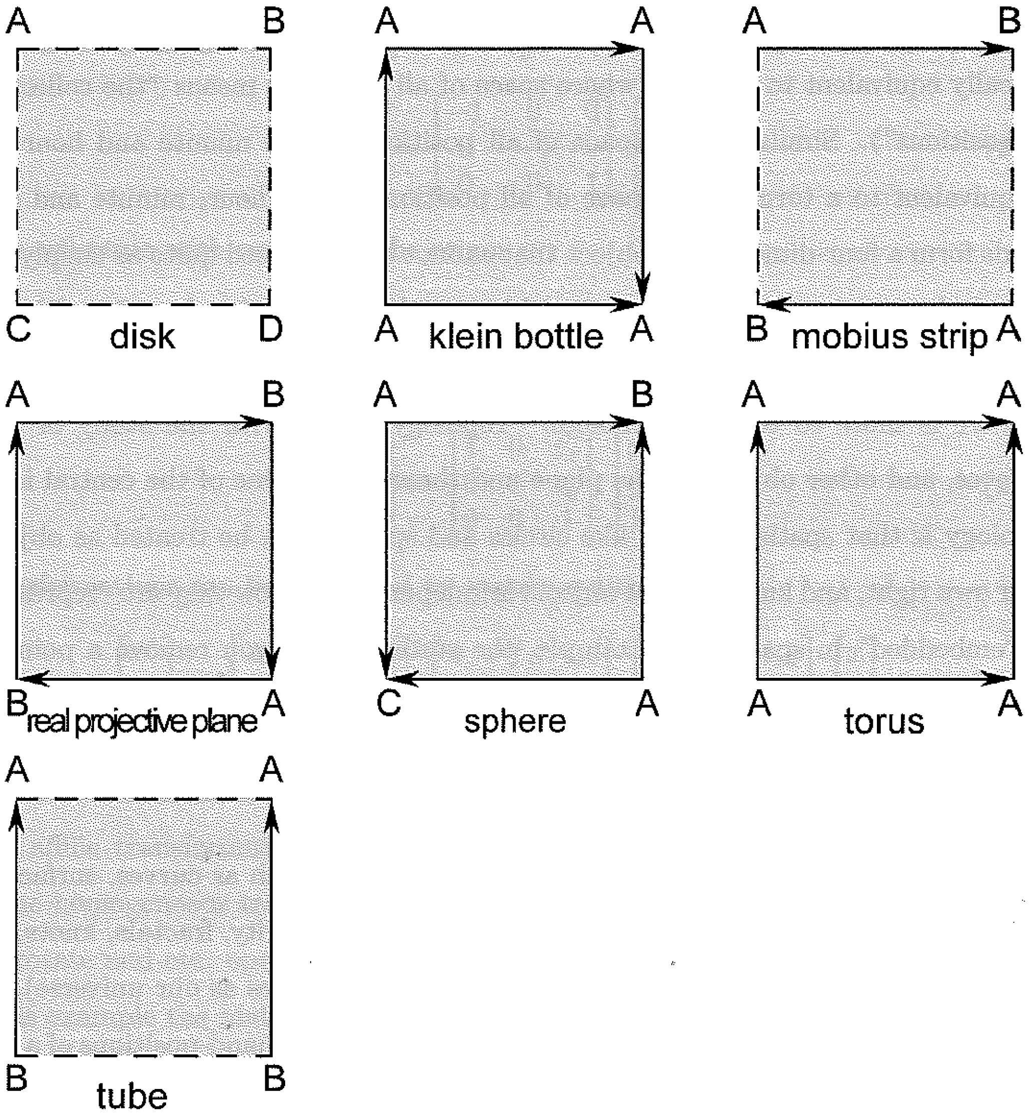


Figure 2.6 Examples of the inherent connectivity of objects

The "objects" of topology are often formally defined as topological spaces. If two objects have the same topological properties, they are said to be homeomorphic (although, strictly speaking, properties that are not destroyed by stretching and distorting an object are really properties preserved by isotopy, not homeomorphism; isotopy has to do with distorting embedded objects, while homeomorphism is intrinsic).

Topology is divided into algebraic topology (also called combinatorial topology), differential topology, and low-dimensional topology.

■**Homeomorphism** Homeomorphism is an equivalence relation and one-to-one correspondence between points in two geometric figures or topological spaces which is continuous in both directions, also called a continuous transformation. A homeomorphism which also preserves distances is called an isometry. Affine transformations are another type of common geometric homeomorphism. Properties preserved by transformations are listed below:

Euclidean shape and size, e.g., translation

Similarity shape but not necessarily size, e.g., scaling

Affine affine properties such as parallelism but not necessarily angles or scale, e.g., rotations

Projective projective properties such as line relationships

Topological topological properties such as connectivity

The similarity in meaning and form of the words "homomorphism" and "homeomorphism" is unfortunate and a common source of confusion.

■**Homeomorphic** There are two possible definitions:

1. Possessing similarity of form,
2. Continuous, one-to-one, onto, and having a continuous inverse.

The most common meaning is possessing intrinsic topological equivalence. Two objects are homeomorphic if they can be deformed into each other by a continuous, invertible mapping. Such a homeomorphism ignores the space in which surfaces are embedded, so the deformation can be completed in a higher dimensional space than the surface was originally embedded. Mirror images are homeomorphic, as are Möbius strip with an even number of half-twists, and Möbius strip with an odd number of half-twists.

2.3.6 Geometric Transformation Group

In this part, we would like to make certain of geometric transformation group [34]. Hierarchy structure of transformation group [34] is illustrated in Figure 2.7. In this

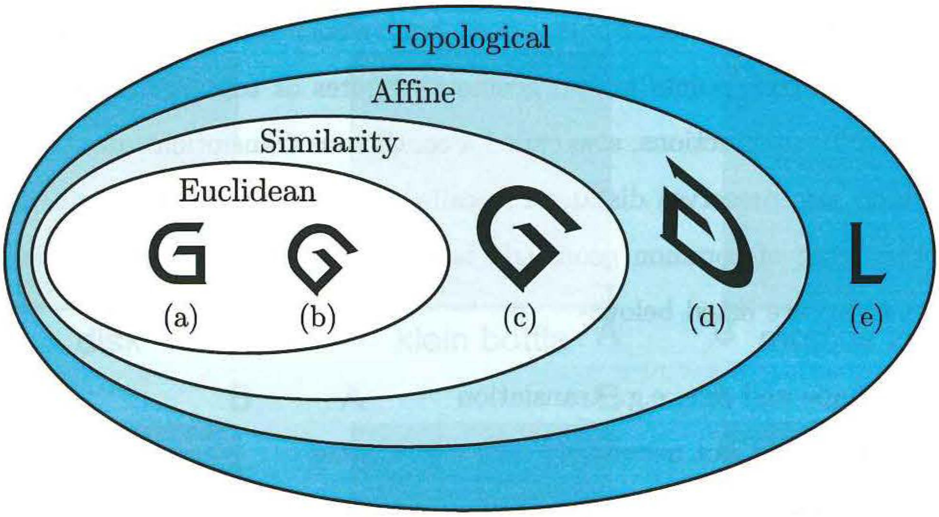


Figure 2.7 Hierarchy structure of transformation group: a) base ; b) rotation and translation (Euclidean transformation group) ; c) scaling (similarity) ; d) reflection and shearing (affine) ; e) connectivity (topological).

Figure 2.7, "G" and "L" are considered as graphic symbols. The base shape is Figure 2.7(a). The most inside group is Euclidean transformation. Figure 2.7(b) is a result of rotation. The second is similarity transformation group. Figure 2.7(c) is a result of scaling. The third is affine transformation group. Figure 2.7(d) is a result of shear and reflection. The most outside is topological transformation group. A connectivity of all "G" and "L" are same, therefore these relation is homeomorphic.

2.4 Genetic Algorithm

2.4.1 Overview of Evolutionary Computation

Evolutionary algorithm is an umbrella term used to describe computer-based problem solving systems which use computational models of some of the known mechanisms of evolution as key elements in their design and implementation. A variety of evolutionary algorithms have been proposed. The major ones are: genetic algorithms, evolutionary programming, evolution strategies, classifier systems, and genetic programming. They all share a common conceptual base of simulating the evolution of individual structures via processes of selection, mutation, and reproduction. The processes depend on the perceived performance of the individual structures as defined by an environment.

More precisely, evolutionary algorithms maintain a population of structures, that evolve according to rules of selection and other operators, that are referred to as "search operators", (or genetic operators), such as recombination and mutation. Each individual in the population receives a measure of its fitness in the environment. Reproduction focuses attention on high fitness individuals, thus exploiting (cf. exploitation) the available fitness information. Recombination and mutation perturb those individuals, providing general heuristics for exploration. Although simplistic from a biologist's viewpoint, these algorithms are sufficiently complex to provide robust and powerful adaptive search mechanisms [35].

2.4.2 Basis of a Genetic Algorithm

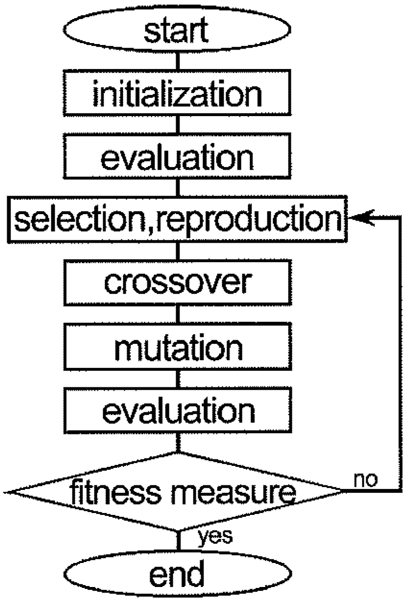


Figure 2.8 Flow chart of a simple genetic algorithm

The genetic algorithm is a type of evolutionary computation devised by John Holland [36]. The genetic algorithm is a model of machine learning which derives its behavior from a metaphor of some of the mechanisms of evolution in nature. This is done by the creation within a machine of a population of individuals represented by chromosomes, in essence a set of character strings that are analogous to the base-4 chromosomes that we see in our own DNA. The individuals in the population then go through a process of simulated "evolution".

Genetic algorithms are used for a number of different application areas. An example of this would be multidimensional optimization problems in which the character string of the chromosome can be used to encode the values for the different parameters being optimized.

In practice, therefore, we can implement this genetic model of computation by having arrays of bits or characters to represent the chromosomes. Simple bit manipulation operations allow the implementation of crossover, mutation and other operations. Although a substantial amount of research has been performed on variable–length strings and other structures, the majority of work with genetic algorithms is focussed on fixed-length character strings. We should focus on both this aspect of fixed-lengthness and the need to encode the representation of the solution being sought as a character string, since these are crucial aspects that distinguish genetic programming, which does not have a fixed length representation and there is typically no encoding of the problem.

When the genetic algorithm is implemented it is usually done in a manner that involves the following cycle: Evaluate the fitness of all of the individuals in the population. Create a new population by performing operations such as crossover, fitness-proportionate reproduction and mutation on the individuals whose fitness has just been measured. Discard the old population and iterate using the new population. A simple genetic algorithm which is mentioned above can be seen in Figure 2.8.

One iteration of this loop is referred to as a generation. There is no theoretical reason for this as an implementation model. Indeed, we do not see this punctuated behavior in populations in nature as a whole, but it is a convenient implementation model.

The first generation (generation 0) of this process operates on a population of randomly generated individuals. From there on, the genetic operations, in concert with the fitness measure, operate to improve the population.

2.4.3 Techniques of a Genetic Algorithm

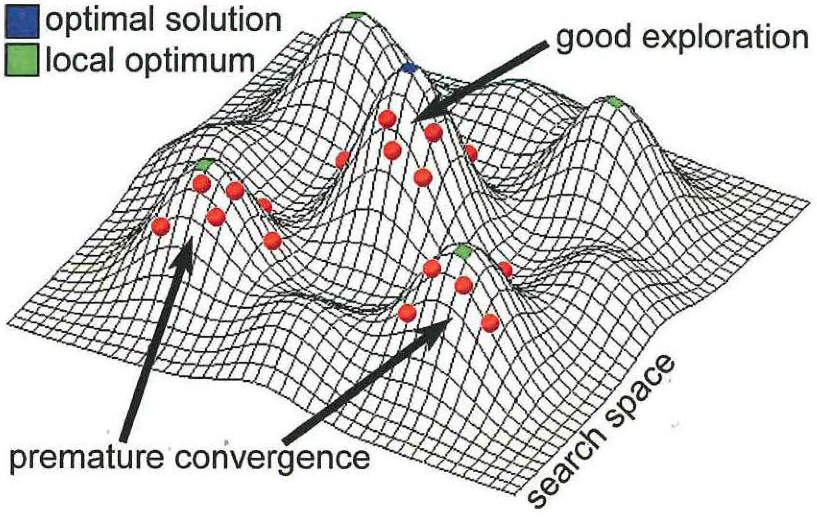


Figure 2.9 Image of exploration

The most important thing of the optimization method such as the genetic algorithms is "to arrive quickly at a better solution". On the contrary, unless this is satisfied, an exploration result in failure. A main cause of this failure is "to finish the process with arrival at a wrong solution". This wrong solution is called "local optimum".

When we try to maximize a function with multiple local maximums (such as a sine-based function, and a example is illustrated in Figure 2.9), then we run into problems. What happens is that the GA starts to explore a few of the local maximums and converges on the solution too early–rather than finding the absolute maximum, one or two of the strings perform so well initially that they reproduce numerous times and the genetic diversity of the population is limited. Thus the GA may come to the conclusion that one of the hills it is exploring is the maximum value when in fact it is not because the population does not contain enough diversity to perform well relatively on the other peaks. This is called "premature convergence". These example are illustrated in Figure 2.9. To aim towards solving this problem, many methods is proposed. The principal methods are the fitness scaling, rules of selection, the genetic operations.

■ **Fitness Scaling** Selection is the process of choosing individuals for reproduction in an evolutionary algorithm. One popular form of selection is called proportional selection. As the name implies, this approach involves creating a number of offspring in proportion to an individual's fitness. This approach was proposed and analyzed by Holland [36] and has been used widely in many implementations of evolutionary algorithms. Besides having some interesting mathematical properties, proportional selection provides a natural counterpart in artificial evolutionary systems to the usual practice in population genetics of defining an individual's fitness in terms of its number of offspring. For clarity of discussion, it is convenient to decompose the selection process into distinct steps, namely:

i. map the objective function to fitness,

ii. create a probability distribution proportional to fitness, and

iii. draw samples from this distribution.

These steps are discussed below.

The evaluation process of individuals in an evolutionary algorithm begins with the user-defined objective function,

$$f : A_x \to \boldsymbol{R} \tag{2.44}$$

where A_x is the object variable space.

The objective function typically measures some cost to be minimized or some reward to be maximized. The definition of the objective function is, of course, application dependent. The characterization of how well evolutionary algorithms perform on different classes of objective functions is a topic of continuing research. However, a few general design principles are clear when using an evolutionary algorithm.

i. The objective function must reflect the relevant measures to be optimized. Evolutionary algorithms are notoriously opportunistic, and there are several known instances of an algorithm optimizing the stated objective function, only to have the user realize that the objective function did not actually represent the intended measure.

ii. The objective function should exhibit some regularities over the space defined by the selected representation.

iii. The objective function should provide enough information to drive the selective pressure of the evolutionary algorithm. For example, "needle-in-a-haystack" functions, i.e. functions that assign nearly equal value to every candidate solution except the optimum, should be avoided.

The fitness function

$$\Phi : A_x \to \boldsymbol{R}_+ \tag{2.45}$$

maps the raw scores of the objective function to a non-negative interval. The fitness function is often a composition of the objective function and a scaling function g:

$$\Phi(a_i(t)) = g(f(a_i(t))) \tag{2.46}$$

where $a_i(t) \in A_x$.

Such a mapping is necessary if the goal is to minimize the objective function, since higher fitness values correspond to lower objective values in this case. For example, one fitness function that might be used when the goal is to minimize the objective function is

$$\Phi(a_i(t)) = f_{\max} - f(a_i(t)) \tag{2.47}$$

where $f_{\max}$ is the maximum value of the objective function.

If the global maximum value of the objective function is unknown, an alternative is

$$\Phi(a_i(t)) = f_{\max}(t) - f(a_i(t)) \tag{2.48}$$

where $f_{\max}(t)$ is the maximum observed value of the objective function up to time t.

There are many other plausible alternatives, such as

$$\Phi(a_i(t)) = \frac{1}{1 + f(a_i(t)) - f_{\min}(t)} \tag{2.49}$$

where $f_{\min}(t)$ is the minimum observed value of the objective function up to time t.

For maximization problems, this becomes

$$\Phi(a_i(t)) = \frac{1}{1 + f_{\max}(t) - f(a_i(t))} \tag{2.50}$$

Note that the latter two fitness functions yield a range of $(0, 1]$.

As an evolutionary algorithm progresses, the population often becomes dominated by high-performance individuals with a narrow range of objective values. In this case, the fitness functions described above tend to assign similar fitness values to all members of the population, leading to a loss in the selective pressure toward the better individuals. To address this problem, fitness scaling methods that accentuate small differences in objective values are often used in order to maintain a productive level of selective pressure. The representative fitness scaling are such as following [37]:

1. Linear Scaling
2. Sigma Scaling
3. Power Law Scaling

■Linear Scaling One approach to fitness scaling is to define the fitness function as a time-varying linear transformation of the objective value, for example

$$\Phi(a_i(t)) = \alpha f(a_i(t)) - \beta(t) \tag{2.51}$$

where α is $+1$ for maximization problems and $+1$ for minimization problems, and $\beta(t)$ represents the worst value seen in the last few generations.

Since $\beta(t)$ generally improves over time, this scaling method provides greater selection pressure later in the search. This method is sensitive, however, to "lethals", poorly performing individuals that may occasionally arise through crossover or mutation. Smoother scaling can be achieved by defining $\beta(t)$ as a recency-weighted running average of the worst observed objective values, for example

$$\beta(t) = \delta\beta\,(t-1) + (1-\delta)\,(f_{\text{worst}}(t)) \tag{2.52}$$

where δ is an update rate of, say, 0.1, and $f_{\text{worst}}(t)$ is the worst objective value in the population at time t.

■Sigma Scaling Sigma scaling is based on the distribution of objective values within the current population. It is defined as follows:

$$\Phi(a_i(t)) = \begin{cases} f(a_i(t)) - (\bar{f}(t) - c\sigma_f(t)) & \text{if } f(a_i(t)) > (\bar{f}(t) - c\sigma_f(t)) \\ 0 & \text{otherwise} \end{cases} \tag{2.53}$$

where $\bar{f}(t)$ is the mean objective value of the current population, $\sigma_f(t)$ is the (sample) standard deviation of the objective values in the current population, and c is a constant, say $c = 2$.

The idea is that $\bar{f}(t) - c\sigma_f(t)$ represents the least acceptable objective value for any reproducing individual. As the population improves, this statistic tracks the improvement, yielding a level of selective pressure that is sensitive to the spread of performance values in the population.

■Power Law Scaling Fitness scaling methods based on power laws have also been proposed. A fixed transformation of the form

$$\Phi(a_i(t)) = f(a_i(t))^k \tag{2.54}$$

where k is a problem-dependent parameter.

■Rules of Selection Selection is one of the main operators used in evolutionary algorithms. The primary objective of the selection operator is to emphasize better solutions in a population. This operator does not create any new solution, instead it selects relatively good solutions from a population and deletes the remaining, not-so-good, solutions. Thus, the selection operator is a mix of two different concepts–reproduction and selection. When one or more copies of a good solution are reproduced, this operation is called reproduction. Multiple copies of a solution are placed in a population by deleting some inferior solutions. This concept is known as selection. Although some evolutionary computation studies use both these concepts simultaneously, some studies use them separately.

The identification of good or bad solutions in a population is usually accomplished according to a solution's fitness. The essential idea is that a solution having a better fitness must have a higher probability of selection. However, selection operators differ in the way the copies are assigned to better solutions. Some operators sort the population according to fitness and deterministically choose the best few solutions, whereas some operators assign a probability of selection to each solution according to fitness and make a copy using that probability distribution. In the probabilistic selection operator, there is some finite, albeit small, probability of rejecting a good solution and choosing a bad solution. However, a selection operator is usually designed in a way so that the above is a low-probability event. There is, of course, an advantage of allowing this stochasticity (or flexibility) in the evolutionary algorithms. Due to a small initial population or an improper parameter choice or in solving a complex nonlinear fitness function, the best few individuals in a finite population may sometimes represent a suboptimal region. If a deterministic selection operator is used, these seemingly good individuals in the population will be emphasized and the population may finally converge to a wrong solution. However, if a stochastic selection operator is used, diversity in the population will be maintained by occasionally choosing not-so-good solutions. This event may prevent evolutionary computation algorithms from making a hasty decision in converging to a wrong solution.

In the following some of the popular selection operators are described.

■Roulette Wheel Selection The simplest selection scheme is roulette-wheel selection, also called stochastic sampling with replacement. This is a stochastic algorithm and involves the following technique:

The individuals are mapped to contiguous segments of a line, such that each individual's segment is equal in size to its fitness. A random number is generated and the individual whose segment spans the random number is selected. The process is repeated until the desired number of individuals is obtained (called mating population). This technique is analogous to a roulette wheel with each slice proportional in size to the fitness, see Figure 2.10.

Table 2.1 shows the selection probability for 11 individuals, linear ranking and selective pressure of 2 together with the fitness value. Individual 1 is the most fit individual and occupies the largest interval, whereas individual 10 as the second least fit individual has the smallest interval on the line (see Figure 2.10). Individual 11, the least fit interval, has a fitness value of 0 and get no chance for reproduction.

Table. 2.1 Selection probability and fitness value

Number of individual	1	2	3	4	5	6	7	8	9	10	11
fitness value	2.0	1.8	1.6	1.4	1.2	1.0	0.8	0.6	0.4	0.2	0.0
selection probability	0.18	0.16	0.15	0.13	0.11	0.09	0.07	0.06	0.03	0.02	0.0

For selecting the mating population the appropriate number of uniformly distributed random numbers (uniform distributed between 0.0 and 1.0) is independently generated.

sample of 6 random numbers: 0.81, 0.32, 0.96, 0.01, 0.65, 0.42

Figure 2.10 shows the selection process of the individuals for the example in Table 2.1 together with the above sample trials.

after selection the mating population consists of the individuals: 1, 2, 3, 5, 6, 9

The roulette-wheel selection algorithm provides a zero bias but does not guarantee minimum spread.

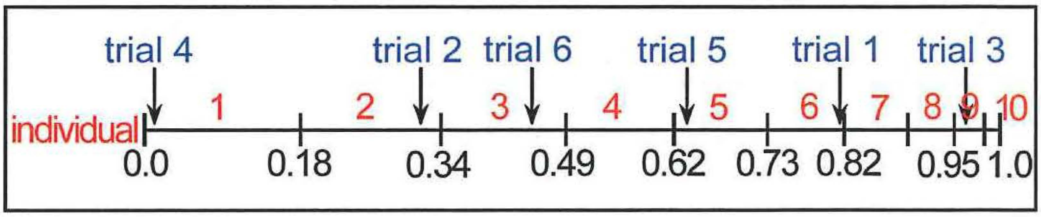


Figure 2.10 Image of the roulette wheel selection

■**Rank-based Fitness Assignment** Selection is the process of choosing individuals for reproduction or survival in an evolutionary algorithm. Rank-based selection or ranking means that only the rank ordering of the fitness of the individuals within the current population determines the probability of selection. As discussed in Section 2.4.3, the selection process may be decomposed into distinct steps:

i. Map the objective function to fitness.
ii. Create a probability distribution based on fitness.
iii. Draw samples from this distribution.

