Study on Model of Object Recognition Using Neuro-Oscillator and Curvature Based Shape Representation in the Visual Cortex

Doctoral Thesis

Thesis Title

大脳視覚情報処理に基づく神経振動と曲率表現を用いた物体認識のモデル化に関する研究

Study on Model of Object Recognition Using Neuro-Oscillator and Curvature Based Shape Representation in the Visual Cortex

Department of Electronic Engineering

Graduate School of Engineering

Tohoku University

Yuma MATSUDA

ID No. B4TD9209

Study on Model of Object Recognition Using Neuro-Oscillator and Curvature Based Shape Representation in the Visual Cortex

ABSTRACT :

This study proposes models to solve issues on the processes of “perceptual grouping”, “shape representation”, and “shape mapping” in order to make it clear how the human visual system recognizes object in real-world scenes. In order to recognize which elements are grouped as those of the same object shape (perceptual grouping), an ill-posed problem is required to be solved. In order to invariantly describe shapes, a shape representation of the human visual system is required to be understood. In order to retrieve shapes even in unpredictably and dynamically changing real-world scenes, a shape retrieval method using a suitable shape mapping process is required to be unveiled. These solutions are proposed as mechanisms which are plausibly working in the human visual system.

Chapter 2 proposes the solution for the first issue “perceptual grouping”, which is challenging for computational visual systems. In order to perceive groups in images, the connections between each two pixels and the groups as the result of these connections is required to be simultaneously found. Thus, the connections and the groups are required to be individually described, and dynamically and interactively formed. Various types of computational neural oscillator network models have previously been proposed to perceive groups. Although previous models dynamically solve the grouping with short-range excitation and long-range inhibition, both the groups and the interaction between the groups and the connections are not described. Therefore, it is not described what shapes of groups are located where, and the balance of the short-range excitation and the long-range inhibition, in principle, cannot be adjusted to the groups included in images. For this reason, we propose a model that describes the connections and the groups individually on different “layers”, and forms them dynamically and interactively. In order to describe the connections and the groups, we model the dynamics of the inter-element interaction (for the connections), the inter-layer interaction, and the inter-variable interaction (for color variables). As a result of this modeling, we show that our model successfully forms groups and segregates them into spatial regions in various color images.

Chapter 3 proposes the solution for the second issue “shape representation”, which is also challenging. The human visual system recognizes general shapes flexibly in real-world scenes through the ventral pathway. The pathway is typically modeled using artificial neural networks. These network models, however, do not construct a shape representation that satisfies the following required constraints: (1) The original shape should be represented by a group of partitioned contours in order to retrieve the whole shape (global information) from the partial contours (local information). (2) Coarse and fine structures of the original shapes should be individually represented in order for the visual system to respond to shapes as quickly as possible based on the least number of their features, and to discriminate between

shapes based on detailed information. (3) The shape recognition realized with an artificial visual system should be invariant to geometric transformation such as expansion, rotation, or shear. We, therefore, propose a visual shape representation with geometrically characterized contour partitions described on multiple spatial scales.

Chapter 4 proposes the solution for the third issue “shape mapping”, which is still challenging even after both of “perceptual grouping” and “shape representation” are understood. The human visual system retrieves shapes from incomplete information in the real world, and it has inspired a lot of computational methods of retrieving shapes. In order to retrieve shapes, the observed shapes are decided to be alike or unlike remembered shapes in memory after the comparison of these shapes. To compare the observed and remembered shapes, they must first be appropriately represented so that the points on each shape can be mapped and compared. For this reason, the shape retrieval process needs appropriate shape representation and shape mapping methods. Moreover, the shape representations should be normalized before the mapping process. However, a normalization process for representations under unpredictable conditions has not yet been established. We, therefore, describe a shape retrieval method that enables us to retrieve shapes under unpredictable conditions with a suitable normalization process. Using the shape representation “curvature partition” and the shape normalization method “angle-length profile”, our shape retrieval method normalizes the shape representation before it does the mapping. As a result, unlike the previously proposed methods, it can be used under unpredictable conditions such as when occlusion, geometric distortion, and differences in image resolution occur simultaneously.