Ranking simplifies step i, the mapping from the objective function f to the fitness function Φ. All that is needed is

$$\Phi(a_i) = \delta f(a_i) \tag{2.55}$$

where δ is $+1$ for maximization problems and -1 for minimization problems.

Ranking also eliminates the need for fitness scaling, since selection pressure is maintained even if the objective function values within the population converge to a very narrow range, as often happens as the population evolves.

Step ii discussed here, the creation of the selection probability distribution based on fitness. The final step iii is independent of the selection method, and the stochastic universal sampling algorithm is an appropriate sampling procedure.

Besides its simplicity, other motivations for using rank-based selection include:

i. Under proportional selection, a "super" individual, i.e. an individual with vastly superior objective value, might completely take over the population in a single generation unless an artificial limit is placed on the maximum number of offspring for any individual. Ranking helps prevent premature convergence due to "super" individuals, since the best individual is always assigned the same selection probability, regardless of its objective value.

ii. Ranking may be a natural choice for problems in which it is difficult to precisely specify an objective function, e.g. if the objective function involves a person's subjective preference for alternative solutions. For such problems it may make little sense to pay too much attention to the exact values of the objective function, if exact values exist at all.

The various forms of linear and nonlinear ranking algorithms are mentioned below.

■**Linear Ranking** Linear ranking assigns a selection probability to each individual that is proportional to the individual's rank (where the rank of the least fit is defined to be zero and the rank of the most fit is defined to be $\mu - 1$, given a population of size μ). For a generational algorithm, linear ranking can be implemented by specifying a single parameter, β_{rank}, the expected number of offspring to be allocated to the best individual during each generation. The selection probability for individual i is then defined as follows:

$$\Pr_{\text{lin\_rank}}(i) = \frac{\alpha_{\text{lrank}} + [\text{rank}(i)/(\mu - 1)](\beta_{\text{rank}} - \alpha_{\text{rank}})}{\mu} \tag{2.56}$$

where α_{rank} is the number of offspring allocated to the worst individual.

The sum of the selection probabilities is then

$$\sum_{i=0}^{\mu-1} \frac{\alpha_{\text{lrank}} + [\text{rank}(i)/(\mu - 1)](\beta_{\text{rank}} - \alpha_{\text{rank}})}{\mu}$$

$$= \alpha_{\text{lrank}} + \frac{\beta_{\text{rank}} - \alpha_{\text{rank}}}{\mu(\mu - 1)} \sum_{i=0}^{\mu-1} i \tag{2.57}$$

$$= \alpha_{\text{rank}} + \frac{1}{2}(\beta_{\text{rank}} - \alpha_{\text{rank}}) \tag{2.58}$$

$$= \frac{1}{2}(\beta_{\text{rank}} + \alpha_{\text{rank}}) \tag{2.59}$$

It follows that $\alpha_{\text{rank}} = 2 - \beta_{\text{rank}}$, and $1 \leq \beta_{\text{rank}} \leq 2$. That is, the expected number of offspring of the best individual is no more than twice that of the population average. This shows how ranking can avoid premature convergence caused by "super" individuals.

■**Nonlinear Ranking** Nonlinear ranking assigns selection probabilities that are based on each individual's rank, but are not proportional to the rank. For example, the selection probabilities might be proportional to the square of the rank:

$$\Pr_{\text{sq\_rank}}(i) = \frac{\alpha + \left[\text{rank}(i)^2/(\mu - 1)^2\right](\beta - \alpha)}{c} \tag{2.60}$$

where $c = (\beta - \alpha)\,\mu\,(2\mu - 1)\,/6\,(\mu - 1) + \mu\alpha$ is a normalization factor. This version has two parameters, α and β, where $0 < \alpha < \beta$, such that the selection probabilities range from α/c to β/c.

Even more aggressive forms of ranking are possible. For example, one could assign selection probabilities based on a geometric distribution:

$$\Pr_{geom\_rank} = \alpha\,(1 - \alpha)^{\mu - 1 - \mathrm{rank}(i)} \tag{2.61}$$

This distribution arises if selection occurs as a result of independent Bernoulli trials over the individuals in rank order, with the probability of selecting the next individual equal to α and was introduced in the GENITOR system [38, 39].

Another variation that provides exponential probabilities based on rank is

$$\Pr_{\mathrm{exp\_rank}}(i) = \frac{1 - e^{-\mathrm{rank}(i)}}{c} \tag{2.62}$$

for a suitable normalization factor c.

Both of the latter methods strongly bias the selection toward the best few individuals in the population, perhaps at the cost of premature convergence.

■ **Tournament Selection** In tournament selection a group of q individuals is randomly chosen from the population. They may be drawn from the population with or without replacement. This group takes part in a tournament; that is, a winning individual is determined depending on its fitness value. The best individual having the highest fitness value is usually chosen deterministically though occasionally a stochastic selection may be made. In both cases only the winner is inserted into the next population and the process is repeated λ times to obtain a new population. Often, tournaments are held between two individuals (binary tournament). However, this can be generalized to an arbitrary group size q called tournament size.

The following description assumes that the individuals are drawn with replacement and the winning individual is deterministically selected.

input: Population $P(t) \in I^{\lambda}$, tournament size $q \in \{1, 2, \cdots, \lambda\}$

```
Output: Population after selection $P(t)'$
tournament $(q, a_1, \cdots, a_\lambda)$:
for $i \leftarrow 1$ to $\lambda$ do
    $a'_i \leftarrow$ best fit individual from $q$ randomly chosen
    individuals from $\{a_1, \cdots, a_\lambda\}$;
od
return $\{a'_1, \cdots, a'_\lambda\}$
```

Tournament selection can be implemented very efficiently and has the time complexity $\mathcal{O}(\lambda)$ as no sorting of the population is required. However, the above algorithm leads to high variance in the expected number of offspring as λ independent trials are carried out.

Tournament selection is translation and scaling invariant. This means that a scaling or translation of the fitness value does not affect the behavior of the selection method. Therefore, scaling techniques as used for proportional selection are not necessary, simplifying the application of the selection method.

Furthermore, tournament selection is well suited for parallel evolutionary algorithms. In most selection schemes global calculations are necessary to compute the reproduction rates of the individuals. For example, in proportional selection the mean of the fitness values in the population is required, and in ranking selection and truncation selection a sorting of the whole population is necessary. However, in tournament selection the tournaments can be performed independently of each other such that only groups of q individuals need to communicate.

2.4.4 Genetic Operations

In this part, other genetic operations such as crossover, elitism and Gray codes, are described.

■Single-point Crossover The single-point crossover has one crossover position. A number of variables of an individual are selected uniformly at random and the variables exchanged between the individuals about this point, then two new offspring are produced. Figure 2.11 illustrates this process.

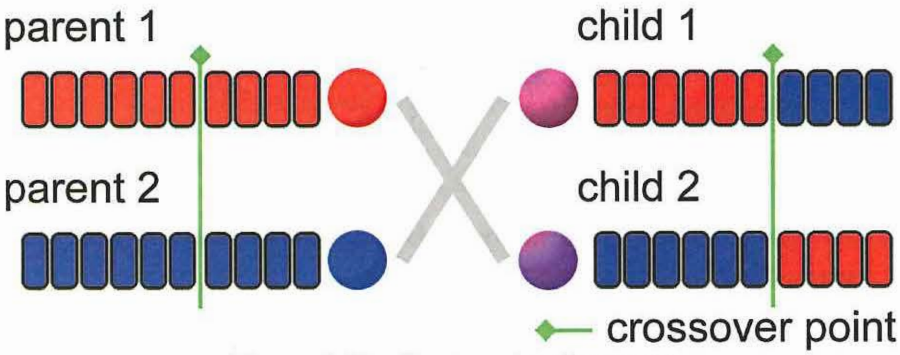


Figure 2.11 Single-point Crossover

■Multi-point Crossover The multi-point crossover has several crossover positions. Number of variables of an individual are chosen at random with no duplicates and sorted in ascending order. Then, the variables between successive crossover points are exchanged between the two parents to produce two new offspring. The section between the first variable and the first crossover point is not exchanged between individuals. Figure 2.12 illustrates this process.

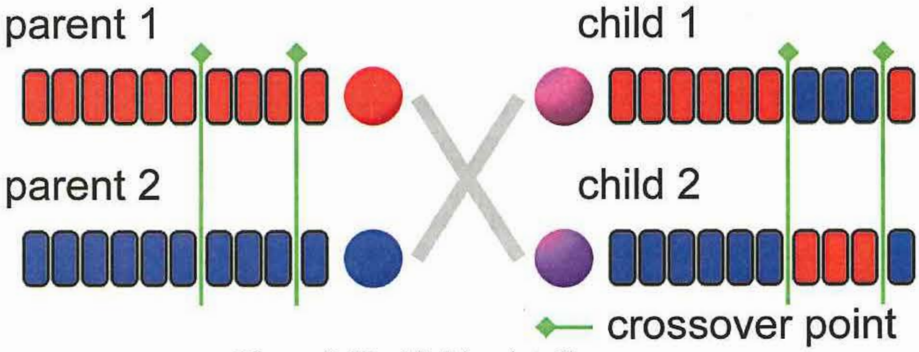


Figure 2.12 Multi-point Crossover

The idea behind multi-point, and indeed many of the variations on the crossover operator, is that parts of the chromosome representation that contribute to the most to the performance of a particular individual may not necessarily be contained in adjacent substrings. Further, the disruptive nature of multi-point crossover appears to encourage the exploration of the search space, rather than favouring the convergence to highly fit individuals early in the search, thus making the search more robust.

■**Uniform Crossover** Single and multi-point crossover define cross points as places between loci where a individual can be split. Uniform crossover generalizes this scheme to make every locus a potential crossover point. A crossover mask, the same length as the individual structure is created at random and the parity of the bits in the mask indicate which parent will supply the offspring with which bits. Figure 2.13 illustrates this process.

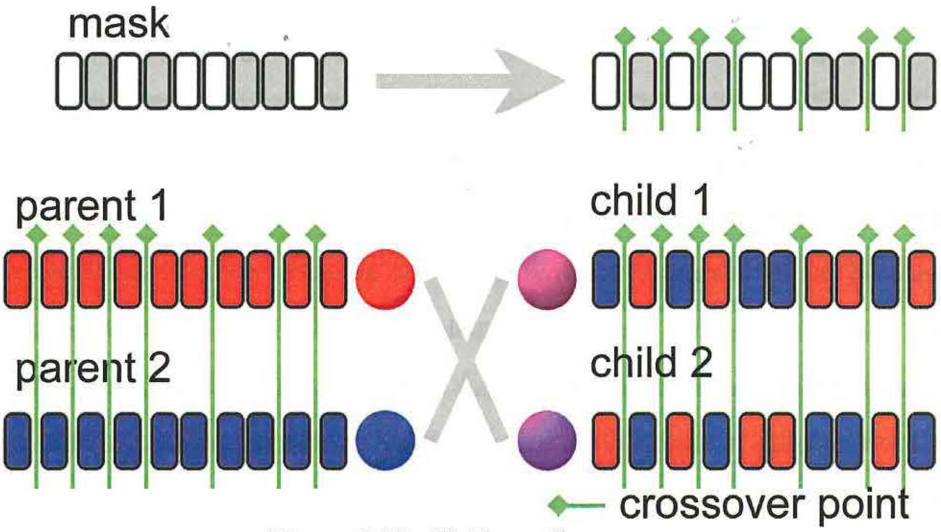


Figure 2.13 Uniform Crossover

Uniform crossover, like multi-point crossover, has been claimed to reduce the bias associated with the length of the binary representation used and the particular coding for a given parameter set. This helps to overcome the bias in single-point crossover towards short substrings without requiring precise understanding of the significance of the individual bits in the individuals representation.

2.4.5 Elitism

Every time a new population is made, the chance occurs that we might lose the string with the best evaluation. This could result in an unstable algorithm and a slower convergence. To overcome this problem, we could simply copy the best member of each generation a number of times into the succeeding generation. This technique is called elitism. This may increase the speed of domination of a population by a super individual, but on balance it appears to improve genetic algorithm performance.

Elitism is not a selection mechanism but rather it is a possible feature of most, if not all, selection mechanisms. Elitism is simply the guarantee that the most fit solution found to date will remain within the population. This is of course very different from saying that the most fit solution will be remembered. At first glance one might wonder why anyone would want a search algorithm that did not have elitism? Why should the active part of the algorithm be allowed to forget the best solution found so far? The perspective of genetic algorithms being heavily inspired by nature allows the simple answer that elitism does not occur in nature. Every living thing dies and a degree of mutation pervades every reproduction event that occurs. Hence no DNA lineage can remain pure. A big debate is whether or not this feature of natural evolution has been selected for or is simply unavoidable.

In the computer realm it is clearly avoidable, so why might we choose it? The main reason is that the elite individual can behave like a nail through the heart of the population trapping it uselessly on an unfit local optimum. With the elite individual stuck at the top, the dynamics of a given genetic algorithm will only allow the rest of the population to wonder away from the elite by a certain distance (on average) if this distance is not enough to allow the discovery of more fertile lands than the current local optimum the population will be stuck there indefinitely. Switching off elitism will allow the population of the same algorithm slightly more scope to wander around. This will give it a slightly better chance of moving off the local optimum (see Figure 2.9).

As always in the theory of search, there is an alternate view on elitism which sees it as a feature which enables the use of higher mutation rates precisely because it will keep hold of the best so far even when the mutation rate is high. One of the problems with higher mutation rates is that the higher they go the harder it is for the genetic algorithm to home in on the exact optimum of a hill on the landscape. Elitism will act like a ratchet mechanism in this quest for the top of the hill, allowing the genetic algorithm to use a mutation rate that does not pussy foot around. This is what will allow a genetic algorithm with elitism to escape local optima that a genetic algorithm without elitism would get stuck on.

2.4.6 Gray codes

Gray codes are named after the Frank Gray who patented their use for shaft encoders in 1953 [40]. Gray codes actually have a longer history, and the inquisitive reader may want to look up the August, 1972, issue of Scientific American, which contains two articles of interest: one on the origin of binary codes, and another by Martin Gardner on some entertaining aspects of Gray codes.

A Gray code represents each number in the sequence of integers $\{0, \cdots, 2^{N-1}\}$ as a binary string of length N in an order such that adjacent integers have Gray code representations that differ in only one bit position. Marching through the integer sequence therefore requires flipping just one bit at a time. Some call this defining property of Gray codes the "adjacency property".

Example ($N = 3$): The binary coding of $\{0...7\}$ is $\{000, 001, 010, 011, 100, 101, 110, 111\}$, while one Gray coding is $\{000, 001, 011, 010, 110, 111, 101, 100\}$. In essence, a Gray code takes a binary sequence and shuffles it to form some new sequence with the adjacency property. There exist, therefore, multiple Gray codings for any given N. The example shown here belongs to a class of Gray codes that goes by the fancy name "binary-reflected Gray codes". These are the most commonly seen Gray codes, and one simple scheme for generating such a Gray code sequence says, "start with all bits zero and successively flip the right-most bit that produces a new string."

Hollstien [41] investigated the use of genetic algorithms for optimizing functions of two variables and claimed that a Gray code representation worked slightly better than the binary representation. He attributed this difference to the adjacency property of Gray codes. Notice in the above example that the step from three to four requires the flipping of all the bits in the binary representation. In general, adjacent integers in the binary representation often lie many bit flips apart. This fact makes it less likely that a mutation operator can effect small changes for a binary-coded individual.

A Gray code representation seems to improve a mutation operator's chances of making incremental improvements, and a close examination suggests why. In a binary-coded string of length N, a single mutation in the most significant bit (MSB) alters the number by 2^{N-1}. In a Gray-coded string, fewer mutations lead to a change this large. The user of Gray codes does, however, pay a price for this feature: those "fewer mutations" lead to much larger changes. In the Gray code illustrated above, for example, a single mutation of the left-most bit changes a zero to a seven and vice-versa, while the largest change a single mutation can make to a corresponding binary-coded individual is always four. One might still view this aspect of Gray codes with some favor: most mutations will make only small changes, while the occasional mutation that effects a truly big change may initiate exploration of an entirely new region in the space of chromosomes.

Chapter 3

Genetic Lips Region Detection and Information Extraction

3.1 Introduction

Lately, mobile devices such as a PDA or a cellular phone have spread throughout. Internet population that used cellular phones increased by 65.1% for the half year, from October 2000 to March 2001. Furthermore, the number of persons with cellular phones that can connect to the Internet increased by 55.1%. On the other hand, Internet population that used a personal computer showed signs of leveling off for four months from December 2000 [42].

There is an issue, when one inputs text to write something like an email using the cellular phone. At the present time, the most common way to input text to cellular phone is by pushing number keys several times. This method is less efficient than using the notebook computer which has a keyboard.

In the field of speech recognition, there is a limit to recognize a paragraph, including a person name and a proper name [43]. To improve the recognition rate, a technique is proposed which uses not only speech data but also other information such as gesture, face image and a motion of lips.

Recently almost all cellular phones contains a small digital camera and can send email with a picture. In addition, the new generation of cellular phone can record a movie data.

There are demands for the input of data by speech when using, the cellular phone with the digital camera and demands for the speech recognition by both speech data and other data. In light of these facts, we propose a speech recognition system which uses not only speech data but also lips images for mobile devices. The main objective of our study is the extraction of a lips region as a preprocessing of that system.

In this chapter, a lips detection and the information extraction method that uses only one template per one user for personal mobile devices is proposed. Our method has invariance to two points that are very important to extract lips region for mobile devices. The first point is lips geometric changes, that is parallel translation, scaling and rotation due to slope of a face or by a non-stable camera. The second point is varying lips shapes, that is, an opened or closed mouth and showing or not showing any teeth, at the moment of speech. In addition, taking into consideration that our method is used with movie data, we hope to extract the lips region with high speed and high accuracy. Therefore this chapter considers that point. We use a single template matching using a GA during the matching process. Simulations in this chapter compare three different fitness functions, and decide the best fitness function for this system.

3.2 Lips Information and Template Shape

This section describes the lips information which must be acquired, and the template shape. At first, the changes of lips are classified according to their causes. Next, the relation between the classification and geometry are described. Taking into account this relation, a new template shape is introduced, which seems appropriate for the lips detection.

3.2.1 Change of Lips and Transformation Group

Changes of lips are divided into two types by the causes. One is changes of the lips region by the independent motion of a camera and a human. In this case, the lips region has changes, such as parallel translation, scaling, and rotation. The other is lips shape deformation by speech.

In this part, the relation between the lips changes and geometry are described with the hierarchy structure of transformation group [34] is illustrated in Figure 2.7.

The lips region change by motion of the camera and the human can be represented by inside group from affine transformation group, because this lips region change has geometric changes, such as parallel translation, scaling, and rotation.

Considering the change of lips shape by speech, in most cases, the mouth is open during speech. Assuming that the lips is a elastic band, the lips during speech can be considered as expand and contract band. In other word, its connectivity is constantly same. Therefore, this change can be represented by topological transformation group. It follows from these that the change of lips can be represented with transformation group.

3.2.2 Shape of Template

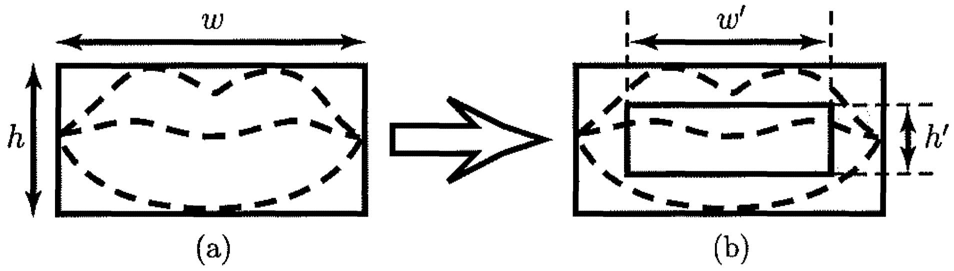


Figure 3.1 Template shape: a) source square ; b) new template shape is called "square annulus."

In general, a typical template shape is a complete square. This shape is suitable for geometric changes, such as parallel translation, scaling, and rotation. However, considering an application of the template matching to the change of lips shapes during speech, the complete square shape of template is unsuitable. This is because, at the moment of speech, the lips region has intense variations such as an opened or closed mouth and showing or not showing any teeth. In other words, changes in oral cavity, such as showing some teeth, and a tongue, cause low extraction accuracy of lips region. For this reason, we use new template shape which considers topological transformation group in the relation between change of lips and transformation group as mentioned above. The new template shape illustrated in Figure 3.1 to cope with the ever-changing lips shape. This template shape is called "square annulus."

In Figure 3.1, w and h are the source square template's width and height, respectively. w' and h' are the new "square annulus" template's width and height, respectively. In simulations, w' and h' are decided experientially. In our preliminary examination, by using"square annulus" extraction accuracy rises up considerably, comparing with the normal template. Furthermore, there are advantages of the "square annulus" that the ignored $w' \times h'$ region reduces the amount of calculation and makes the lips region extraction high speed. In this paper, in order to simplify the system, we fix the rate of interior area, the interior height w' is 80 % of the exterior w, h' is 50 % of h. In our future work, we hope to acquire the relation between the exterior and interior lips contour by non-fixed rate of interior area for many applications.

3.3 Genetic Lips Detection

In this section, the details of our proposed lips detection and lips information acquisition system is described as follow: a structure of chromosome in GA, and a fitness function, a dynamic search domain control (dynamic SD-Control), and a flow chart of the proposed system. In particular, the dynamic SD-Control is a main technique in this proposed system.

3.3.1 Structure of Chromosome

In GA, an individual is a solution candidate to be optimized. The individual has a chromosome to be a source of the solution. In our optimization problem, this solution represents a transformation matrix, which transforms the template on the

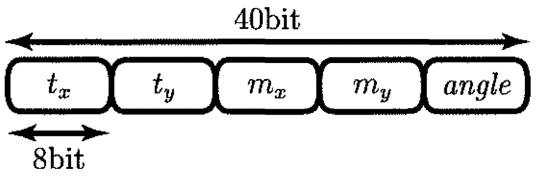


Figure 3.2 A structure of chromosome

target image. In fact, the template is transformed by the matrix in homogeneous coordinates, then, the template matching is performed.

The structure of chromosome is shown in Figure 3.2, where t_x and t_y are coordinates after parallel translation, m_x and m_y are scaling rates, and *angle* is rotation angle of lips region. These parameters are called phenotype, and these are encoded in some bit-string, which is called genotype in GA. We use the binary genotype. For this encode, from the phenotype to genotype, we do not use binary-coding but Gray-coding, since Gray-coding is generally superior [44].

The proposed system can detect the lips and also acquire the lips information directly by this optimized chromosome. The template's width and height should be changed separately, because of shape deformation of lips by speech is not only similarity change. Consequently, we use two-dimensional scaling by m_x and m_y.