Throughout this study, the processes of “perceptual grouping”, “shape representation” and “shape mapping” are well understood from the view of both biological and computational visual system. Chapters 5 and 6, the computational visual system “One Shot Detector (OSD)” is finally proposed as a combination of these processes. This system is proposed as a promising new system of recognizing object without using machine learning schemes. For object recognition in real world scenes, reliability is a quite challenging accomplishment since real world environments change unpredictably and dynamically. In these environments, objects are also observed as unpredictably and dynamically changed shapes. For recognizing these objects, a huge number of changed shapes (sample shapes) should be learned beforehand. However, sample shapes cannot always be obtained since they are unpredictably and dynamically changed. On the other hand, our OSD is shown to achieve great performance using examples of side-scan sonar images in maritime environments (Chapter 5) and satellite images (Chapter 6). We, therefore, introduce our object recognition system based on OSD with great performance. Then, we finally conclude our study in the Chapter 7.

Contents

Chapter 1

Introduction

The real world is an unpredictably and dynamically changing uncertain environment. Although the human visual system always encounters these uncertainty, it has not yet been clarified how the human visual system recognizes object in these real-world scenes. For this reason, it is still a challenging issue for visual systems to identify where objects are in real-world scenes.

Computational visual systems describe the real world based on images that consist of pixels. In these images, an object is recognized as a group of pixels. However, one of the biggest questions in this process is how the visual systems recognize which pixels belonging to which objects. Since images are matrices of pixels, these images do not have any "cues" to recognize objects. Therefore, it is generally challenging to recognize objects included in images.

When our brain sees an object with our eyes, the image is projected onto the retina. Then, this image is relayed through the lateral geniculate nucleus (LGN) to the visual cortex. The visual cortex processes the information input from the retina with the neurons' receptive fields (RFs). Lower visual cortex, using smaller RFs, extracts simple features such as the orientation, color, disparity, motion of objects included in the image [1, 2]. As a feature of object shapes, primary visual cortex (area V1) has orientation-selective neurons, which extract the orientation of contours of the object shapes [3]. In higher visual cortex (area V4), it is known that more complex features are sensed [4, 5]. In this area, a set of the orientation information is recognized as a set of tangential lines to a circle [6, 7]. Then, curvature information as a radius of a circle is recognized. This mechanism is verified with representing neurons in area V4 which react with curvature and location included in contours of shapes. Then, it is assumed that the curvature information, which contains more complex information than simple local lines represented in area V1, can represent local and global features of shapes simultaneously. In area posterior inferotemporal cortex (PIT), which is next to area V4, some of these curvatures are observed to be jointed as a description of a closed contour, which is located next to area V4 [8, 9]. The brain's visual area is assumed to apply these sets of curvature information for invariant representations of objects since the inferotemporal cortex including PIT is responsible for the shape recognition [10]. It is, then, also assumed that the sets of curvature information are invariantly represented in area PIT since the neurons in this area are observed to react invariantly with geometrically deformed shapes.

The important fact is that the human visual system accomplishes "figure-ground separation" through this process. The visual system recognizes which elements of images belonging to which objects, and separates them from other elements of the images. This figure-ground separation is done even under the condition that any "cues" for recognizing objects is included in the images. Although it is well-known

that figure-ground separation occurs through the process of the human visual system, little is known on the mechanism of this figure-ground separation. We assume that each process of "perceptual grouping", "shape representation", and "shape mapping" has issues to be solved in order for us to understand how the human visual system recognizes objects.