3.3.2 Transformation by Chromosome

In this part, geometric transformation is explained. This transformation solves geometric changes of lips, such as parallel translation, scaling, and three-dimensional rotation, as mentioned in Section 3.3.1. All individuals in GA have a unique chromosome. In this proposed system, coordinate transformation of the template is performed on the target image by this chromosome.

The transformation is represented by a simple combination of matrix multiplications, because homogeneous coordinate [45] are used. The matrix is obtained from the chromosome of the genetic algorithm described in Section 3.3.1.

Let A be a point on the template image, and A^* is a point, which corresponds to a transformed point A on the target image. A and A^* are represented by homogeneous coordinates as follow:

$$A = [X, Y, 1], \tag{3.1}$$

$$A^* = [X^*, Y^*, 1]. \tag{3.2}$$

The template image is transformed for geometric changes. As below, some matrices are specified by chromosome of the genetic algorithm (refer to Section 3.3.1). M

represents the scaling, R is the rotation on y-axis, and T is the parallel translation.

$$M = \begin{bmatrix} m_x & 0 & 0 \\ 0 & m_y & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{3.3}$$

$$R = \begin{bmatrix} \cos(angle) & \sin(angle) & 0 \\ -\sin(angle) & \cos(angle) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{3.4}$$

$$T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix}. \tag{3.5}$$

The point A^* is given by the following simple equation.

$$A^* = AMRPT. \tag{3.6}$$

3.3.3 Pixel Difference

The template image is transformed on the target image by the chromosome as mentioned above. After the transformation the pixel difference between the template and target image is calculated for every point of template image as follow:

$$D_{ij} = \begin{cases} |p^*_{ij} - p_{ij}| & (p^*_{ij} \in \text{target image}) \\ P_{\max} & (p^*_{ij} \notin \text{target image}) \end{cases}, \tag{3.7}$$

where $P_{\max}$ is the maximum value of pixel, p is a pixel value of a point P on coordinate (i, j) in the template image, p^* is a pixel value of a point P^* on coordinate (i, j) in the target image. In other words, P is a point on the template image, and P^* is the point on the target image that corresponds to a point transformed from the point P. D_{ij} is a value of the pixel difference between p and p^*, however, in case that a point P^* is out of region in the template image, D_{ij} is the worst $P_{\max}$.

In our system, the pixel value is the x component value as described in Section 2.2.

3.3.4 Objective Function

An object in GA exploration of this system is to locate lips (the template) on the target image. Therefore, An objective function is a summation of the pixel difference D_{ij} between the template and the target image. This objective function is a minimization problem, and defined by equation (3.8).

$$O = \sum_{j=1}^{h} \sum_{i=1}^{w} D_{ij}, \tag{3.8}$$

where O is the objective value, which is a summation of pixel differences D_{ij}.

In the next section, some different variations on the fitness function are shown, which change the minimization problem to a maximization problem.

3.3.5 Flow Chart

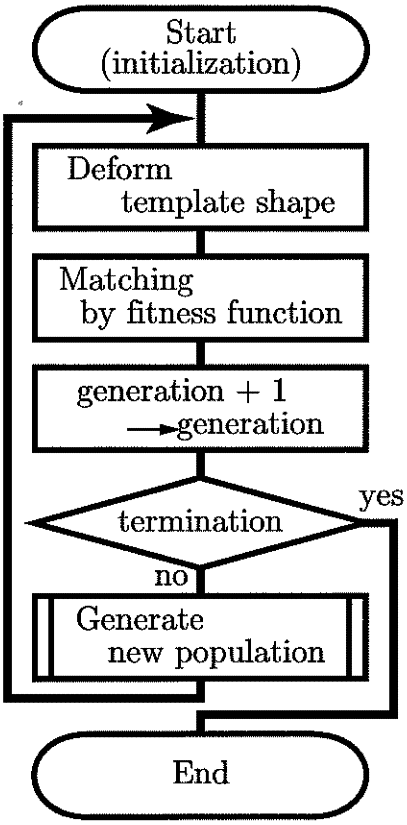


Figure 3.3 Flow charts.

Flow charts of our system are illustrated in Figure 3.3.

At first, an initial population is generated, then GA is started. In GA processing, the template shape is deformed to an unique "square annuls" from a normal square, as explained in Section 3.2.2. Then, matching process is executed between the template and a target image using a fitness function by the objective function. This fitness function is described in Section 3.4. The generation is increased, until a termination condition of GA is satisfied. In this paper, GA is terminated by the number of generations. If the termination criterion is not satisfied, a new population of the next generation is generated according to the fitness of each individual. After GA process is completed, the result is obtained as numerical data. This numerical data represents the lips information which can be used for the applications as exemplified in Section 3.1.

3.4 Fitness Function

As mentioned above, the objective function is regarded as a minimization problem. Typically, the fitness function is a maximization problem in GA. In this section, two different static fitness functions and one dynamic fitness function are described. Three fitness functions are evaluated by comparison of these fitness functions in Section 3.5.

3.4.1 Static Fitness Function 1

At first valuation of static fitness function is the most basic, which uses normalization and penalization.

After the objective function O is calculated, the fitness function is calculated as follows:

$$\mathit{fitness} = 1.0 - \frac{O}{(w \times h)\,(P_{\max})}, \tag{3.9}$$

where $fitness$ is the fitness value, the template size is w and h, $A_{\max}$ is the maximum value of pixel, and O is a value of the objective function (refer to Section 3.3.4). In this fitness function, the objective function is normalized. Therefore, the $fitness$ allows us to a good exploration that the fitness approaches to 1.

After many experiments we found out that the exploration of codomain near $fitness = 1$ is important. Moreover, the search efficiency is not good and the result is unstable by using the above fitness function as mentioned in [46], because, there is less reflection of the fitness change to the result. The reason why such imbalance occurred is that the small pixel difference between a and a^* is ignored by the normalization, and the individual difference is very small. Thus, the small pixel difference should not be ignored and the individual difference should be emphasized by introducing penalty in the fitness function. Therefore we propose the following penalized method.

Before the objective fitness function (refer to Section 3.3.4) is calculated, the pixel difference D_{ij} is checked whether penalize or not. If a pixel difference satisfies equation (3.10), D_{ij} is the worst maximum value $A_{\max}$.

if

$$D_{ij} \geq TH, \tag{3.10}$$

then

$$D_{ij} = P_{\max}, \tag{3.11}$$

where TH is the threshold value to penalize. This value is calculated by the following equation:

$$TH = \left(1 - \frac{S}{100}\right) \times P_{\max}, \tag{3.12}$$

where S is the similarity per pixel and decided experientially and the unit is [%]. The more S increases, the stricter a condition becomes. In simulations, S is 90. Because, a probability to 100[%] match is very low. After this penalization, equation (3.9) is calculated.

3.4.2 Static Fitness Function 2

The Second static fitness function is the simplest fitness function. In the static fitness function 1, the fitness function is normalized in range $[0, 1]$, because of reducing a dispersion of pixel differences whose pixel values are real values, x component of Yxy colour space. Moreover, this normalization changes genetic algorithm process into a simple maximization problem. However, after many experiments we found out that the exploration in near $fitness = 1$ is important. In other words, the matching result is not affected very well by the fitness function. In static fitness function 2, the objective function is changed to minimization problem without normalization as follow:

$$fitness = (w \times h)\,(P_{\max}) - O, \tag{3.13}$$

where $w \times h$ is the template size, $P_{\max}$ is a maximum value of pixel, O is value of the objective function. O_{ij} allows us to achieve a good exploration where the value of the objective function approaches 0. In other words, *fitness* allows us to do a good exploration where the fitness value becomes large.

3.4.3 Dynamic Fitness Function

It is important to control the selection pressure, because the selection pressure and population diversity are inversely related [47]. In other words, as the selection pressure is increased, the population diversity is lost, moreover this can cause a premature convergence. Against that, the lack of selection pressure can cause a evolutionary retardation.

A major approach to avoid this problem is "ranking" [48]. However this search speed is slow except appropriate cases [47], hence we used other technique in a fitness function.

The dynamic fitness function is shown in equation (3.14),

$$\mathit{fitness} = \max\{W_t, W_{t-1}, \ldots, W_{t-n}\} - O, \tag{3.14}$$

where *fitness* is a fitness value, W is the worst objective value, t is a current index of generation. This fitness function is a difference between the objective value and the worst objective value for last $n+1$ generations. This technique is called "scaling window" [47, 49], which is used for controlling selection pressure of GA.

To use this scaling window, a scaling window size n must be decided, then we use $n = 5$ in the experiment which decided experientially. Generally, if the size is too large, GA exploration depends on a longstanding worst individual, hence GA search becomes slow. Against that, if the size is too small, GA exploration is sensitive to a noise and incidental good individual, hence GA exploration becomes easy to trap into a local optimum. We have not decided the size conclusively, and there has been no report about the decision, as far as we know.

3.5 Computer Simulation Results and Considerations

3.5.1 Input Images and Output image

The template images are illustrated in Figure 3.4. Template image size of subject 1 and subject 2 is 18 × 8 [pixels], and subject 3 is 20 × 9 [pixels].

Figure 3.5 below shows examples (pronounce the vowel /e/) of target images. The images captured using a video camera include a face and background while each of three objects pronounces the vowels. The target images are then cut from the video streams. In consideration of the use by mobile devices, the target images have geometric changes based on the template image. The geometric changes mean parallel translation, scaling, rotation. The size of all target images is 240 × 180 [pixels]. These input images are used in the following experiments.

Figure 3.6 shows examples of lips region extraction results in case of successful cases. Almost the same results are obtained by any fitness function mentioned above. The rectangle region is the extracted lips region. The shape deformations of lips by speech are extracted exactly as shown in Figure 3.6.

Table 3.1 shows the true solution obtained manually for /e/ of subject 2 in Figure 3.5 and the solution obtained by the proposed method. It is found that these both solutions are similar.

Table. 3.1 Example of solution of result (subject 2 /e/)

	coordinate		scaling [rate]		rotation
	x	y	x	y	[deg]
true solution	82	128	1.39	2.00	19.20
experimental result	81	131	1.68	2.51	17.33

3.5.2 GA Settings

We choose uniform crossover, because of its many advantages [50]. The parameters of genetic algorithms are: population size is 70, probability of crossover is 0.7, and probability of mutation is 0.05.

If the same elite fitness value continues for over some generations, the solution

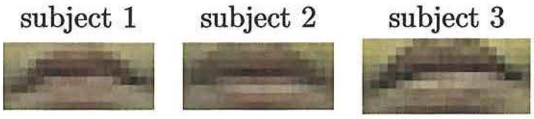


Figure 3.4 Template images

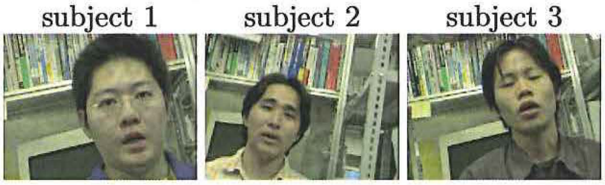


Figure 3.5 Target images

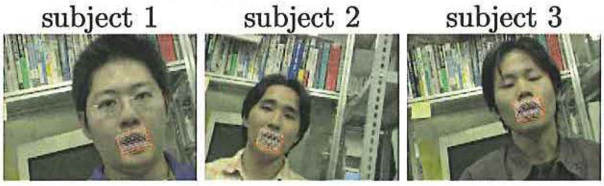


Figure 3.6 Resulting images

is regarded as having converged and the GA is terminated. The number of these generations is a termination criterion. The more this value becomes large, the more the termination criterion becomes fair. The reason is that if this value is small, the GA cannot evaluate for a long time.

3.5.3 Static Fitness Function 1 VS. Static Fitness Function 2

In this part, Static Fitness Function 1 is compared with Static Fitness Function 2. The effectiveness of our method is demonstrated using 20 times simulations for each person (total is 60 times simulations per one vowel) being tested as shown in Tables 3.2, 3.3 and 3.4. In case of Table 3.2, the termination criterion value is 50. This means that the termination criterion in Table 3.2 is fair. Against that, in case of Tables 3.3 and 3.4, their termination criterion value is 30. This means that the termination criterions in Tables 3.3 and 3.4 are tough.

In Table 3.3, processing time is faster than others. However, the extraction accuracy is not good.

Comparing Table 3.2 with Table 3.3, they indicate that the Static Fitness Function 1 obtains high extraction accuracies on the fair criterion, however on the tough criterion, it obtains low extraction accuracies. Against that, the Static Fitness Function 2 works on the tough criterion in Table 3.4.

Now, see Figures 3.7(a) and 3.7(b). They illustrate a typical transition of the elite fitness and objective value with the Static Fitness Function 1 and the Static Fitness Function 2. Good results such as Figure 3.6 are obtained by these simulations. In Figure 3.7(a), the elite fitness changes with very small value and narrow range as opposed to Figure 3.7(b). This indicated that Figure 3.7(b) performed more efficient exploration. The reason is that the Static Fitness Function 1 normalizes the objective value and reduces individual differences, while the Static Fitness Function 2 does not normalize.

3.5.4 Effectiveness of Dynamic Fitness Function

The effectiveness of our method is demonstrated using 20 times simulations for each person (total is 60 times simulations per one vowel) being tested as shown in

Table. 3.2 Results of simulation (Static Fitness Function 1, fair criterion)

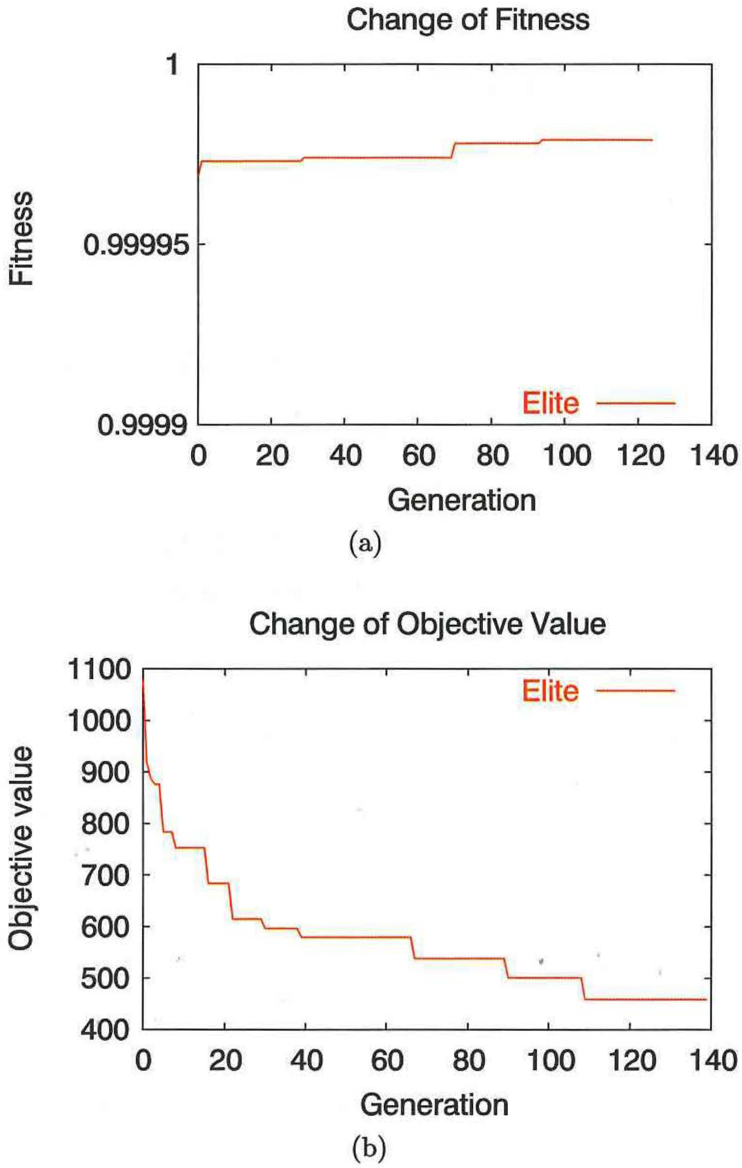
	/a/	/i/	/u/	/e/	/o/	total
extraction accuracy [%]	98.33	96.67	91.67	93.33	91.67	94.33
processing time [msec]	0.323	0.318	0.332	0.299	0.329	0.320
generation	140.67	139.93	147.13	130.77	144.20	140.54

Table. 3.3 Results of simulation (Static Fitness Function 1, tough criterion)

	/a/	/i/	/u/	/e/	/o/	total
extraction accuracy [%]	25.00	21.67	26.67	21.67	26.67	24.33
processing time [msec]	0.088	0.089	0.110	0.105	0.101	0.099
generation	61.80	62.69	76.06	75.69	72.81	69.81

Table. 3.4 Results of simulation (Static Fitness Function 2, tough criterion)

	/a/	/i/	/u/	/e/	/o/	total
extraction accuracy [%]	96.67	91.67	96.67	85.00	91.67	92.33
processing time [msec]	0.206	0.187	0.177	0.206	0.211	0.198
generation	87.81	78.16	74.07	86.20	88.53	82.95



(a)

(b)

Figure 3.7 Transition of GA evolution: a) elite fitness (Static Fitness Function 1) ; b) elite objective value (Static Fitness Function 2)

Tables 3.2, 3.3, 3.4 and 3.5.

In case of Table 3.2, the termination criterion is 50 generations. Against that, in case of Tables 3.3, 3.4 and 3.5, their termination criterion value is 25 generations. In other words, Table 3.2 criterion is fairer than that in Tables 3.3, 3.4 and 3.5.

In Table 3.3, a processing time is faster than others. However, the extraction accuracy is not good. Comparing Table 3.5 with Table 3.4, they indicate that the

Dynamic Fitness Function obtain higher extraction accuracy than the Static Fitness Function 2, and the processing time is almost same in both methods.

Figures 3.8 illustrates a typical transition of the objective value with the Dynamic Fitness Function. Comparing Figure 3.8 with Figure 3.7(b), as you can see, the Dynamic Fitness Function converge faster than the Static Fitness Function 2. This indicated that Figure 3.8 performed more efficient exploration. The reason is that the Dynamic Fitness Function method controls the selection pressure dynamically by equation (3.14).

Table. 3.5 Results of simulation (Dynamic Fitness Function, tough criterion)

	/a/	/i/	/u/	/e/	/o/	total
extraction accuracy [%]	95.00	93.33	95.00	91.67	98.33	94.67
processing time [sec]	0.198	0.203	0.199	0.197	0.199	0.199
gene	78.48	81.15	80.06	77.25	76.50	78.69

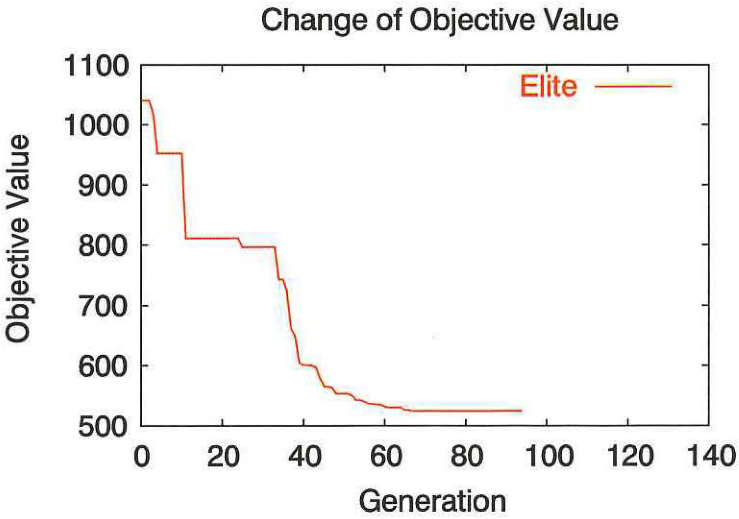


Figure 3.8 Transition of GA evolution: elite objective value (Dynamic Fitness Function)

3.6 Conclusion

In this chapter, a lips extraction method that uses only one template per one user for mobile devices is proposed. Our method has invariance to two points that are very important to detect lips region for mobile devices. The first point is lips geometric changes, that is parallel translation, scaling and rotation due to slope of a face or by a non-stable camera. The second point is varying lips shapes, that is, an opened or closed mouth and showing or not showing any teeth, at the moment of speech. In addition, taking into consideration that our method is used with movie data, we hope to detect the lips region with high speed and high accuracy. Therefore this paper considers that point.

We use a single template matching using a GA during the matching process. Three variations of fitness function are proposed and compared in simulations. The results shows that the best fitness function for this system is the Dynamic Fitness Function. The following chapters use this Dynamic Fitness Function.

This system has some problems, such that the processing time is long. Moreover, this system is unsuitable for practical use, because this system deals with two-dimensional changes only. In order to support the real-world, we must develop the system to three-dimensional space. In the next chapter, this development is described.

Chapter 4

Invariant Detection for Horizontal Direction

4.1 Introduction

In ubiquitous and pervasive computing environments, such as intelligent buildings, mobile devices, a mobile robot, a mobile phone, one of the most useful interface is a speech recognition. This speech recognition must overcome the background noise and exceed the limit of recognition accuracy. To solve these problems, a number of researches had presented audio-visual speech recognition [5, 6, 7, 8, 9]. It is well known that audio-visual integration is very useful for human speech perception [1, 3, 2].

The purpose of our study is lips region extraction as a new visual front end of audio-visual speech recognition on the mobile devices. Several related work [5, 6, 8, 9, 7] has carried out and demonstrated that the audio-visual speech recognition is effective. However, in most cases, the condition are restricted, which the camera and the human is stationary and an input image is only a face region. On the mobile devices, the camera and the human head run around separately. Therefore, this lips region extraction must be robust for some considerable geometric changes in three-dimensional space, as well as varying lips shapes by speech.

In the previous Chapter 3, introduced a system which deals with rotation in two-dimensional space—lips region rotates in parallel to the camera. In this chapter, development of this system toward a three-dimensional space is carried out. A proposed system in this chapter is based on Chapter 3. The shape of template is the same as "square annulus" with Chapter 3. In experiments, we try to extract lips region from a face image with background, varying shape by speech, geometric changes, and face horizontal direction change.

4.2 Input images

The template images are illustrated in Figure 4.1. Template image size of subject 1 is 18 × 8 pixels, subject 2 is 21 × 11 pixels, and subject 3 is 23 × 9 pixels.

Figure 4.2 below shows examples (pronounce the vowel /a/) of target images. The images captured using a video camera include a face and background while each of three subjects pronounces the vowels. Target images are then cut from the video streams. In consideration of the use by personal mobile devices, the lips region on the target images has some considerable geometric changes based on the template image. These geometric changes in this paper mean parallel translation, scaling, and three-dimensional rotation by change of face horizontal direction. Parameters which represent these geometric changes can be regarded as the solutions of GA (refer to Section 4.4.1). The size of all target images is 240 × 180 pixels.

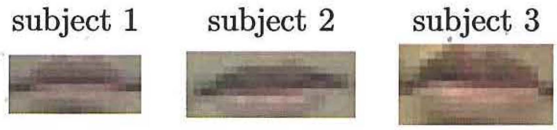


Figure 4.1 Template images

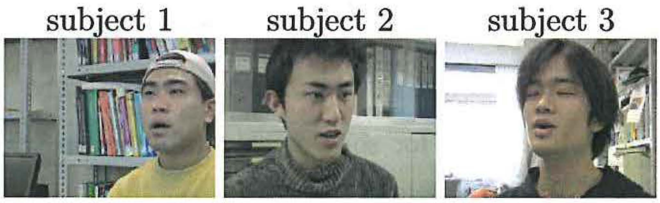


Figure 4.2 Target images

4.3 Lips Information and Template Shape

4.3.1 Change of Lips and Transformation Group

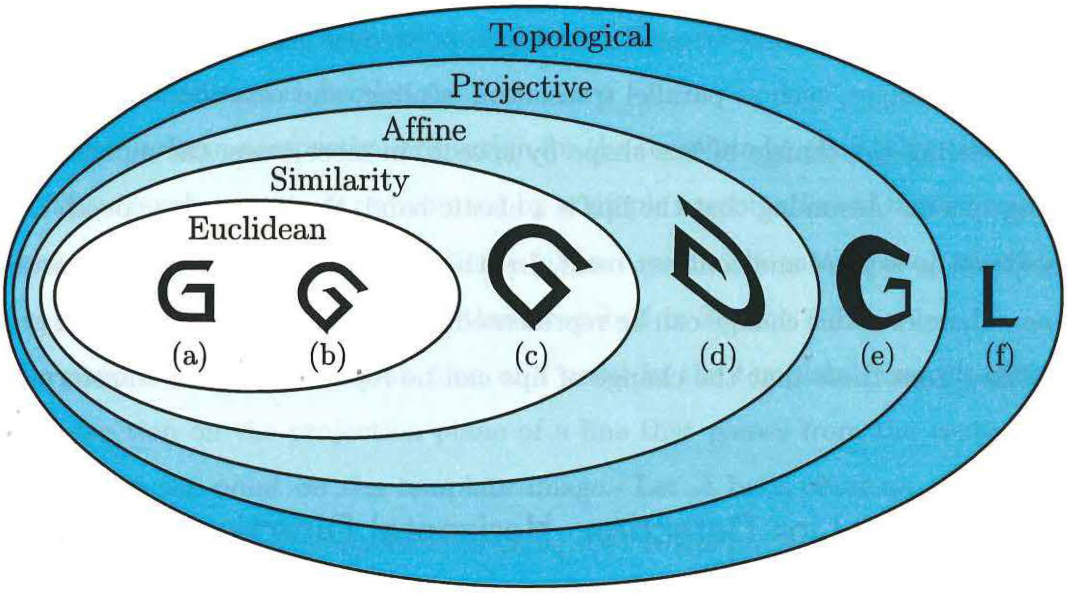


Figure 4.3 Hierarchy structure of transformation group: a) base ; b) rotation and translation (Euclidean transformation group) ; c) scaling (similarity) ; d) reflection and shearing (affine) ; e) projection ; f) connectivity (topological).

Changes of lips are divided into two types by the causes. One is changes of the lips region by the independent motion of a camera and a human. In this case, the lips region has changes, such as parallel translation, scaling, and rotation. The other is lips shape deformation by speech.

Before description of the relation between the lips changes and geometry, we would like to make certain of geometric transformation group [34]. Hierarchy structure of transformation group [34] is illustrated in Figure 4.3. This hierarchy structure is not same with Figure 2.7. In this Figure 4.3, "G" and "L" are considered as graphic symbols. The base shape is Figure 4.3(a). The most inside group is Euclidean transformation. Figure 4.3(b) is a result of rotation. The second is similarity transformation group. Figure 4.3(c) is a result of scaling. The third is affine transformation group. Figure 4.3(d) is a result of shear and reflection. The fourth is projective transformation group. Figure 4.3(e) is a result of projection. The most outside is topological transformation group. A connectivity of all "G" and "L" are same, therefore these relation is homeomorphic.

The lips region change by motion of the camera and the human can be represented by inside group from affine transformation group, because this lips region change has geometric changes, such as parallel translation, scaling, and rotation.

Considering the change of lips shape by speech, in most cases, the mouth is open during speech. Assuming that the lips is a elastic band, the lips during speech can be considered as expand and contract band. In other word, its connectivity is constantly same. Therefore, this change can be represented by topological transformation group. It follows from these that the change of lips can be represented with transformation group.

4.4 Genetic Lips Detection: Horizontal Direction Invariance

4.4.1 Structure of Chromosome

A chromosome is an optimised solution. In other words, chromosomes are parameters which represent coordinates, scaling and rotation of an object to be explored on the target image. Figure 4.4 shows the structure of a chromosome. In Figure 4.4, t_x and t_y are coordinates after parallel translation, m_x and m_y are scaling rates, and *angle* is rotation angle on y-axis (horizontal direction) of lips shape. Each gene length is 8 bits and therefore, the total chromosome length is 40 bits.

The template's width and height should be changed separately, because of varying shape of lips by speech which is not only similarity change. Thus, we use 2-dimensional

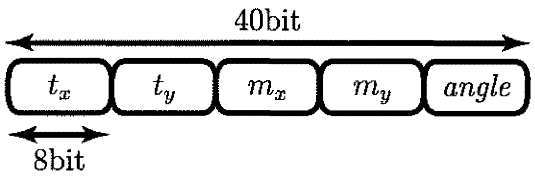


Figure 4.4 A structure of chromosome

scaling by m_x and m_y.

4.4.2 Projective Transformation

In this part, geometric transformation is explained. This transformation solves geometric changes of lips, such as parallel translation, scaling, and three-dimensional rotation, as mentioned in Section 4.4.1.

We must perform perspective projection for the template by projective transformation. This transformation is represented by a simple combination of matrix multiplications, because homogeneous coordinate [45] are used. The matrix is obtained from the chromosome of the genetic algorithm described in Section 4.4.1.

The position on the projection plan of a point in the template image is given by the intersection on the projection plane of a line that passes from the center of the projection to the point on the template image. Let A be a point on the template image, and A^* is called perspective drawing. A^* is the point that corresponds to a transformed point A on the target image. A and A^* are represented by homogeneous coordinates as follows:

$$A = [X, Y, Z, 1], \tag{4.1}$$

$$A^* = [X^*, Y^*, Z^*, 1]. \tag{4.2}$$

In this chapter, the center of the projection is a point $C\left(x_c, y_c, z_c\right)$, the projection plane is xy-plane ($z = 0$), and a center of the template image locates at the origin. Therefore, the horizontal rotation of face direction means a rotation on y-axis. The line through the point $A\left(X, Y, Z\right)$ on the template and the point C is shown by a parameter t as follows:

$$\begin{cases} x = x_c + \left(X - x_c\right) t \\ y = y_c + \left(Y - y_c\right) t \\ z = z_c + \left(Z - z_c\right) t \end{cases}. \tag{4.3}$$

The intersection on the projection plane of a line that passes from the center of the projection to the point on the template image is represented as follows, by $t =$

$-z_c/\left(Z-z_c\right)$, because $z=0$.

$$\begin{cases} X^* = x_c - \dfrac{z_c\left(X-x_c\right)}{Z-z_c} \\ Y^* = y_c - \dfrac{z_c\left(Y-y_c\right)}{Z-z_c} \\ Z^* = 0 \end{cases} . \tag{4.4}$$

Therefore, A^* is shown by the homogeneous coordinates as follows. P is a matrix of the projective transformation. Relation $\sim$ represents the equivalence relation [45].

$$\begin{aligned} A^* &= \left[\frac{Xz_c - Zx_c}{z_c - Z}, \frac{Yz_c - Zy_c}{z_c - Z}, 0, 1\right] && (4.5) \\ &\sim \left[Xz_c - Zx_c, Yz_c - Zy_c, 0, z_c - Z\right] && (4.6) \\ &= A\begin{bmatrix} z_c & 0 & 0 & 0 \\ 0 & z_c & 0 & 0 \\ -x_c & -y_c & 0 & -1 \\ 0 & 0 & 0 & z_c \end{bmatrix} && (4.7) \\ &= AP. && (4.8) \end{aligned}$$

Before the projective transformation, the template image must be transformed for other geometric changes. As below, some matrices are specified by chromosome of the genetic algorithm (refer to Section 4.4.1). M represents the scaling, R is the rotation on y-axis, and T is the parallel translation.

$$M = \begin{bmatrix} m_x & 0 & 0 & 0 \\ 0 & m_y & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4.9}$$

$$R = \begin{bmatrix} \cos\left(angle\right) & 0 & -\sin\left(angle\right) & 0 \\ 0 & 1 & 0 & 0 \\ \sin\left(angle\right) & 0 & \cos\left(angle\right) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4.10}$$

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ t_x & t_y & 0 & 1 \end{bmatrix}. \tag{4.11}$$

As mentioned above, the point A^* is given by the following simple equation.

$$A^* = AMRPT. \tag{4.12}$$

4.4.3 Fitness Function

In this proposed system, the Dynamic Fitness Function, which is described in Section 3.4, is used as fitness function. This fitness function has an objective function and a dynamic fitness function. The objective function is regarded as a minimisation problem and the dynamic fitness function is regarded as a maximisation problem.

4.5 Computer Simulation Results and Considerations

4.5.1 Configurations of System

The parameters of the genetic algorithm are: population size is 100, probability of crossover is 0.7, and probability of mutation is 0.05. Parameters of the ignored region of the square annulus (refer to Figure 3.1 in Section 3.2.2) are decided by trial and error every subject, because the template and lips size of each subject are not the same as shown in Figure 4.1. In subject 1, $w'/w = 0.8$ and $h'/h = 0.3$. In subject 2, $w'/w = 0.7$ and $h'/h = 0.2$. In subject 3, $w'/w = 0.6$ and $h'/h = 0.2$. We use $n = 2$ in equation (3.14) of the Dynamic Fitness Function. If the same fitness value continues for some generations, the solution is regarded as having converged and extraction is terminated. We use this number of generations as termination criterion of the genetic algorithm. The more this value becomes large, the more the termination criterion becomes fair. The machine speck which we use for simulation is Pentium4: 2GHz.

4.5.2 Result image

Figure 4.5 shows examples of results obtained from the computer simulation. The rectangle region is the extracted lips region. The shape deformations of lips by speech are extracted exactly as shown in Figure 4.5.

4.5.3 Evaluation

Table 4.1 shows the true solution obtained manually and the experimental result obtained by the proposed method for /a/ of subject 1 in Figure 4.2. This result shows a exact lips region extraction and both these solutions are similar.

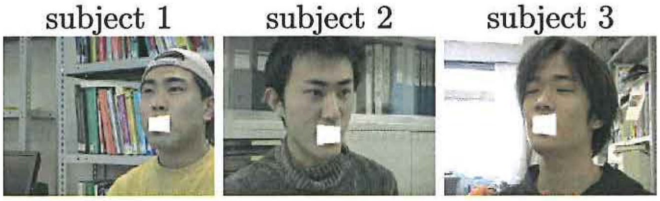


Figure 4.5 Result images

Table. 4.1 Example of solution of result (subject 1 /a/)

	coordinate		scaling rate		rotation
	x	y	x	y	[deg]
true solution	175	100	1.48	2.17	-30.0
experimental result	177	100	1.53	2.26	-30.33

The solution obtained by a manual operation, is called a true solution. Our method results are judged to be good or not good by comparison with the true solution. The comparison is performed by the following equations.

$$\begin{cases} C - 3 \leq c \leq C + 3 \\ M \leq m \leq 1.3 \times M \\ ANGLE - 5^\circ \leq angle \leq ANGLE + 5^\circ \end{cases} , \tag{4.13}$$

where capital letters are the solution obtained manually, and small letters are a solution by the proposed method. c represents the x or y-coordinate, m is a scaling rate and *angle* is a rotation angle. If a result satisfies these conditions, a good result for the speech recognition is obtained.

The effectiveness of our method is demonstrated using 20 times simulations per one vowel for every subject, therefore, the total is 300 times simulations being tested as shown in Tables 4.2, and 4.3.

Table. 4.2 Results of simulation (fair criterion)

	/a/	/i/	/u/	/e/	/o/	total
extraction accuracy [%]	98.3	98.3	95.0	96.7	98.3	97.3
processing time [sec]	0.75	0.71	0.78	0.64	0.75	0.73
generation	320.2	302.5	346.2	281.0	324.2	314.8

Table. 4.3 Results of simulation (tough criterion)

	/a/	/i/	/u/	/e/	/o/	total
extraction accuracy [%]	71.7	83.3	71.7	70.0	80.0	75.3
processing time [sec]	0.20	0.21	0.19	0.16	0.17	0.19
generation	81.0	88.8	81.2	68.3	71.3	78.1

The configuration of the simulations in Tables 4.2 and 4.3 are described in Section 4.5.1. In these tables, results are shown by vowel to examine the relationship between the ignored region of the square annulus (refer to Section 3.2.2) and the lips shape by pronunciation.

The difference between Tables 4.2 and 4.3 is the termination criterion of the genetic algorithm. As mentioned above (refer to Section 4.5.1), the termination criterion is the number of generations, until which the same fitness value continues. If the termination criterion value is too small, exploration of the genetic algorithm will be defective. Against that, if this value is too large, the exploration will be waste. Table 4.2 is obtained by the fair termination criterion value which is 100. Table 4.3 is obtained by the tough termination criterion value which is 25. In other words, Table 4.2 criterion is fairer than that in Table 4.3.

In Table 4.2, we obtain a good result which total extraction accuracy is 97.3%. However, about vowel /u/ the extraction accuracy is lower than other vowels, and the processing time is longer than the others. This is due to the relationship between the inside ignored region of the square annulus and the lips shape of pronunciation. This ignored region is decided by two parameters as described in Section 4.5.1, whose width is longer than height. Against that, the interior lips contour of pronunciation /u/ is not in consistency with this completely.

Comparing Table 4.2 with Table 4.3, all extraction accuracy of Table 4.2 is better than that of Table 4.3, against that all processing time of Table 4.2 is fewer than that of Table 4.3. These indicate the tough termination criterion has a possibility to reduce the processing time, however, probably genetic algorithm exploration is defective.

4.6 Conclusion

In this chapter, a simple method for lips region extraction is proposed, which has robustness for varying shape and horizontal direction (rotation on y-axis) three-dimensional geometric changes by using only one image template. From simulation results in Section 4.5, it is shown that the proposed method has invariance to the varying lips shape and the oral cavity by speech, some geometric changes, such as parallel translation, scaling, and three-dimensional rotation. Furthermore, out of consideration of the processing time, this results indicate that high extraction accuracy can be obtained in the extraction processing of all the vowels. In this chapter, as the development of the system as described in Chapter 3, the object detection system using projective geometry which has invariance for the horizontal direction is proposed. This system can be applied to three-dimensional geometric change—rotation on y-axis. However, in a real world, rotation on all axes must be considered. In the next chapter, this system is improved to deal with every geometric change in three-dimensional space.

Chapter 5

Information Extraction of Lips Region in 3D space

5.1 Introduction

In Chapter 3, the genetic object detection and numerical parameters extraction system is described. The object can cause complex changes, such as shape deformations and two-dimensional geometric changes. This system is applicable only in two-dimensional space—rotation on z-axis. As the development of this system, object detection using projective geometry which has invariance for the horizontal direction is proposed. This system can be applied to three-dimensional geometric change—rotation on y-axis. The system should be applied to rotation on x, y, z-axis in more natural scene. In this chapter, a genetic lips extraction with rotational invariance on all axes is proposed. This system can deal with every geometric change, which is derived from a relation between a camera and an object. This proposed system is based on Chapters 3 and 4, and projective geometry [45] are used to take into account three-dimensional change of lips.

For mobile devices, some problems must be solved, which are listed below.

1. Three-dimensional geometric change
2. Changes of whole scene by free camera motion
3. Acquisition of numerical information of lips information
4. Real-time processing with keeping high accuracy

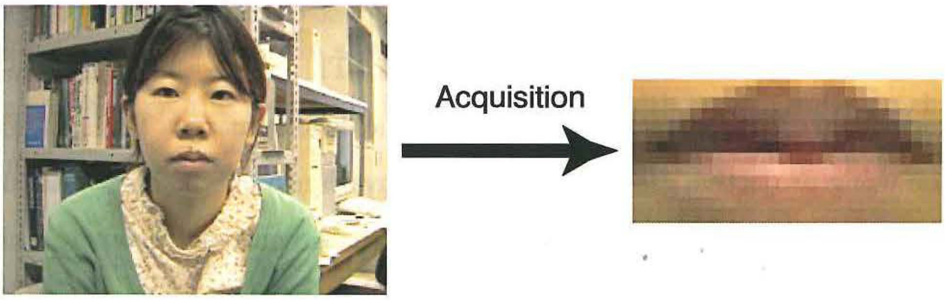
The lips geometric change information is used for correction of lips region. This correction is important, because a lips region normalized by this correction will make speech recognition easy.

The chapter is organized as follows: input images are shown in Section 5.2. Our approach is explained in the next Section 5.3. Section 5.4 shows an evaluation of the approach and discussion of simulation results. Finally, Section 5.5 gives a conclusion.

5.2 Input images

5.2.1 Input Images

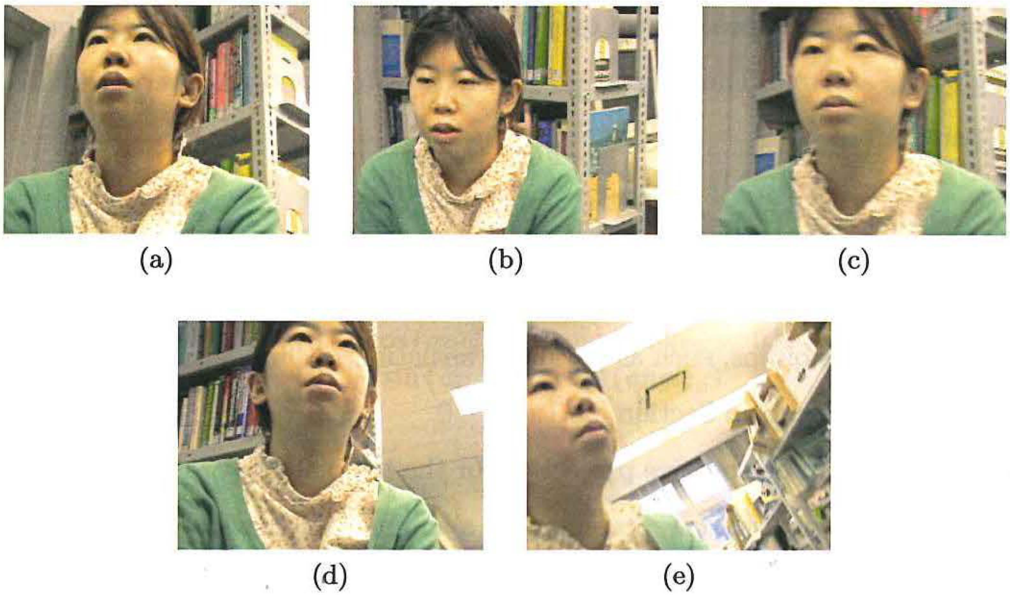
At first, we input only one closed mouth template and a target image in that order. The template image is illustrated in Figure 5.1(b), which is acquired from Figure 5.1(a). The template image size is 28 × 13 pixels.



(a) (b)

Figure 5.1 Template acquisition: a) source of template image (240 × 180 pixels) ; b) template image (28 × 13 pixels).

The target images are presented in Figure 5.2. These are trimmed from a video sequence, which is taken by a digital video camera. Some red color objects are included in the background. The subject makes the Japanese vowel sound iteratively. Moreover, on the assumption that the camera moves and joggles by free hand, the shaking of scene is caused artificially by hand. Therefore, the lips region has some geometric changes. All target image size are 240 × 180 pixels.


(a) (b) (c)

(d) (e)

Figure 5.2 Target images: Subject makes the Japanese vowel sound: starting from a) to e) ; /a/, /i/, /u/, /e/, and /o/.

5.3 Three-dimensional Genetic Lips Detection

5.3.1 Structure of Chromosome

A chromosome of GA is a solution candidate of the problem which must be solved. In other words, chromosomes are parameters which represent coordinates, scaling and rotation of an object to be explored on the target image. Figure 5.3 shows the structure of a chromosome. t_x and t_y are coordinates after parallel translation, m_x and m_y are scaling rates, and $angle_x$, $angle_y$, and $angle_z$ is rotation angle on x-, y-, and z-axis of lips shape. Each gene length is 8 bits and therefore, the total chromosome length is 56 bits.

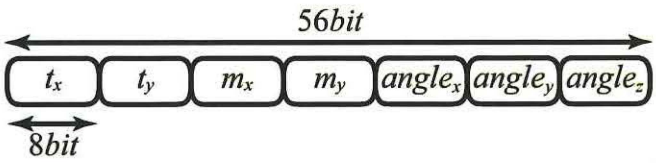


Figure 5.3 A structure of chromosome

The template's width and height should be changed separately, because of shape deformation of lips by speech is not only similarity change. Thus, we use 2-dimensional scaling by m_x and m_y. The template is transformed by homogeneous coordinates with these parameters.

5.3.2 Projective Transformation

In this part, geometric transformation is explained. This transformation solves geometric changes of lips, such as parallel translation, scaling, and three-dimensional rotation, as mentioned in Section 5.3.1.

We must perform perspective projection for the template by projective transformation. This transformation is represented by a simple combination of matrix multiplications, because homogeneous coordinate [45] is used. The matrix is obtained from the chromosome of the GA described in Section 5.3.1.

The position on the projection plane of a point in the template image is given by the intersection on the projection plane of a line that passes from the center of the projection to the point on the template image. Let A be a point on the template image, and A^* is called perspective drawing. A^* is the point that corresponds to a transformed point A on the target image. A and A^* are represented by homogeneous coordinates as follows:

$$A = [X, Y, Z, 1], \tag{5.1}$$

$$A^* = [X^*, Y^*, Z^*, 1]. \tag{5.2}$$

In this paper, the center of the projection is a point $C\,(x_c, y_c, z_c)$, the projection plane is xy-plane ($z = 0$), and a center of the template image locates at the origin.