1. **PERCEPTUAL GROUPING**
   In order to recognize which elements are grouped as those of the same object shape (perceptual grouping), an ill-posed problem is required to be solved [11]. Because an image is originally described as a set of pixels, each of which has different color, it is not pre-determined which pixels belong to the same group. It is unknown which pixels should be connected to form the same groups unless it is found what shapes of groups are located where. In the same way, it is unknown what shapes of groups are located where unless it is found which pixels are connected or disconnected. In order to solve this ill-posed problem and achieve the perceptual grouping, we assume that three factors are required to be unveiled.
2. **SHAPE REPRESENTATION**
   Unless the shapes are described as whole representations, computational visual systems cannot basically extract and identify them. The human visual system is, on the other hand, assumed to identify shapes flexibly using a suitable shape representation. For this reason, the shape representation described in the human visual system needs to be unveiled. The conventional shape representations suitable for shape retrieval, however, have not yet been constructed since none of them fulfills all of the constraints that are required for shape matching:
   (a) The original shape should be represented with a group of partitioned contours in order to retrieve the whole shape (global information) from the partial contour (local information) [12–15].
   (b) Coarse and fine structures of the original shapes should be individually represented in order for the visual system to respond to shapes as quick as possible with least number of features of them, and to discriminate a shape from others with the detailed information [16–19].
   (c) The shape recognition realized with the visual system should be invariant to geometric transformation such as rotation or shear [20, 21].
   Therefore, a biologically plausible shape representation which is suitable for shape retrieval is required to be proposed.
3. **SHAPE MAPPING**
   Even if shapes are described with suitable representations, there still remain a large issue to be solved. Complete shapes must be retrieved from incompletely described shapes affected by occlusion, geometric distortion, and differences in image resolution. Such a retrieval will be successful if the shape representations appropriately describe the incompletely observed shapes and if the observed shapes are appropriately mapped to shapes in memory (the point correspondences are appropriately solved). Thereby, the incompletely observed shapes can be appropriately matched to shapes in memory and it can be decided if they are alike or unlike. For this reason, the task of shape retrieval will require appropriate shape-representation and shape-mapping methods. It is assumed that the shape retrieval methods should appropriately normalize the shape representations in the mapping process even under the condition of occlusion, geometric distortion, and differences in image resolution.

In this paper, we propose models to solve issues on the processes of "perceptual

grouping", "shape representation", and "shape mapping". We, then, propose a system to accomplish "figure-ground separation" using these models. The issue on perceptual grouping is solved by a model of the perceptual grouping in color images with the inter-element interaction, the inter-layer interaction (the interaction between the globally formed groups and locally formed connections), and the inter-variable interaction (the interaction among RGB values). The issue on shape representation is, then, solved by a shape representation "called curvature partition". The curvature partition, which is composed of geometrically characterized contours partitioned by zero-crossing points on the contours with coarse and fine shapes, describes general shapes well. Finally, The issue on shape mapping is solved by a shape mapping process using a normalization using an "angle-length profile". By exploiting the fact that the angle variation is Euclidean invariant even under occlusions, we construct an angle-length profile as a basis for the normalization of the curvature partition. As a result, unlike the previously proposed methods, our method can be applied in situations with occlusion, geometric distortion, and differences in image resolution. Chapter 2 proposes a model of accomplishing perceptual grouping (image segmentation) with the title of "Model of perceptual grouping with inter-element, inter-layer and inter-variable interactions". Chapter 3 proposes a shape representation called "curvature partition" with the title of "Visual shape representation with geometrically characterized contour partitions". This chapter is based on an author's work [22]. Chapter 4 proposes a method of shape mapping and retrieval with the title of "Method of retrieving general shapes with occlusion using geometrically characterized contour partitions". This chapter is based on an author's work [23]. Then, Chapter 6 and Chapter 5 present the evaluation results that our proposed model of perceptual grouping, shape representation, and shape mapping method are applied for recognizing object in real-world scenes. Chapter 6 deals with objects on satellite images, and Chapter 5 deals with objects on side scan sonar images. Finally, Chapter 7 presents a discussion and concludes this paper.