The point A^* is given by the following simple equation.

$$A^* = AMR_xR_yR_zPT, \tag{5.3}$$

where M represents the scaling, R_x is the rotation on x-axis, R_y is the rotation on y-axis, R_z is the rotation on z-axis, T is the parallel translation, and P is a matrix of the projective transformation. These matrixes are shown as follows:

$$M = \begin{bmatrix} m_x & 0 & 0 & 0 \\ 0 & m_y & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{5.4}$$

$$R_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(angle_x) & \sin(angle_x) & 0 \\ 0 & -\sin(angle_x) & \cos(angle_x) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{5.5}$$

$$R_y = \begin{bmatrix} \cos\left(angle_y\right) & 0 & -\sin\left(angle_y\right) & 0 \\ 0 & 1 & 0 & 0 \\ \sin\left(angle_y\right) & 0 & \cos\left(angle_y\right) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{5.6}$$

$$R_z = \begin{bmatrix} \cos(angle_z) & \sin(angle_z) & 0 & 0 \\ -\sin(angle_z) & \cos(angle_z) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{5.7}$$

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ t_x & t_y & 0 & 1 \end{bmatrix}, \tag{5.8}$$

$$P = \begin{bmatrix} z_c & 0 & 0 & 0 \\ 0 & z_c & 0 & 0 \\ -x_c & -y_c & 0 & -1 \\ 0 & 0 & 0 & z_c \end{bmatrix}. \tag{5.9}$$

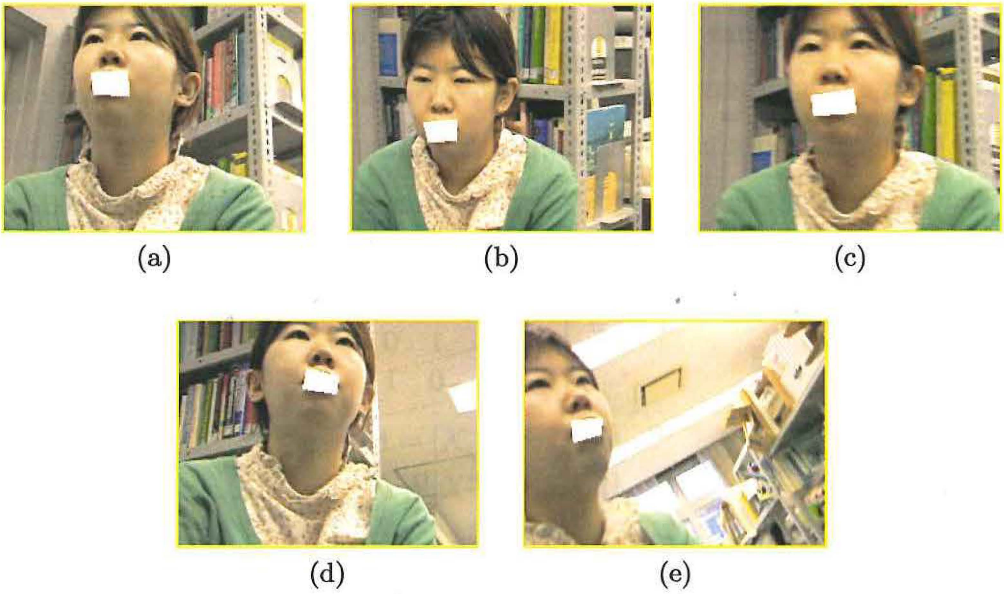
5.3.3 Fitness Function

The matching process of the template matching is evaluated by a fitness function. A fitness function is calculated by the objective function and changes dynamically. This fitness function is described in Section 3.4 as “Dynamic Fitness Function”.

5.4 Computer Simulation Results and Considerations

All simulations are performed on the same computer: CPU is Pentium4 2.0GHz. As GA parameters, we use 150 population size, 70 % crossover probability, 20 % mutation probability, and the system is terminated at 200 generations.

The best results for the target images in Figure 5.4 by the template image in Figure 5.1(b), are illustrated in Figure 5.4. The numerical parameters, that represent the lips are shown in Table 5.1. These parameters are optimized and extracted by GA. These numerical results make it possible that the detected lips region is corrected, then normalized lips region can be acquired. Figure 5.5 shows the collected lips region by the extracted numerical parameters. These collected lips has little affection


(a) (b) (c)

(d) (e)

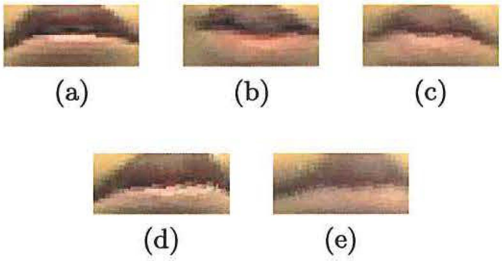
Figure 5.4 Examples of best resulting image for Figure 5.2 images.

Table. 5.1 Examples of acquired lips information.

result	Coordinate		Scaling		Rotation (deg)		
(Figure 5.4)	x	y	x	y	x	y	z
(a)	85	62	1.06	1.58	-12.76	-15.78	-1.23
(b)	72	100	0.88	1.52	-32.80	-7.27	-4.53
(c)	106	78	1.25	1.40	2.33	-11.67	-6.18
(d)	116	49	1.03	1.22	8.37	21.00	13.04
(e)	52	86	0.98	1.22	-17.16	32.53	-11.39

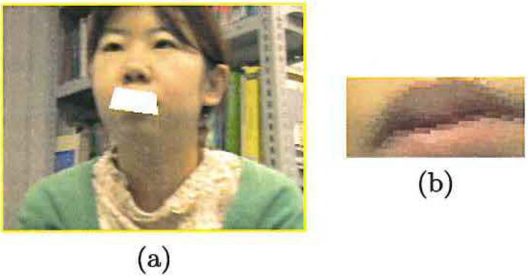
of geometric changes, such as parallel translation, scaling, and rotation.

To evaluate the proposed system, we simulate 30 times for each target in Figure 5.2, therefore the total is 150 times. The detection accuracy is shown in Table 5.2. In this table, "Failed" is the case of failure to detect the lips region, and "Successful" is success to detect it. Moreover, the total of "Best" and "Detected" is the row labeled "Successful". This "Best" case is that all parameters are good as shown in Figure 5.4, and "Detected" case is that the location is good but the rotation parameters are not good in three-dimension as Figure 5.6. As you can see the Table 5.2, accuracy to locate the lips region is good 90 %, however, to acquire the exact rotation value in three-dimensional space is not good 30 %. This is due to GA feature which a local search is not good in GA. Moreover, the average processing time is 0.94 second, this


(a) (b) (c)

(d) (e)

Figure 5.5 Examples of best extracted lips region for Figure 5.2 images.


(b)

(a)

Figure 5.6 Example of not good result: a) a detection result ; b) a corrected lips

Table. 5.2 Detection accuracy (%).

	/a/	/i/	/u/	/e/	/o/	average
Failed	3.33	13.33	0.00	16.67	16.67	10.00
Successful	96.67	86.67	100.00	83.33	83.33	90.00
Best	26.67	30.00	46.67	10.00	36.67	30.00
Detected	70.00	56.67	53.33	73.33	46.67	60.00

is very slow. This reason is a trade-off between search speed and accuracy. In other words, to obtain high accuracy GA population size must be large, however this causes the increase of the computation. To avoid these problem, search efficiency of the GA must be raised.

5.5 Conclusion

In this chapter, the lips detection system in three-dimensional space was proposed. This development toward three-dimensional space gives estimation of the face direction by the posture angle of lips. However the search speed is very slow, and it indicates that this system cannot apply to obtain the exact rotation angle in three-dimension in real-time. The reason is that a trade-off between exploration accuracy and speed. In the next chapter, we will improve this trade-off by "downsized GA".

Chapter 6

High Speed and Accuracy Lips Region Detection by Downsized GA

6.1 Introduction

Genetic Algorithms (GAs) have been applied successfully to optimise solutions in image processing [8, 25, 51, 52]. In Chapter 3, the genetic object detection and numerical parameters extraction system was described. The object is lips, and has complex changes, such as shape deformations and two-dimensional geometric changes. Human speech causes the lips shape deformations, and free camera motion causes the geometric changes. Moreover, the genetic object detection and numerical parameters extraction system in three-dimensional space was described in Chapters 4 and 5.

In this chapter, high-speed object detection, tracking, and information extraction of the object, is proposed. Our approach is based on a template matching with GA. GA is a probabilistic search technique which is suitable for the exploration of large and complex search spaces. Typically, the GA has a trade-off between exploration accuracy and speed. In other words, to obtain high accuracy, the size of the population and the number of generations must be increased. Therefore, high-speed exploration with keeping high accuracy—downsizing of GA is necessary.

In the proposed system, the object is lips region, because of the following reasons.

A speech recognition is one of the major non-contact interface, and useful for mobile devices. Many studies of audio-visual speech recognition [3, 5, 6, 7, 8, 9, 53] have been reported, in order to overcome a performance limitation and background noise by various situations. For mobile devices, some problems must be solved, which are listed below.

1. Complex target image (not only face region)
2. Drastic change of whole scene by a camera motion
3. Acquisition of numerical information of lips geometric changes
4. Real-time processing with keeping high accuracy

The lips geometric change information is used for correction of lips region. This correction is important, because a normalised lips region by this correction will make speech recognition easy.

As examples of approach for lips image, eigenlip methods [9, 7] have been proposed. In these methods, the training data must be chosen carefully to include all possible lips configurations. Active Shape Model [6], and Genetic Snakes [8, 25] which is an improved version of Snakes [26] have been proposed. These approaches have some constraints, such that a target image is only a face region and a subject wears a helmet with a camera to obtain a mouth image, because of the initial setting problem. Therefore, these approaches are difficult to be applied to our purpose. On the one hand, high speed face tracking method [29] was proposed. In this method, the many facial feature patch templates must be prepared as training set. These templates are regions surrounding the feature, such as eye and mouth. Therefore, the geometric information of lips region cannot be extracted at the detection. These methods by using a whole face are difficult to be applied to our purpose. Because, it is hoped that the lips region is detected and its geometric information is extracted directly for the real-time processing.

To overcome the problems which are listed above, we use simple template matching with downsized GA. This downsizing means speed up searching with keeping the accuracy. The downsizing GA is carried out by control of a search domain with a very small population. The search domain is controlled automatically. The chapter is organised as follows: at first, basic parts of GA are mentioned in Section 6.2. Our approach is explained in the next Section 6.3. Section 6.4 shows an evaluation of the approach. Finally, Section 6.5 gives a conclusion.

6.2 Basic Parts of Genetic Algorithm

6.2.1 Structure of Chromosome

A chromosome is a solution candidate to be optimized. In other words, chromosomes specify parameters which represent coordinates, scaling and rotation of an object to be explored on the target image. The same chromosome as Section 3.3 is used in this proposed method.

6.2.2 Fitness Function

The matching process of the template matching is evaluated by an objective function. A fitness function is calculated by the objective function and changes dynamically. This fitness function is described in Section 3.4 as "Dynamic Fitness Function".

6.3 Downsized GA with Search Domain Control

Typically, the GA has a trade-off between exploration accuracy and speed. In other words, to obtain high accuracy, the size of the population and the number of generations must be increased. This part describes the method to overcome this trade-off. This method is called "Search Domain Control (SD-Control)".

6.3.1 Why to Do It

The method described in Chapters 3-5 may be unsuitable for the motion image sequence, because of slow performance. The slow performance is due to the trade-off between exploration accuracy and speed. In other words, to obtain high accuracy, the size of the population and the number of generations must be increased. This means that the search speed of the GA is reduced. From earlier experiences, when the size of the population and the number of generations are decreased, GA individuals can be stuck in local optima, as shown in Figure 6.1. This is attributed to the fact that the distribution of the redness (refer to Section 2.2) is similar to lips region. This indicates that the GA is a global optimization algorithm and is not good for local optimization [21].

As a search efficiency improvement, we can use a technique, in which after the domain including the optimal solution is specified, its neighborhood is searched in detail.

In GAs, the search starts from a population of many points, rather than starting from just a single point. This parallelism means that the search will not become trapped in local optima. GA tries to escape from the local optimum and to find the global optimum by crossover and mutation operators. If the population is too small, GA converge prematurely and is trapped into a local optimum. As a search efficiency improvement, we can use a technique, in which after the domain including the optimal solution is specified, its neighbourhood in the target image is searched in detail.

Generally speaking, this is a risky method, because it is not guaranteed that the domain where that optimal solution is included clearly, can be specified. In other words, this method cannot find the optimal solution because of premature convergence to local optima. However, this is not a critical problem for our system, because in our simulations, typical local optima is a part of the face (see Figure 6.1). This reason is that skin area is included in the template image and redness data (refer to Section 2.2) is used. This means that it is highly possible that the optimal solution is in the neighborhood of local optima.

With controlling the search domain of GA, we expect that the GA became easier to escape from the local optimum, and the GA can work as not only global optimization but also local optimization. We therefore hope that, by decreasing the population and controlling the search domain, high accuracy and high speed exploration can be achieved.

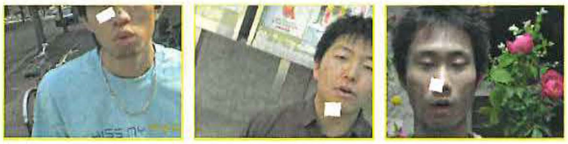


Figure 6.1 Examples of the local optimum: GA individuals can be stuck in local optima, optimization is failed.

6.3.2 What to Do and How to Do It

The location and the size of the search domain are controlled. The search domain is controlled depending on both an elite individual and the number of generations. The elite individual can be found out by comparison of the objective value of all individuals. The location of the search domain is decided by a coordinate of the elite individual. In other words, the location of the search domain and the elite individual is same, and these locations change together. This coordinate is obtained from t_x and t_y of the elite chromosome (see Figure 3.2). The search domain center is set to this coordinate.

Next, the size of the search domain is decided by the number of generations. The search domain is renewed as follows:

$$\begin{bmatrix} width^* \\ height^* \end{bmatrix} = \alpha \begin{bmatrix} width \\ height \end{bmatrix} \tag{6.1}$$

In equation (6.1), *width* and *height* are transformed to *width** and *height** respectively by α which is a scale factor. This α is changed by the number of generations because the search of GA keeps getting near to the global optimum gradually according to the number of generations. In this paper, simply, we use the following equations for decision of α.

$$\alpha = \begin{cases} 1 & (generation < 10) \\ 0.5 & (10 \leq generation < 50) \\ 0.375 & (50 \leq generation < 75) \\ 0.25 & (75 \leq generation) \end{cases} \tag{6.2}$$

where *generation* is the number of generations. In equation (6.2), the search domain is reduced in multi-step as evolution progresses.

6.3.3 What is Going on Outside?

You may wonder what is going on outside of the search domain. Figure 6.2 illustrates the process of the object detection by GA to explain this problem. In Figure 6.2, the small filled rectangle region is the detected lips region. The not-filled rectangle

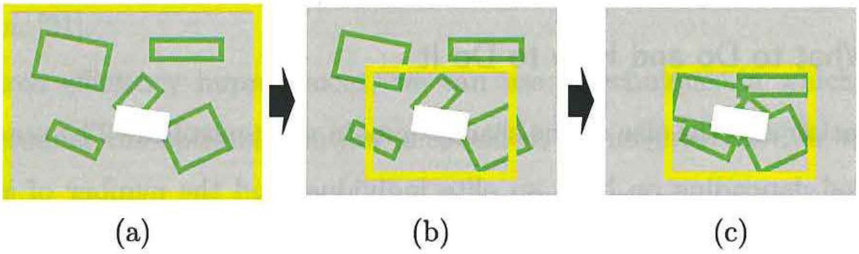


Figure 6.2 What is going on outside of the search domain?: a) early stage of GA evolution ; b) the search domain move to the elite individual ; c) re-coding of other individual

regions are other individuals. The larger square frame represents the search domain by the SD-Control method.

Figure 6.2(a) shows the early stage of the exploration, and the search domain is the whole target image. Some individuals can be located outside of the search domain during the search domain change using the SD-Control, as illustrated in Figure 6.2(b).

The easiest way is to eliminate these outside individuals and create new individuals. However, this is not a best solution because all genes may evolve into good direction except genes that represent the position. Therefore, whenever the search domain is changed, only t_x and t_y in the chromosome is re-coded for all individuals by a new search domain, as illustrated in Figure 6.2(c). This re-coding is calculated by the following equation.

$$location_{int} = \frac{location_{real} - S_{min}}{S_{max} - S_{min}} \times \left(2^{bit} - 1\right), \tag{6.3}$$

where $location_{int}$ and $location_{real}$ are integer value and real value of a re-coded gene, S_{min} and S_{max} are the minimum and the maximum coordinate of the search domain, respectively.

Using this process, other genes are inherited to the next generation. This means that the GA optimization is controlled.

6.3.4 Flow Chart

Flow charts of our system are illustrated in Figure 6.3.

At first, an initial population is generated and after that the GA process is started.

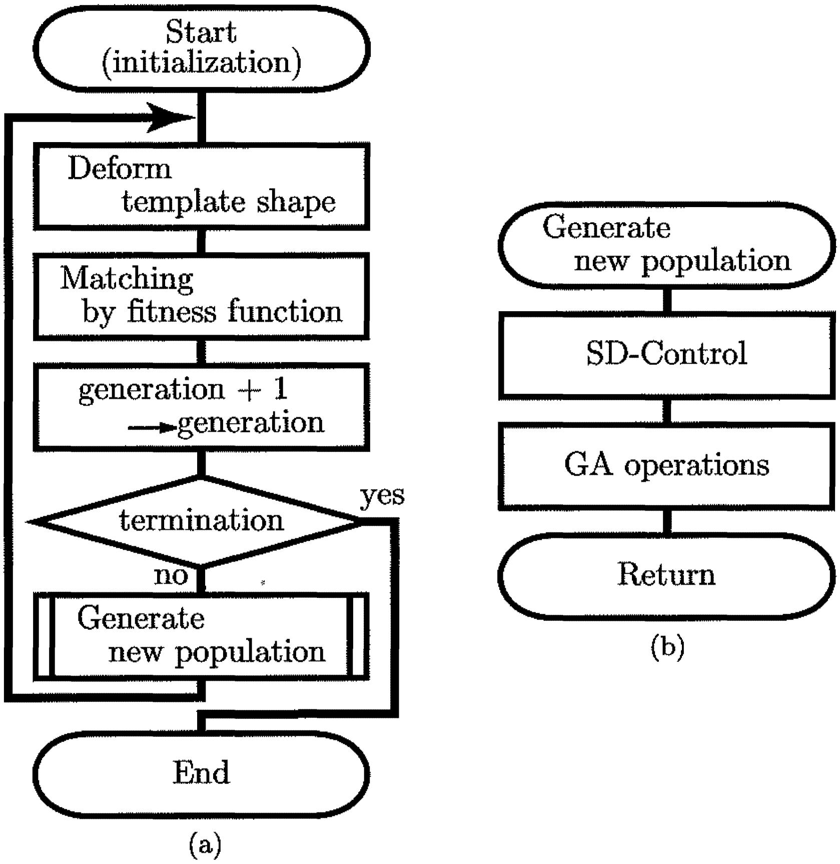


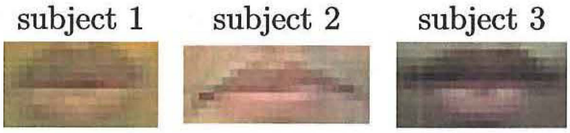
Figure 6.3 Flow charts: a) main GA process ; b) generate a new population.

In GA processing, the template shape is deformed to an unique "square annulus" from a normal square, as explained in Section 3.2. Then, the matching process is executed between the template and a target image using the fitness function. The generation is increased, until a termination condition of GA is satisfied. In this study, the GA is terminated by the number of generations. If the termination criterion is not satisfied, a new population of the next generation is generated according to the fitness of each individual (Figure 6.3(b)). In this process, the search domain is controlled dynamically, as described in Section 6.3. This technique is the main part of this proposed system, and achieves high speed and the high accuracy lips detection. After GA process is completed, the result is obtained as numerical data. This numerical data represents the lips information and can be used in many applications as described in Section 1.

6.4 Computer Simulation Results and Considerations

In this section, simulations to evaluate the proposed system and considerations of the results are described. We estimate the effectiveness of our proposed system by comparison between it and the system without the SD-Control (hereinafter referred to as "the no SD-Control"). All simulations are performed on the same computer: CPU is Pentium4 2.0GHz.

6.4.1 Input Images



subject 1 subject 2 subject 3

Figure 6.4 Template images

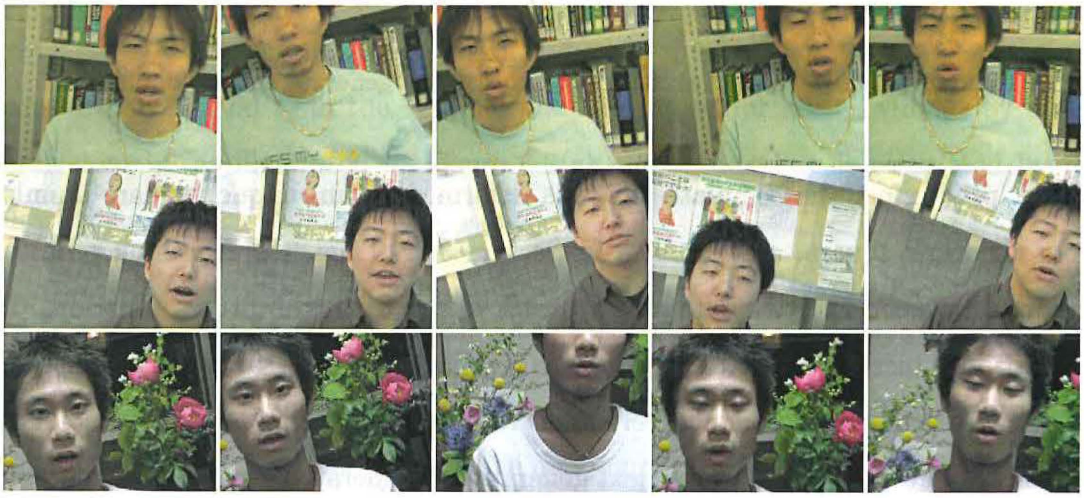


Figure 6.5 Target images: : first row is subject 1, second is subject 2, and third is subject 3. Subjects make the Japanese vowel sound: starting from the left; /a/, /i/, /u/, /e/, and /o/.

The template images are illustrated in Figure 6.4. Template image size of subject 1 is 20 × 11 pixels, subject 2 is 24 × 10 pixels, and subject 3 is 22 × 11 pixels.

Figure 6.5 shows target images. The images captured using a shaking video camera include a face and background with some red color while each of three subjects pronounces the vowels. Target images are cut from the motion image sequence. In consideration of the use by mobile devices, the lips region on the target images has some considerable geometric changes based on the template image. These geometric changes in this chapter mean parallel translation, scaling, and two-dimensional rotation. Parameters represented these geometric changes can be regarded as the solutions of GA (See Figure 3.2). All target images size are 240 × 180 pixels.

6.4.2 GA Configurations

Other GA settings are explained in this part. As mentioned in Section 3.3, we use the binary genotype. We choose uniform crossover, because of its many advantages [50]. In this system, five parameters must be prepared as follows.

- Population size
- Crossover probability
- Mutation probability
- Scaling window size
- Termination criterion

These parameters affect the efficiency of GA exploration. The relation between the population size and search speed is described in Section 6.3. The crossover and mutation probabilities are decided after many trials. The scaling window is explained in Section 3.4. For the GA termination, a variety of termination criterion has been used, such as the change of fitness value and the number of generation. In this system, we use the number of generations as described in Section 6.3.4, because of its simplicity and low computational cost. Final values of these parameters used in simulations are shown in Table 6.1.

Table. 6.1 Parameters in the simulation of estimate the proposed system

Population Size	Crossover probability	Mutation probability	Scaling window size	Termination criterion
10	0.7	0.15	1	200

The parameters of the genetic algorithm are: population size is very small 10, probability of crossover is 0.7, and probability of mutation is 0.15. Parameters of the ignored region of the "square annulus" (refer to Figure 3.1) are set $w'/w = 0.8$ and $h'/h = 0.5$. We use $n = 1$ in equation (3.14) of the Dynamic Fitness Function method. The GA is terminated at 200 generations. We demonstrate the effectiveness of the SD-Control method in the next part.