Chapter 2

Image Segmentation Model of perceptual grouping with inter-element, inter-layer and inter-variable interactions

2.1 Introduction

It is still a challenging issue for computational visual systems to identify where objects are in real-world scenes. Computational visual systems describe the real world based on images that consist of pixels. In these images, an object is recognized as a group of pixels. However, one of the biggest questions in this process is how the pixels that are assumed to be the components of an object can be separated from the other pixels in the images. In other words, how does "figure-ground separation" occur? The human retinal system describes real-world scenes based on the neuronal firing patterns as if the neurons work as the pixels in the computational images. These patterns are transmitted to the visual cortex. In this process, a "group" of firing neurons, which are activated synchronously, describes a certain object or a spatial region [24, 25]. As a result of the perceptual grouping mechanism, it is assumed that the human visual system finds the local features belonging to the same objects (border-ownership) based on only a glance [26–28].

In order to achieve the perceptual grouping, an ill-posed problem is required to be solved [11]. Because an image is originally described as a set of pixels, each of which has different color, it is not pre-determined which pixels belong to the same group. It is unknown which pixels should be connected to form the same groups unless it is found what shapes of groups are located where. In the same way, it is unknown what shapes of groups are located where unless it is found which pixels are connected or disconnected. In order to solve this ill-posed problem and achieve the perceptual grouping, we assume that three factors are required to be unveiled.

1. **Dynamics of the inter-element interaction**: All the elements of images (pixels) are uniquely described, and each element has its own RGB values. Therefore, it is not pre-determined if arbitrary selected neighboring two elements are connected or disconnected. The connection between two elements is required to be determined in consideration of those against other peripheral elements, and vice versa. For this reason, the mechanisms to dynamically

and interactively determine the connecting condition between two elements by reflecting those against other peripheral elements are required to be unveiled.

2. **Dynamics of the inter-layer interaction**: Assuming that the perceptual groups and elements are described on different layers, the states of these layers are required to be dynamically and interactively determined since these states cannot be preliminary determined. The perceptual groups on "higher layer" are formed through the bottom-up connections, and the local connections on "lower layer" are explicitly determined with the states of the perceptual groups. Globally formed groups are the final compilation of the locally formed connections. The locally formed connections are, at the same time, strongly influenced by the globally formed groups. For this reason, the mechanisms to dynamically and interactively determine the states of the two layers are required to be unveiled.
3. **Dynamics of the inter-variable interaction**: The human visual system handles multiple variables such as motion vectors, border patterns, colors, or the saliency with neural oscillations so that it perceives groups (surfaces). In this context, the human visual system plausibly makes use of the inter-variable interaction when it perceives surfaces. The mechanisms for integrating the dynamics of the elements corresponding to the different parameters (variables) are required to be unveiled.

Historically, Various types of neural oscillator models have previously been proposed in order to solve the ill-posed problem for grouping [29, 30]. The previous models are roughly classified into four categories: binocular stereopsis models, figure-ground segregation caused by relative motion models, border patterns of the surfaces, or stereo still images (gray-level and color images). The binocular stereopsis models help explain the mechanism for the perception of surfaces when the binocular images are combined especially when using random-dot stereograms [31, 32]. The figure-ground segregation models explain the mechanism for the perceptual grouping of elements in images and the segregation of these groups from the background groups using the cues of relative motion [33, 34], the border patterns of the surfaces [13, 30, 35, 36], the gray-level images [37–40], or color images [41]. Although various oscillator models are proposed for the perceptual grouping, the ill-posed problem especially on stereo still images has not yet successfully been solved.

Typically, Wang's neural oscillator network model describes each element of images with a non-linear oscillator [37, 42, 43]. With his model, the elements belonging to the same groups are determined as those which have phase synchronized with each other. Although the connection between two elements is pre-determined with the differences of the luminances of the two elements, the relationship between two elements can dynamically and interactively change since the phase of each element is determined in consideration of those of its adjacent elements. For this reason, the first factor for solving the ill-posed problem (dynamics of the inter-element interaction) is considered. In addition, his model apply an oscillator on "higher layer" to cause long-range inhibitory connections. With this inhibition, groups can be clearly formed. However, the higher layer that we point out requires not only to cause long-range inhibitory connections but also to describe each formed group. Without the descriptions of groups and the dynamics of the interaction between groups and elements, the ill-posed problem cannot be solved (it is unknown what shapes of groups are located where). For this reason, the dynamics of the inter-layer interaction is required to be unveiled. Similarly, the dynamics of the inter-variable interaction has not yet been solved. In order to perceive groups from color images, Quiles et al. pro-