6.4.3 Effectiveness of Search Domain Control

This part shows the effectiveness of SD-Control by the transition of an actual GA search.

The location of the search domain and the elite individual change together, and the size of the search domain changes by equations (6.1) and (6.2), as explained in Section 6.3.

The transition of GA exploration by the objective value (Section 3.4) on the proposed system is plotted in Figure 6.6. The solid line and the dashed line show the objective value of the elite individual and the average objective value of the population. A visual transition are shown in Figure 6.7 to lead the reader to understand. In both figures, alphabets from "a" to "o", where the generation is 0, 5, 10, 15, 20, 25, 50, 75, 125, 130, 135, 140, 145, 150, and 199, are correspondent with each other. In Figure 6.7, the small filled rectangle region is the detected lips region. The not-filled rectangle regions are other individuals. The larger square frame represents the search domain by the SD-Control method. These cases are the most typical examples.

The effectiveness of the SD-Control method is described with linking Figures 6.6 and 6.7. The objective value is worst in early stage of GA evolution, and a part of wall and a human in the poster are detected as lips region. The SD-Control is started in "c"(the number of generation is 10). The search domain moves to a local optimum and the size is reduced. However, in "d" the search domain moves to a face region. This reason is that the search domain includes a part of face, and the template image includes a part of skin (see Figure 6.5).

The population is changing in region "A", which includes "e" to "i", however the

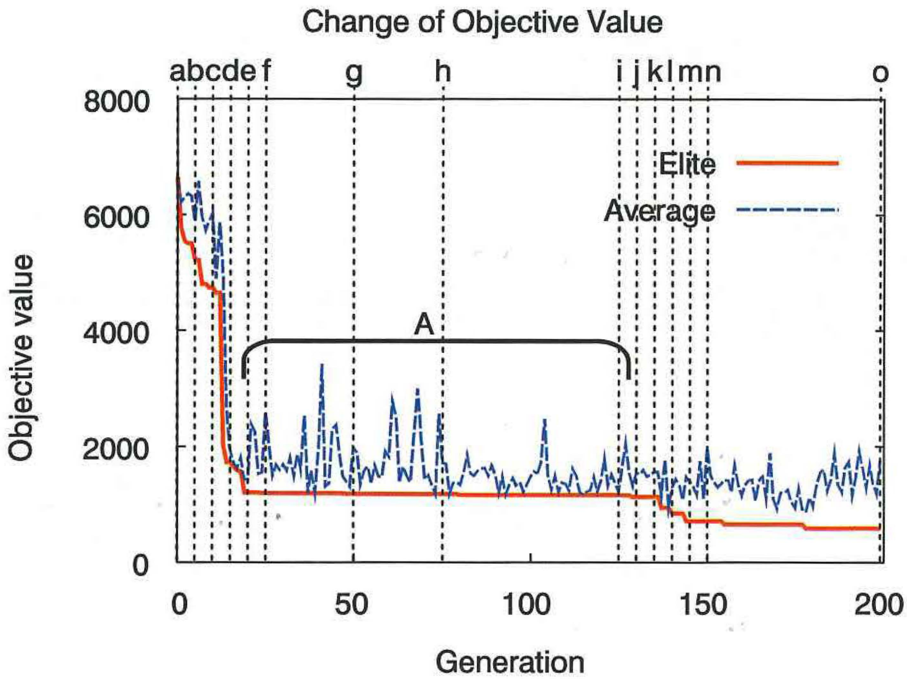
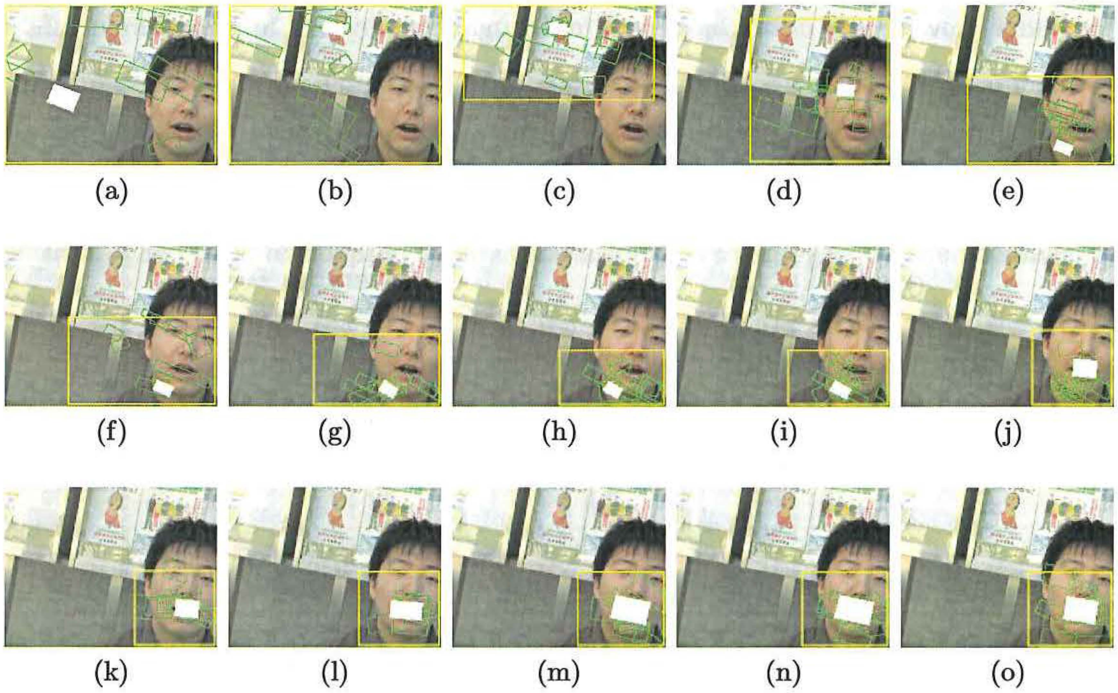


Figure 6.6 Relation between generations and objective values: "A" is the region where GA is trapped in a local optimum, and from "a" to "o" are points where the generation is 0, 5, 10, 15, 20, 23, 50, 75, 125, 130, 135, 140, 150, and 199. These alphabets are correspondent with alphabets in Figure 6.7

objective value of the elite individual and the elite individual does not change in the visual translation. The GA exploration converges into a part of jaw in this "A" region. This is attributed to the fact that the distribution of the redness (refer to Section 2.2) is similar to lips region. This shows that GA individuals stuck in a local optimum. In this region, the size of the search domain is reduced at "g" and "h", as described in equations (6.1) and (6.2). The elite individual cannot escape from the local optimum by the SD-Control in "g". Although the search domain size becomes minimum in "h", the elite individual does not change.

At "j", the elite individual evolves near the lips region, and the search domain moves. This indicates that the GA evolution of the proposed system can escape from the local optimum. The reason for this is that a search domain is reduced by the SD-Control and the GA can explore the reduced search domain with meticulous detail



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

(f) (g) (h) (i) (j)

(k) (l) (m) (n) (o)

Figure 6.7 Visual transition: these alphabets are correspondent with alphabets in Figure 6.6

from "h" to "j". The GA evolution gradually closes to the optimal solution from "j" and finally the lips region is detected and lips information is extracted as illustrated in "o".

In other words, the GA in the proposed system performs not only global optimization but also local optimization. The result of our demonstration clearly shows that a trade-off between exploration accuracy and speed is overcome by the SD-Control method.

6.4.4 Resulting Images

Figure 6.8 shows examples of results obtained from the computer simulation. The filled rectangle region is the extracted lips region, and the rectangular frame which contains that is the final search domain. The shape deformations of lips by speech are extracted exactly as shown in Figure 6.8.

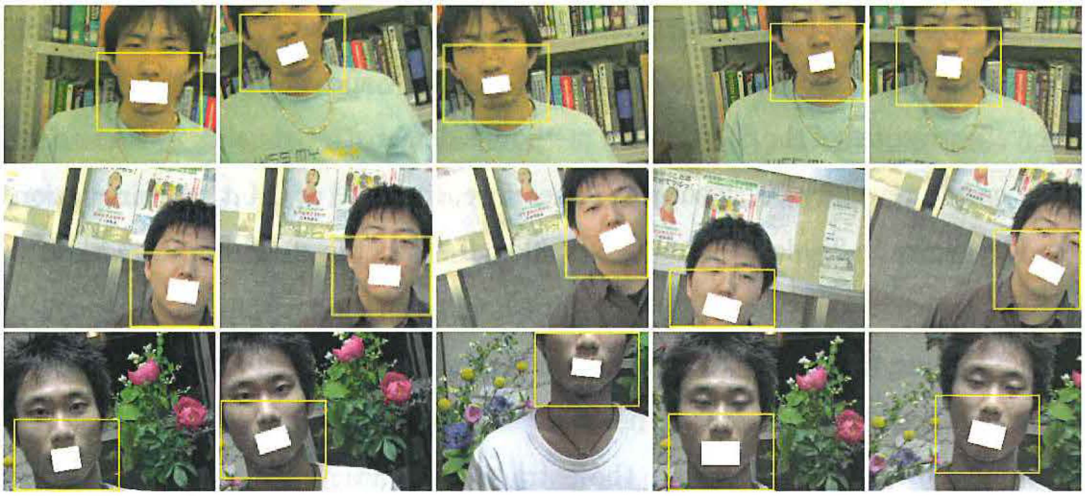


Figure 6.8 Examples of successful result image for Figure 6.5 images

6.4.5 Demonstration of Search Domain Control

The solution obtained by a manual operation, is called a true solution. Our method results are judged to be good or not good by comparison with the true solution. The comparison is performed by the following equations.

$$\begin{cases} T-3 \le t \le T+3 \\ M \le m \le 1.3 \times M \\ ANGLE - 5^\circ \le angle \le ANGLE + 5^\circ \end{cases} . \tag{6.4}$$

These capital letters are the solution obtained manually, and small letters are a solution obtained by the proposed method. t represents the x or y-coordinate, m is a scaling rate and *angle* is a rotation angle. If a result satisfies these conditions, the results is acceptable for many applications described in Chapter 1.

Table. 6.2 Examples of acquired numerical lips information by manual and the proposed system

type of solution	t_x	t_y	m_x	m_y	*angle* [deg]
manual	164	100	1.776	2.260	5.70
GAs	165	99	2.160	2.327	3.98

Table 6.2 shows lips data for Japanese vowel /a/ of subject 1 in Figure 6.8. All system solutions of Table 6.2 satisfy the conditional equation (6.4). These results

indicate that the proposed system can robustly and accurately detect lips region in images taken by free camera motion and deformed by speech.

The effectiveness of the SD-Control method is demonstrated using 20 times simulations per one vowel for every subject. Therefore, the total of 300 times simulations is tested as shown in Tables 6.3 and 6.4. Table 6.3 shows the result of our method with the SD-Control method, and Table 6.4 the is normal method without that. From these tables, by using the flexible search domain control, we obtain a better result than the normal method. In both cases, the processing time is very high speed, because of the population size is 10. About the extraction accuracy, our proposed method is 96.67 %, and the normal method is lower 74.00 %. At the final stage of search, search efficiency is improved. In the normal method, the solution of the failure is the local optimum by premature convergence, because the population size is too small. In the proposed method, the search domain is reduced and close to the optimal solution, and GA can search in more detail by the re-coded chromosome, at the multiple stages. In other words, the present GA acts as not only global search but also local search.

Table. 6.3 Results of simulation obtain using the proposed method (until 200 generations)

	/a/	/i/	/u/	/e/	/o/	total
accuracy [%]	98.33	95.00	91.67	98.33	100.00	96.67
processing time [msec]	35.44	35.09	34.29	34.03	37.83	35.34

Table. 6.4 Results of simulation obtain using the normal method (until 200 generations)

	/a/	/i/	/u/	/e/	/o/	total
accuracy [%]	71.67	75.00	68.33	85.00	70.00	74.00
processing time [msec]	36.40	35.65	36.75	34.71	37.41	36.18

6.5 Conclusion

This chapter presents high-speed object detection, tracking, and information extraction of the object, as improvement of proposed methods, which described in previous chapters. Our approach is based on a template matching with GA. The GA is a probabilistic search technique which is suitable for the exploration of large and complex search spaces. However, the GA has a trade-off between exploration accuracy and speed. In other words, to obtain high accuracy, the size of the population and the number of generations must be increased. In order to avoid this trade-off this chapter presented a new lips extraction method by GAs, which has a simple algorithm, a high speed and a high accuracy is achieved. This proposed method controls the search domain. We demonstrated the effectiveness of this method and compared the proposed method with the previous normal method by small population. The results of simulations show our proposed method is more effective than other ones. By the the SD-Control method, the GA can act as not only global search but also local search. The downsized GA is achieved because of high accuracy and high speed — using small population.

In the next chapter, the evolutionary video processing, the main part of this dissertation, is described. The SD-Control method is basis of the evolutionary video processing.

Chapter 7

Real-Time Lips Region Detection by Evolutionary Video Processing

7.1 Introduction

Image Understanding is the process to understand the content of images in order to automate visual tasks by computers. The technical challenge is to make the computer understand the contents of the images. In other words, the most difficult problem is to automatically produce a reasonable description from an image. It is clear that the nature of images and descriptions have a bug distance. In the fields of Artificial Intelligence, Scene Analysis, Image Analysis, Image Processing, and Computer Vision, the many researchers work on reducing of this distance in the last twenty years.

However, there is few Image Understanding systems which are suitable for practical use. The reasons is that it is difficult to extract the relevant information to represent the object stably, supporting real-world. The object has complex changes by various causes. A new object detection and information extraction approach, which can be applied to these various changes, is necessary for Image Understanding.

Previous chapters dealt with a single image, and described the genetic object detection and extraction of the information, which represents the object. These approach has a trade-off between the processing speed and the accuracy, then the previous system cannot be applied to the real-time processing.

Chapter 6 introduced the downsized GA to overcome the trade-off for a single image processing. The processing time comes a step closer to the real-time processing.

In this chapter, we describe a new video processing which can be applied to a real-time processing, by a flexible control of search domain and inheritance of genetic information between video frames. The detection object is a lips region of a talking person as previous chapters.

It is very important to accurately detect and track lips region in real-time from natural video scenes for many applications, such as audio-visual speech recognition, video compression, and robot perception.

In order to make recognition processing simple and easy, it is preferred that a detected the lips region is normalized geometrically in relation to position, scaling, and rotation. For a real-time application, the system should detect lips region and extract this normalization information at the same time. In this chapter, these geometric change information to be treated is parallel translation, scaling, and rotation, by free camera work in natural scenes.

In this chapter, we address three issues for lips detection and tracking as follows.

1. Active scene by free camera motion.
2. High accuracy in detection of lips region and extraction of lips geometric information.
3. High processing speed.

The chapter is organized as follows: at first, basic parts of GA are mentioned in Section 7.2. Section 7.3 presents our technique. Section 7.4 shows an evaluation of the system, and Section 7.5 gives a conclusion.

7.2 Basic Parts of Genetic Algorithm

7.2.1 Structure of Chromosome

A chromosome is a solution candidate to be optimized. In other words, chromosomes specify parameters which represent coordinates, scaling and rotation of an object to be explored on the target image. The same chromosome as Section 3.3 is used in this proposed method.

7.2.2 Fitness Function

The matching process of the template matching is evaluated by an objective function. A fitness function is calculated by the objective function and changes dynamically. This fitness function is described in Section 3.4 as "Dynamic Fitness Function".

7.3 Evolutionary Video Processing

7.3.1 Flexible Search Domain Control

The methods which we have proposed may be unsuitable for the motion image sequence. When it tries to obtain high accuracy, increase of a population and the number of generations occurs. From our experience, when the population and the number of generations are decreased, GA individuals can be stuck at local optima as in Figure 6.1. This is because the GA is a global optimization algorithm. The search starts from a population of many points, rather than starting from just a single point. This parallelism means that the search will not become trapped on local maxima. GA tries to escape from the local maximum and to find the global optimum by crossover and mutation operators. If the population is too small, GA converge prematurely and is trapped into a local optimum. As a search efficiency improvement, we can use a technique, in which after the domain including the optimal solution is specified, the neighbourhood of that is searched in detail. However, generally speaking, this is a risky method. Because the domain where the optimal solution is included clearly, cannot be specified. Against that, in our many past experiments, we find that a part of a face is extracted as lips region, in case of the failure (see Figure 7.1). This reason is that we use x component (redness) in the Yxy color space [31] which is used in this image data. Therefore, we hope that, even by decreasing the population, by control of the search domain, high accuracy extraction becomes possible and also extraction speed becomes high.

The search domain is controlled depending on both an elite individual and the number of generations. The elite individual can be found out by comparison of the

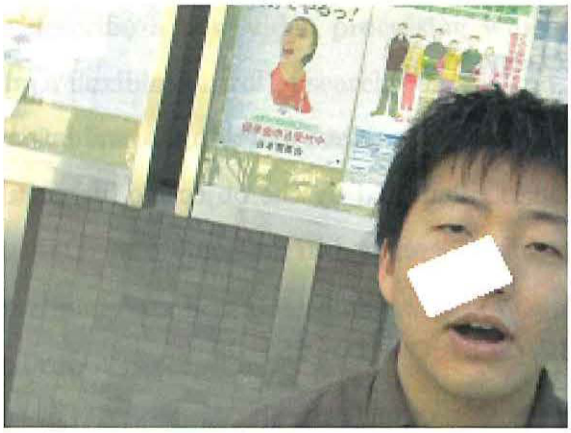


Figure 7.1 An example of the local optimum

objective value of all individuals. The location of the search domain is decided by a coordinate (t_x and t_y, see Figure 3.2) of the elite individual. The search domain center is set to this coordinate.

Next, the size of the search domain is decided by the number of generations. The search domain is renewed as follows:

$$\begin{bmatrix} width^* \\ height^* \end{bmatrix} = \alpha \begin{bmatrix} width \\ height \end{bmatrix}. \tag{7.1}$$

In equation (7.1), *width* and *height* are the target video frame's width and height, and transformed to *width** and *height** respectively by α which is a scale factor. The α is controlled by the number of video frames, the coordinate of elite individual, and the number of generations. In particular, the detection in the first frame is very important. Because, if this lips region detection is failure, there is a probability to fail in the next one, so that the detection information is herited to the next one. If this lips region detection is failure at the first frame, the probability of failure become strong in the next one. Because, the genetic information in the last frame is recycled for high speed and high accuracy detection and tracking. Therefore, the search domain is full range in case of the first frame and *generation* < 10. The value of α can be defined as follow:

$$\alpha = \begin{cases} 1 & (generation < 10) \\ 0.5 & (10 \leq generation < 50) \\ 0.375 & (50 \leq generation < 75) \\ 0.25 & (75 \leq generation) \end{cases}, \tag{7.2}$$

where *generation* is the number of generations. In case that it is not the first frame, the value of α can be defined as:

$$\alpha = \begin{cases} 0.5 & (generation < 50) \\ 0.375 & (50 \leq generation < 75) \\ 0.25 & (75 \leq generation) \end{cases} . \tag{7.3}$$

In both equations (7.2) and (7.3), the search domain is reduced in multistep as evolution goes on. An important point is that equations (7.2) and (7.3) show that the α can be reset from 0.25 to 0.5. Because, two functions are necessary; the first is a self-repairing function in case of failure in the last frame, and the second is a fail-safe function in case of the considerable parallel translation of lips region.

Some individuals can be located on outside of the search domain by the control of search domain. All genes may evolve for good direction except genes which represent the position. For this reason, these individuals cannot be eliminated. As a solution, whenever the search domain is changed, only t_x and t_y in the chromosome is re-coded for all individuals by a new search domain. By this process, other genes are inherited to the next generation. This means that the GA optimization is controlled.

7.3.2 Inheritance of genetic information between video frames

In case of video processing, it is very difficult to use information of between video frames. Generally, in order to detect a moving object, an inter-frame difference picture is used as the information of between video frames. However, it is difficult to use the difference picture in our system, because the camera moves intensively.

Therefore, we use genetic information as a relation between video frames. In fact, without making a new population, lips detection for a next frame is proceeded with a population used in last frame. This method is very important in our proposed system, because initial exploration is reduced in GA by this method.

7.3.3 Flow Chart

Flow charts of our system are illustrated in Figure 7.2. Figure 7.2a represents the main part of our system. In Figure 7.2b, GA process is represented. New population

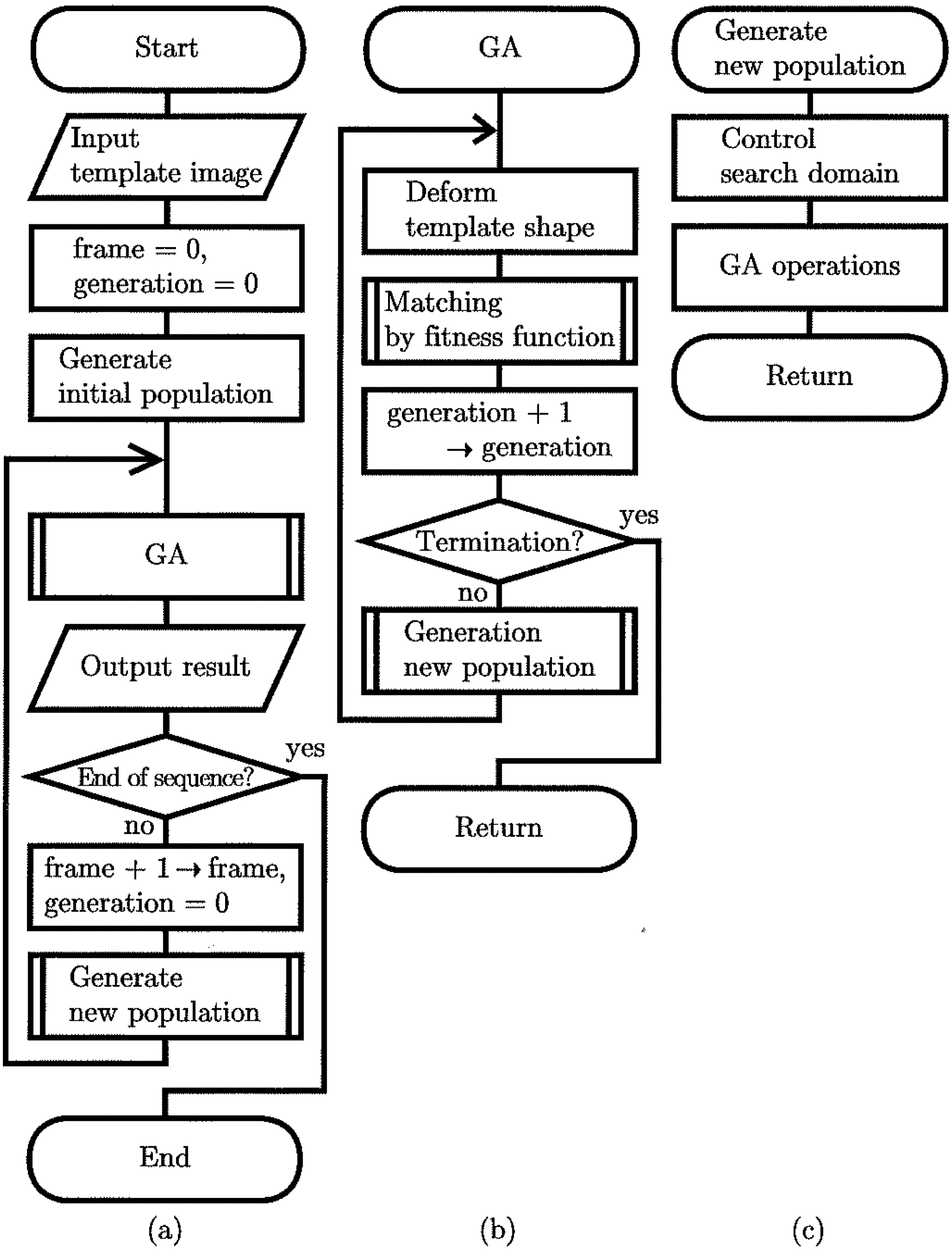


Figure 7.2 Flow charts: a) main process ; b) GA process ; c) generate a new population.

is made in In Figure 7.2c. In Fig 7.2, "frame" and "generation" are variables, which count the number of frames in video sequence and the number of generations in GA.

Our approach consists of dual-loop; the outside loop is for video sequence (see Figure 7.2a), and the inside loop is for GA (see Figure 7.2b). At first only one

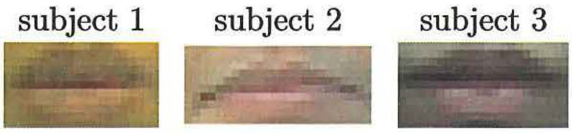


Figure 7.3 Template images

template is prepared, which is a closed mouth (see Figure 7.3). Image data put into the system is expressed with the x component (redness) in the Yxy color space [31].

After an initial population is generated, GA is started. In GA processing, the template shape is deformed to an unique "square annuls" (see Figure 3.1) from a normal square in Figure 7.2b. Then matching process is executed between the template image and a target frame. The generation is increased, till a termination condition of GA is satisfied. In this paper, GA is terminated if *generaion* > 200. If the termination criterion is not satisfied, a new population of the next generation is generated according to the fitness of each individual (Figure 7.2c).

In this process, the search domain is controlled dynamically by the number of generations. This method is described in Section 7.3.1. After GA process is completed, the results are obtained as an image and numerical data. Then, a new process begins for the next frame. At this time, some genetic information of the last GA are inherited to the new GA process. This method is described in Section 7.3. By the flexible search domain control and the inheritance of genetic information, we can detect the lips region and extract its geometric information with high accuracy in real-time.

The above process is continued to the end of the video sequence.

7.4 Computer Simulation Results and Considerations

7.4.1 Input Images and Video Sequence

At first, we input one template image, after that, target video frames are read sequentially. The template images are illustrated in Figure 7.3, which are prepared for subjects and situations. The template image size of subject 1 is 20 × 11 pixels, subject 2 is 24 × 10 pixels, and subject 3 is 22 × 11 pixels.

The target video frame examples are presented in Figures 7.4-7.6. These examples are trimmed from a video sequence, which is taken by a digital video camera. Some red color objects are included in the background, such as people in a poster, some flowers, and bicycles. The subjects repeat pronunciation of the vowels. Moreover, on the assumption that the camera moves and joggles by free hand, the shaking of scene is caused artificially by hand. Therefore, the lips region has some geometric changes. All target video frame size are 240 × 180 pixels. For the simulations, we use five seconds video sequence (150 frames) with 30 frames per second.

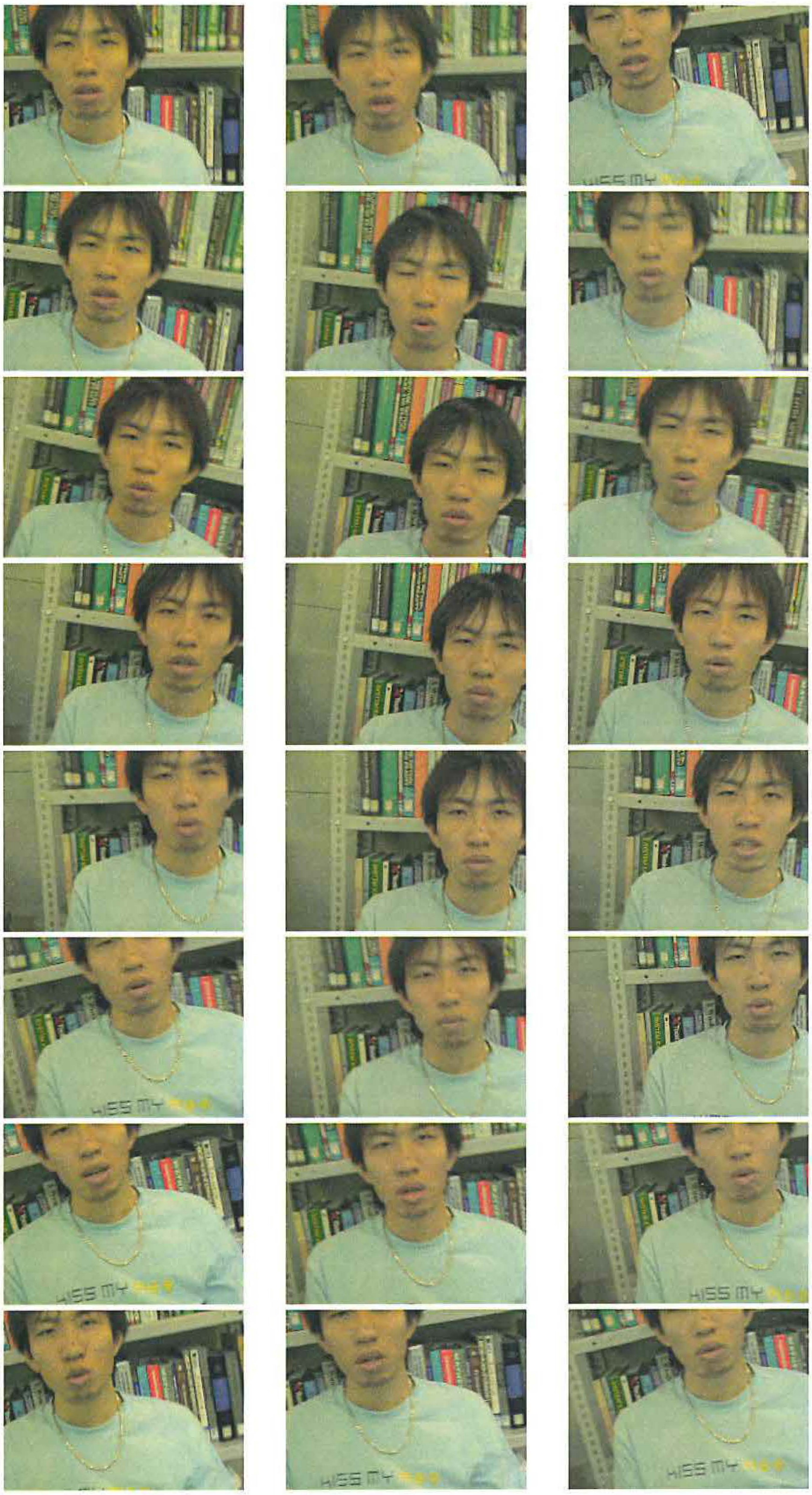


Figure 7.4 Target video frame examples (subject 1): see from top left to bottom right.

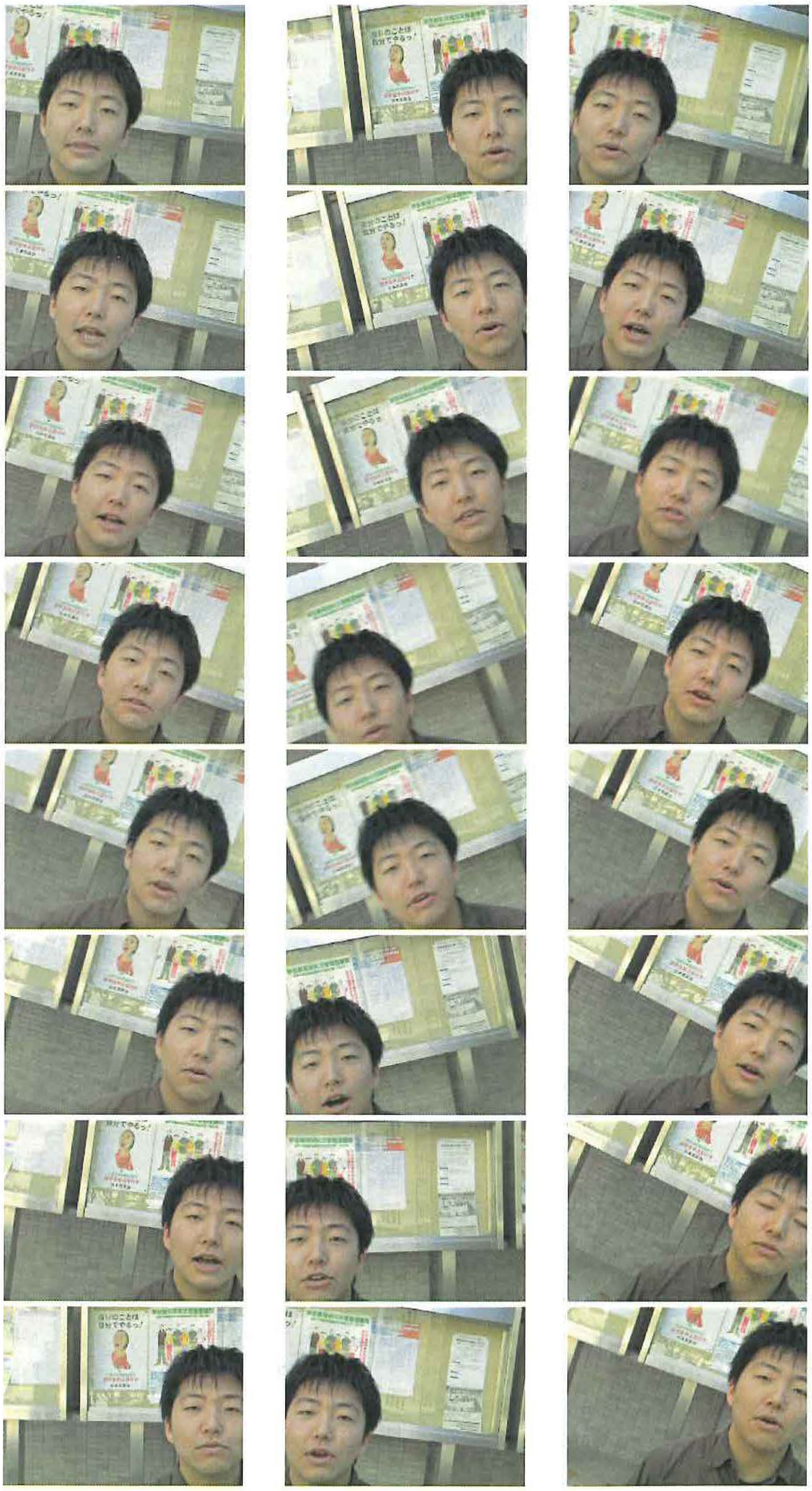


Figure 7.5 Target video frame examples (subject 2): see from top left to bottom right.

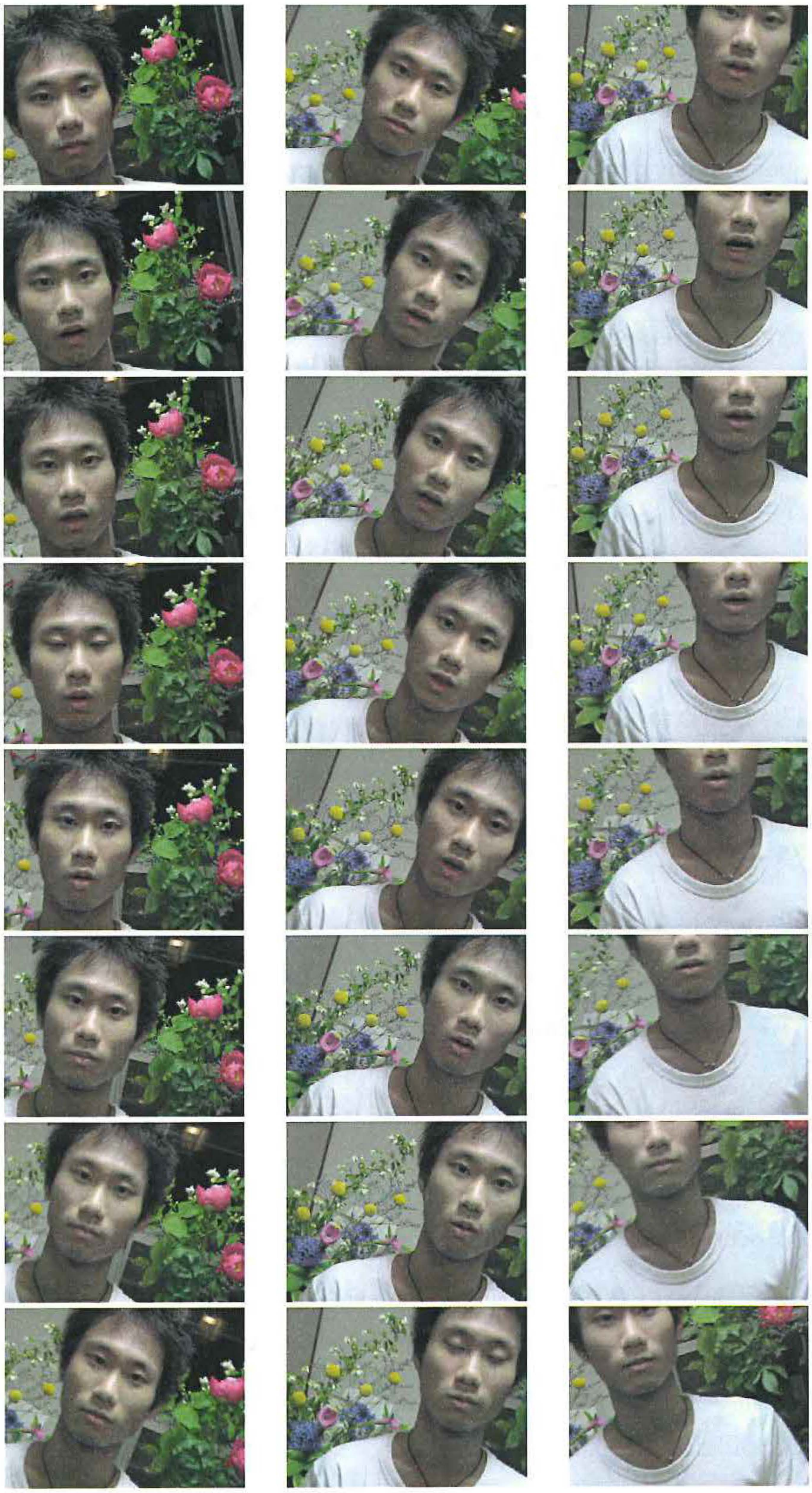


Figure 7.6 Target video frame examples (subject 3): see from top left to bottom right.

7.4.2 GA Configurations

Other GA settings are explained in this part. As mentioned in Section 3.3, we use the binary genotype. We choose the uniform crossover, because of its many advantages [50]. In this system, five parameters must be prepared as follows.

Table. 7.1 Parameters in the simulation of the proposed system

Population Size	Crossover probability	Mutation probability	Scaling window size	Termination criterion
10	0.7	0.15	1	200

The parameters of the genetic algorithm are: population size is very small 10, probability of crossover is 0.7, and probability of mutation is 0.15. Parameters of the ignored region of the "square annulus" (refer to Figure 3.1) are set $w'/w = 0.8$ and $h'/h = 0.5$. We use $n = 1$ in equation (3.14) of the Dynamic Fitness Function method. The GA is terminated at 200 generations. We demonstrate the effectiveness of the SD-Control method in the next part.

7.4.3 Results of Simulations and Considerations

For evaluation of the proposed system, we tried six times simulations for each subject with five seconds video sequence with 30 frames per second.

Figures 7.7-7.9 shows tracking results of simulations. In these results, the filled rectangle region is the detected lips region. The outside rectangular, which includes that, represents the final search domain. In Figure 7.8, the thirteenth frame of subject 2 failed, however the next frame can be repaired by the flexible search domain control described in Section 7.3.1. This failure occurs by frame-out of a little bit part of lips region. Such a self-repair worked in all simulations.

We simulated six times for each three subjects by five seconds video sequence (150 frames). The number of frames are 6 × 150 frames for each subject, and the total number of frames is 2,700. In Table 7.2, accuracy and processing time are shown. This processing time is the average time to detect and track the lips region and to output its geometric change information for five second, without frame input/output

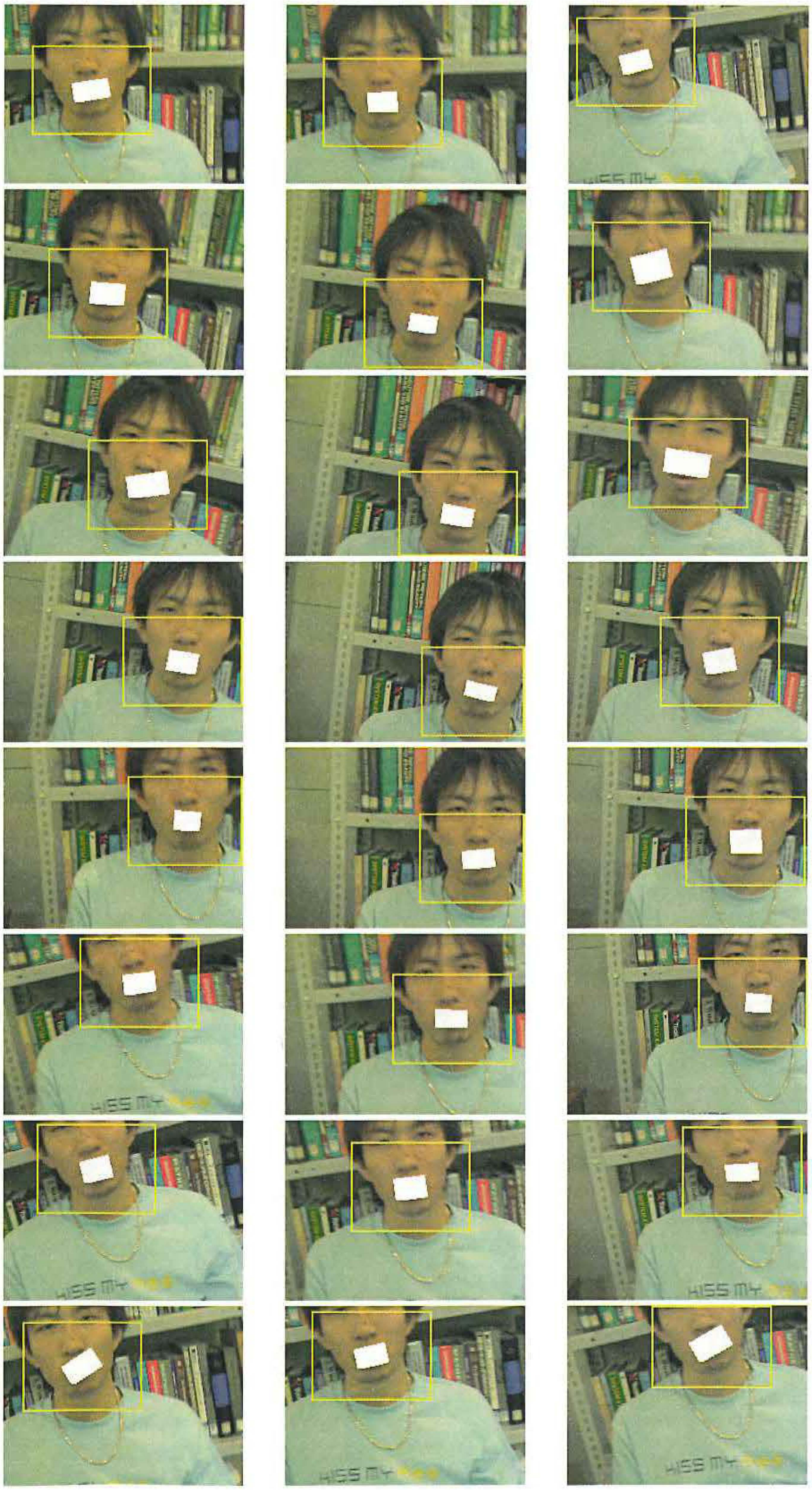


Figure 7.7 Target video frame examples (subject 1): see from top left to bottom right.

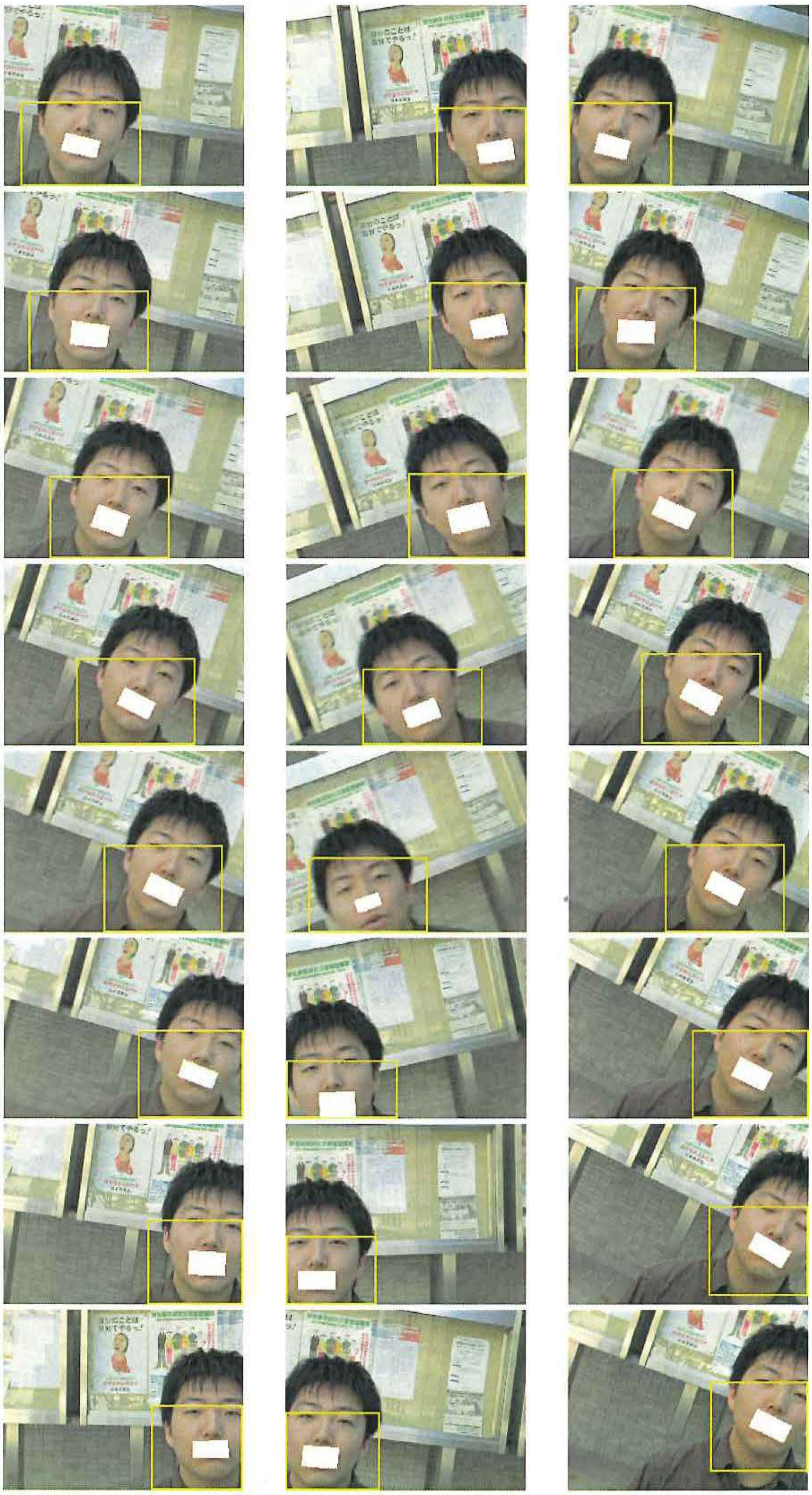


Figure 7.8 Target video frame examples (subject 2): see from top left to bottom right.

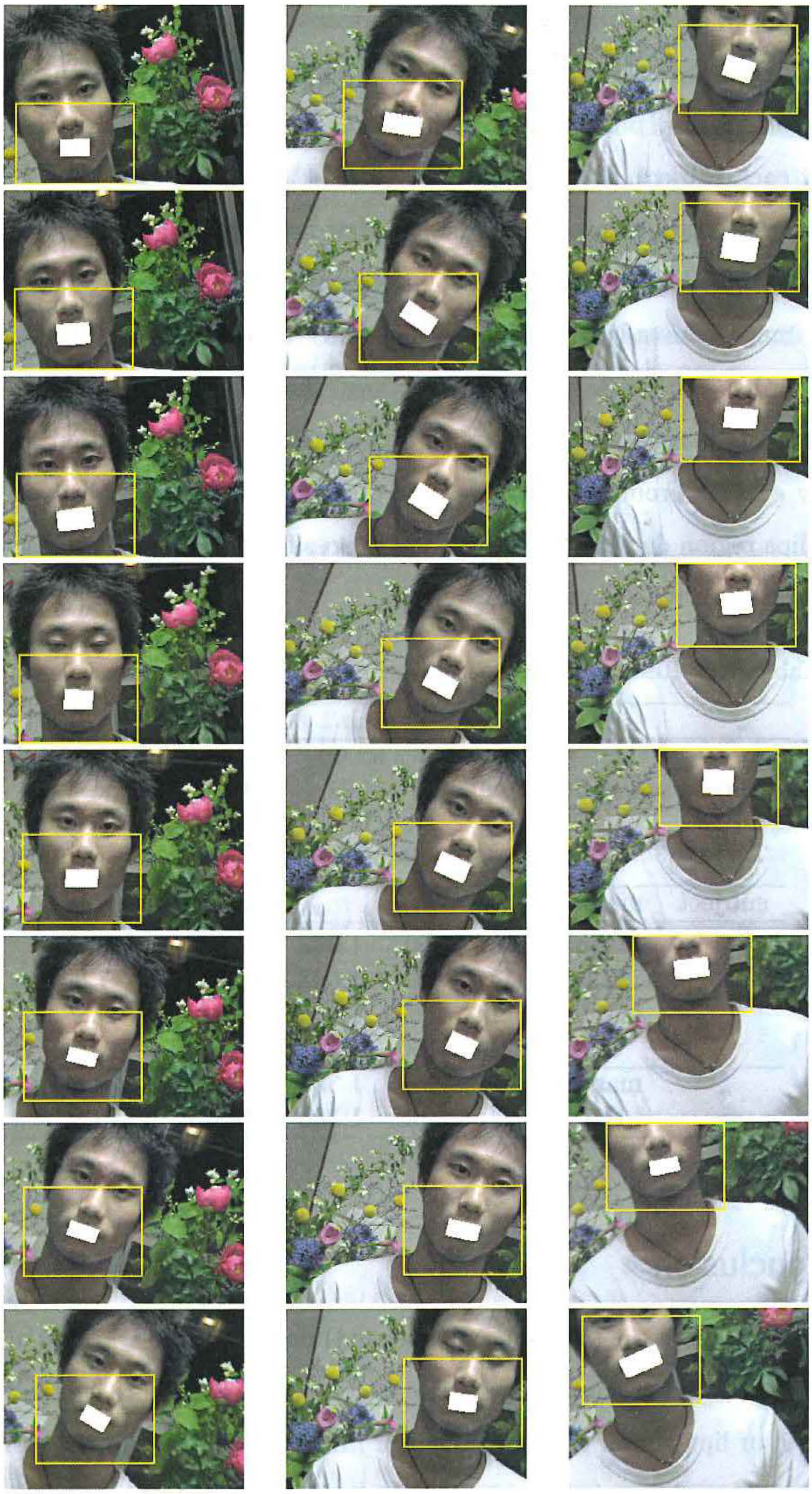


Figure 7.9 Target video frame examples (subject 3): see from top left to bottom right.

time. We judged success or failure by discriminant equation described in Section 3.3. From Table 7.2, it is shown that detection and tracking are carried out with high speed and a real-time processing. Furthermore, the detection and tracking accuracy is high for each subject.

Table 7.3 presents numerical results of the first frame of each subject in Figures 7.7-7.9. The row labelled "manual" is a solution, which is acquired by matching with a template image manually, and the row labelled "GAs" is a result of our proposed system. In this Table 7.3, t_x and t_y represent the coordinate, m_x and m_y are the scaling rate, and *angle* is the rotation angle. The manual solution and system's are nearly equal. From simulations, it is demonstrated that high speed and high accuracy lips region detection and tracking is possible to be done, with acquisition of its numerical geometric change information.

Table. 7.2 Results of simulation (accuracy and average processing time)

	sub.1	sub.2	sub.3	total
accuracy [%]	92.00	94.56	96.78	94.44
average processing time [sec]	4.61	4.44	4.45	4.50

Table. 7.3 Numerical results (comparison of manual with GAs)

subject	method	t_x	t_y	m_x	m_y	*angle* [deg]
1	manual	136	50	1.642	1.713	-11.0
	GAs	137	49	1.643	1.965	-7.8
2	manual	114	159	1.416	1.994	-25.7
	GAs	116	160	1.541	2.467	-29.2
3	manual	81	109	1.385	1.360	19.6
	GAs	80	108	1.392	1.455	18.8

7.5 Conclusion

In this paper, real-time (30 frames per second) detection and tracking of lips region of a talking person in natural scenes were presented. Furthermore we tried to acquire the numeric of lips region geometric change information. Our approach is based on image template matching with GAs. Moreover, this consists of some novel features of lips color and shape deformations, and improvements of GAs.

In our simulations, some failures occurred, because of the lips region frame-out. However, such things often happen, in natural scene. From simulation results, it is evaluated that our proposed system can continue to chase the lips region even in such a case. It is demonstrated that the lips region detection and tracking at high speed and with high accuracy is possible, with acquisition of its numerical geometric change information. This means that our proposed system can apply to robot perception and interface of mobile devices. Because, by using the geometric information, the lips region can be normalized as a visual front end of audio-visual speech recognition. Our future work is to simulate by more situations and to try these applications.

Chapter 8

Way Ahead...

8.1 Formidable Challenge

Image Understanding systems start by processing images to remove noise and irrelevant information and to enhance the relevant information, then they analyze the image with feature extraction techniques. However, there is few Image Understanding systems which are suitable for practical use. The reasons is that it is difficult to extract the relevant information to represent the object stably, supporting real-world. The object has complex changes by various causes. The complex changes are classified by the causes as follow.

1. changes of the scene by camera motion
2. changes of appearance by motion of itself
3. shape deformations of the object
4. changes of the color information by illumination.

This dissertation dealt with 1-3. Forth problem illumination is an important but formidable problem which is the same as others. The reason is that the color is important information for human to detect an object, and for Image Understanding too. If the object can be detected with invariance for all illumination condition, many applications will be achieved, such as robots in nuclear power stations, robotic planetary exploration, video-based security systems, and so on.

Chapter 9

Conclusions

The purpose of this work is object detection in an active scene. As a part of the objectives, we also try to acquire numerical information to represent the object.

Image Understanding is the process to understand the content of images in order to automate visual tasks by computers. The technical challenge of this study is to make the computer understand the contents of the images. In other words, the most difficult problem is to automatically produce a reasonable description from an image. It is clear that the nature of images and descriptions have a big distance. In the fields of Artificial Intelligence, Scene Analysis, Image Analysis, Image Processing, and Computer Vision, the many researchers work on reducing this distance in the last twenty years. However, there is few Image Understanding systems which are suitable for practical use. The reasons is that it is difficult to extract the relevant information to represent the object stably, supporting real-world. The object has complex changes by various causes. A new object detection and information extraction approach, which can be applied to these various changes, is necessary for Image Understanding.

In this dissertation, the object detection and the information extraction system is proposed. The object is lips region of a talking person. The camera is free to move and independent with the human.

The first chapter showed the methodology as basis of the proposed system in this dissertation, and described the choice of the best fitness function for this system. The best fitness function is the Dynamic Fitness Function. The other chapters dealt with this Dynamic Fitness Function.

This system had some problems. The first is this system was unsuitable for practical use. The reasons is that this system deals with two-dimensional changes only. In order to support the real-world, we must develop a system to three-dimensional space. This development was described in Chapters 4 and 5.

The second problem is the search speed. GA is a probabilistic search technique which is suitable for the exploration of large and complex search spaces. Typically, the GA has a trade-off between exploration accuracy and speed. In other words, to obtain high accuracy, the size of the population and the number of generations must be increased. Therefore, high speed exploration with keeping high accuracy—downsizing of GA is necessary. In Cahpter 6, we proposed the downsized GA with the Search Domain Control method.

The proposed methods which we mentioned above, dealt with a single image. In order to achieve real-time video processing and use previous methods efficiently, the evolutionary video processing was proposed in Chapter 7. The main techniques of this system are the flexible control of search domain and the inheritance of genetic information between video frames. The simulation results indicated that high speed and high accurate object detection and information extraction was achieved by using this method.

The research result which are mentioned above is very useful in many situations where the picture understanding system is necessary, as follows:

Dangerous situation Tasks which is too dangerous for human.

Examples are: robots in nuclear power stations, robotic planetary exploration.

Sensitive situation Tasks which suffer if the human fatigues, and which are prone to this problem.

Examples are: industrial inspection, video-based security systems.

Economical situation Tasks which require specialized training, resulting in human resources that are rare and costly.

Example are: Medical screening for tumors, intelligence gathering from satellite imagery.

Strict situation Tasks which humans do poorly because visual items need to be measured accurately.

Example are: progress of disease, efficacy of medication, growth of cracks in weldments, number of specific cells in a microscope slide.

Humanly impossible situation Tasks which have too much data for effective application of humans.

Examples are: counting the potholes in highways, inspection of every bottle in a bottling plant, keeping up with intelligence data during wartime.

The Genetic and Evolutionary Computation (GEC) is the generic name for GA, Genetic programming (GP) that is the extension of GA, Evolutionary Strategy (ES), and Evolutionary Programming (EP) [22]. Recently, GECs have gained a growing popularity and a fairly great number of attempts to use GECs to solve complex problems in various application fields [22]. In this work, it is shown that GECs can be applied to the complex problem as mentioned above, which is object detection in an active scene for Image Understanding. I will pursue in future research about Image Understanding technology, such as detection, tracking, and information acquisition of not only lips region but also other objects.

Bibliography

[1] Harry McGurk and John MacDonald. Hearing lips and seeing voices. *Nature*, Vol. 264, No. 5588, pp. 746–748, December 1976.

[2] Sabine Windmann. Effects of sentence context and expectation on the mcgurk illusion. *Journal of Memory and Language*, Vol. 50, No. 2, pp. 212–230, February 2004.

[3] Ingrid R. Olson, J. Christopher Gatenby, and John C. Gore. A comparison of bound and unbound audio-visual information processing in the human cerebral cortex. *Cognitive Brain Research*, Vol. 14, No. 1, pp. 129–138, June 2002.

[4] Marc Liévin, Patrice Delmas, Pierre Yves Coulon, Franck Luthon, and Vincent Fristot. Automatic lip tracking: Bayesian segmentation and active contours in a cooperative scheme. In *Proceedings IEEE International Conference on Multimedia Computer and Systems (ICMCS '99)*, Vol. 1, pp. 691–696, Firenze, Italy, June 1999.

[5] Trent W. Lewis and David M. W. Powers. Lip feature extraction using red exclusion. In Peter Eades and Jesse Jin, editors, *Selected papers from Pan-Sydney Workshop on Visual Information Processing*, Vol. 2 of *Conferences in Research and Practice in Information Technology*, pp. 61–67, Sydney, Australia, October 2001. Australian Computer Society Inc.

[6] Juergen Luettin, Neil A. Thacker, and Steve W. Beet. Visual speech recognition using active shape models and hidden markov models. In *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '96)*, Vol. 2, pp. 817–820, Atlanta, USA, May 1996.

[7] Lionel Revéret. From raw image of the lips to articulatory parameters: A viseme-based prediction. In *Proceedings of the Fifth biennial European Conference on*

Speech Communication and Technology (Eurospeech '97), pp. 2011–2014, Rhodes, Greece, September 1997.

[8] Renaud Séguier and Nicolas Cladel. Multiobjectives genetic snakes application on audio-visual speech recognition. In *Proceedings of the Fourth EURASIP Conference focused on Video / Image Processing and Multimedia Communications (EC-VIP-MC 2003)*, pp. 625–630, Zagreb, Croatia, July 2003.

[9] Christoph Bregler and Yochai Konig. "eigenlips" for robust speech recognition. In *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP-94)*, Vol. ii, pp. II/669–II/672, Adelaide, Australia, April 1994.

[10] Andreas Birk and Holger Kenn. Roboguard, a teleoperated mobile security robot. *Control Engineering Practice*, Vol. 10, No. 11, pp. 1259–1264, November 2002.

[11] William E. Green and Paul Y. Oh. An aerial robot prototype for situational awareness in closed quarters. In *Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)*, Vol. 1, pp. 61–66, Las Vegas, USA, October 2003.

[12] Jean-Christophe Zufferey, Dario Floreano, Matthijs van Leeuwen, and Tancredi Merenda. Evolving vision-based flying robots. In *Proceedings of the Second International Workshop on Biologically Motivated Computer Vision*, pp. 592–600, Tüingen, Germany, November 2002.

[13] T. J. Wark and Sridha Sridharan. Adaptive fusion of speech and lip information for robust speaker dentification. *Digital Signal Processing*, Vol. 11, No. 3, pp. 169–186, July 2001.

[14] Juergen Luettin, Neil A. Thacker, and Steve W. Beet. Speaker identification by lipreading. In *Proceedings of the Fourth International Conference on Spoken Language Processing (ICSLP'96)*, Vol. 1, pp. 62–65, Philadelphia, USA, October 1996.

[15] Ulrich Kremer, Jamey Hicks, and James M. Rehg. A compilation framework for power and energy management on mobile computers. In *Proceedings of the Fourteenth International Workshop on Parallel Computing (LCPC2001)*, pp. 115–131,

Cumberland Falls, KY, USA, August 2001.

[16] Linda L. Otis, Daquing Piao, Carolyn W. Gibson, and Quing Zhu. Quantifying labial blood flow using optical doppler tomography. *Oral Surgery, Oral Medicine, Oral Pathology, Oral Radiology, and Endodontology*, Vol. 98, No. 2, pp. 189–194, August 2004.

[17] Jr Hugh M. Gloster. The use of second-intention healing for partial-thickness Mohs defects involving the vermilion and/or mucosal surfaces of the lip. *Journal of the American Academy of Dermatology*, Vol. 47, No. 6, pp. 893–897, December 2002.

[18] Sarah Weitzul, Sarah Weitzul, and Sarah Weitzul. Lip reconstruction. eMedicine Journal [serial online], 2002.

[19] Babak Jahan-Parwar, Keith Blackwell, and Babak Jahan-Parwar. Lips and perioral region anatomy. eMedicine Journal [serial online], 2002.

[20] Takuya Akashi, Minoru Fukumi, and Norio Akamatsu. Accuracy and speed improvement in lips region extraction. In *Proceedings of the Eighth Australian and New Zealand Conference on Intelligent Information Systems (ANZIIS 2003)*, pp. 495–500, Australia, December 2003.

[21] David E. Goldberg. *Genetic Algorithms in search optimization & Machine lerningn.* Addison-Wesley Publishing Campany, Inc., Boston, MA, USA, 1989.

[22] Hisashi Shimodaira. Strategies for optimizing image processing by genetic and evolutionary computation. In *Proceedings of the twelfth IEEE International Conference on Tools with Artificial Intelligence (ICTAI 00)*, pp. 151–154, Vancouver, BC, Canada, 2000.

[23] Cheng-Chin Chiang, Wen-Kai Tai, Mau-Tsuen Yang, Yi-Ting Huang, and Chi-Jaung Huang. A novel method for detecting lips, eyes and faces in real time. *Real-Time Imaging*, Vol. 9, No. 4, pp. 277–287, August 2003.

[24] Koji Iwano, Satoshi Tamura, and Sadaoki Furui. Bimodal speech recognition using lip movement measured by optical-flow analysis. In *Proceedings of the First ISCA/IEEE/ASJ/IEICE International Workshop on Hands-Free Speech Communication (HSC2001)*, pp. 187–190, Kyoto, Japan, April 2001.

[25] Lucia Ballerini. Genetic snakes for medical images segmentation. In *Proceedings of SPIE Mathematical Modeling and Estimation Techniques in Computer Vision*, Vol. 3457, pp. 284–295, San Diego, USA, September 1998.

[26] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active shape models. *International Journal of Computer Vision*, Vol. 1, pp. 321–331, 1988.

[27] Kenji Mase. Recognition of facial expression from optical flow. *IEICE Transactions*, Vol. E-74, No. 10, pp. 3474–3483, October 1991.

[28] Irfan A. Essa and Alex P. Pentland. Facial expression recognition using a dynamic model and motion energy. In *Proceedings the Fifth International Conference on Computer Vision (ICCV 95)*, pp. 360–367, Cambridge, USA, June 1995.

[29] David Cristinacce and Timothy F. Cootes. Facial feature detection using ADABOOST with shape constraints. In *Proceedings of the Fourteenth British Machine Vision Conference*, pp. 231–240, Norwich, UK, Sep. 2003.

[30] Ltd. Minolta Co. Story of color knowledge, 2002.

[31] Konstantinos N. Plataniotis and Anastasios N. Venetsanopoulos. *Color Image Processing and Applications.* Springer-Verlag, Berlin, Germany, 2000.

[32] Japanease Standards Associstion. *JIS Hand Book Vol.61 Colour.* Japanease Standards Associstion, Tokyo, Japan, 2002.

[33] F. Yamaguchi. *Graphics Pcocessing and Geometric Modeling by Four-dimensional Theory.* NIKKAN KOGYO SHIMBUN, Tokyo, Japan, 1996.

[34] Felix Klein. *Erlangen program.* Inaugural address at the University of Erlangen, 1872.

[35] H. Ida. *Genetic Algorithms.* IGAKU-SHUPPAN, Tokyo, Japan, 2002.

[36] John H. Holland. *Adaptation in natural and artificial systems.* The University of Michigan Press, Michigan, USA, 1975.

[37] Thomas Bäck, David Fogel, and Zbigniew Michalewicz. *Handbook of Evolutionary Computation.* Institute of Physics Publishing Ltd., Bristol, UK, 1998.

[38] Whitley D. and Kauth J. Genitor: a different genetic algorithm. In *Proceedings of Rocky Mountain Conference on Artificial Intelligence*, pp. 118–130, Colorado, USA, 1988.

[39] Whitley D. The genitor algorithm and selective pressure: why rank-based allocation of reproductive trials is best. In *Proc. 3rd Int. Conf. on Genetic Algorithms*, pp. 116–121, Virginia, USA, June 1989.

[40] F. Gray. (1953). pulse code communication in 1953, march. U. S. Patent 2 632 058.

[41] R. B. Hollstien. Artificial genetic adaptation in computer control systems. In *PhD thesis, University of Michigan*, Michigan, USA, 1971.

[42] NetRatings Japan Inc. (2001). internet population survey in 2001, march. <http://www.netratings.co.jp/press_releases/pr_190401.html/> [2001, May 24].

[43] H. Kashiwazaki and S. Hirose. Extraction of issue for spoken language learning -using perceptual decision of spoken language and sound spectrograph-. In *The 17th Annual Meeting of the Japanese Cognitive Science Society*, pp. 1–35, Japan, June 2000.

[44] Richard A. Caruana and J. David Schaffer. Representation and hidden bias: Gray vs. binary coding for genetic algorithms. In *Fifth International Conference on Machine Learning*, pp. 153–161, San Mateo, USA, 1988.

[45] Jorge Stolfi. Oriented projective geometry. In *Proceedings of the Third ACM Annual Symposium on Computational Geometry (SOCG'87)*, pp. 76–85, Waterloo, Canada, June 1987.

[46] Takuya Akashi, Yasue Mitsukura, Minoru Fukumi, and Norio Akamatsu. Genetic lips extraction method for varying shape. *IEEJ Transaction on Electronics, Information and Systems*, Vol. 124, No. 1, pp. 495–500, January 2004.

[47] Darrell Whitley. The GENITOR algorithm and selection pressure: Why rank-based allocation of reproductive trials is best. In *Proceedings of the Third International Conference on Genetic Algorithms (ICGA'89)*, pp. 116–121, Fairfax, Virginia, USA, 1989.

[48] James E. Baker. Adaptive selection methods for genetic algorithms. In *Proceedings of the First International Conference on Genetic Algorithms and their applications*, pp. 101–111, Hillsdale, New Jersey, USA, 1985.

[49] John J. Grefenstette. GENESIS: a system for using genetic search procedures. In *Proceedings of Conference on Intelligent Systems and Machines*, pp. 161–165, Rochester, MI, 1984.

[50] Gilbert Syswerdar. Uniform crossover in genetic algorithms. In *Proceedings of the Third International Conference on Genetic Algorithms (ICGA89)*, pp. 2–9, an Mateo, CA, USA, August 1989.

[51] Takuya Akashi, Minoru Fukumi, and Norio Akamatsu. Invariant lips extraction for variation of horizontal direction. In *The Sixth IASTED International Conference on Signal and Image Processing (SIP2004)*, pp. 58–63, Hwaii, USA, August 2004.

[52] Yong Fan, Tianzi Jiang, and David J. Evans. Volumetric segmentation of brain images using parallel genetic algorithms. *IEEE Transactions on Medical Imaging*, Vol. 21, No. 8, pp. 904–909, 2002.

[53] Hiroshi G. Okuno, Kazuhiro Nakadai, and Hiroaki Kitano. Social interaction of humanoid robot through auditory and visual tracking. In *Proceedings of the Eighth International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA/AIE-2002)*, Vol. 2358 of *Lecture Notes in Artificial Intelligence*, pp. 725–735, June 2002.