pose a computational visual system with a saliency map [44, 45] and Wang's neural oscillator network [41]. Because the saliency is closely related to the visual perception, the dynamics of the oscillators can reflect results of "perception". Also, the saliency map plays a role to reduce the parameter dimension of elements. However, the relationships among variables of elements are so complex that it is hard for the saliency map to comprehend them. These relationships can change by the context of inter-element relationships or inter-layer relationships. For this reason, the dynamics of the inter-variable interaction is required to be unveiled to solve the ill-posed problem.

In this chapter, we propose a model of the perceptual grouping in color images with the inter-element interaction, the inter-layer interaction (the interaction between the globally formed groups and locally formed connections), and the inter-variable interaction (the interaction among RGB values) to solve the ill-posed problem. Our proposed model consists of two hierarchical layers of neural oscillator networks and their interactions [31, 46, 47].

1. Hierarchical layers:
   (a) (higher layer) dynamically formed group
   (b) (lower layer) oscillators with dynamically changing local connection strength
2. Their interactions:
   (a) group (higher layer) formed by local connection (lower layer)
   (b) local connection strength influenced by both group (higher layer) and peripheral connection (lower layer)

The higher layer is responsible for forming groups according to the local connections formed on the lower layers. The lower layer, which consists of locally interacted neural oscillators, is responsible for forming the local connections modified by both the groups in the higher layer and the peripheral connections in the lower layer. The groups formed by our proposed model are the results in the cues for the figure-ground segregation in real-world scenes. As a result, we show that our model can perceive not only the color surface but also the gradation on the surface. Section 2.2 describes our approach. Section 2.3 presents the experimental results. Section 2.4 is a discussion based on these results and concludes this chapter.

2.2 Model

Our model consists of two hierarchical layers (Fig. 2.1). The first layer, the "lower layer", represents the RGB values of each pixel in an image using the firing patterns of oscillatory neural elements. On this layer, each of the RGB values of a pixel are explained using each respective oscillator. These oscillators communicate with their neighbors, and adjust the connection strength among them according to the similarities in the RGB values. Strongly connected oscillators are, then, grouped together. These formed groups are, on the second "higher layer", explained using the average RGB values. The connection strengths among the oscillators are adjusted again based on these averaged RGB values explained on the higher layer.

The lower layer, which describes an image, is an $N \times M$ array of sets of nonlinear oscillators (Fig. 2.2). On this layer, a pixel in the image at position (i, j) is described by focusing on the activities of a set of oscillators such as $\left(x^R_{i,j}(t), x^G_{i,j}(t), x^B_{i,j}(t)\right)$, which correspond to the R, G, and B values of the pixel, respectively. These activities are the functions of time t. In order to flexibly reform the relationship toward the neighboring oscillators, the KYS oscillator [31, 48], which is a modified van der Pol oscillator, is applied. The dynamics of the oscillators that describe the pixel at position (i, j) are given by

$$\begin{aligned} d^2x^R_{i,j}/dt^2 + \left(A_1 {x^R_{i,j}}^2 + B_1 x^R_{i,j} + C_1\right) dx^R_{i,j}/dt \\ + \left(A_2 {x^R_{i,j}}^2 + B_2 x^R_{i,j} + C_2\right) x^R_{i,j} = D^R_{i,j}, \end{aligned} \tag{2.1a}$$

$$\begin{aligned} d^2x^G_{i,j}/dt^2 + \left(A_1 {x^G_{i,j}}^2 + B_1 x^G_{i,j} + C_1\right) dx^G_{i,j}/dt \\ + \left(A_2 {x^G_{i,j}}^2 + B_2 x^G_{i,j} + C_2\right) x^G_{i,j} = D^G_{i,j}, \end{aligned} \tag{2.1b}$$

$$\begin{aligned} d^2x^B_{i,j}/dt^2 + \left(A_1 {x^B_{i,j}}^2 + B_1 x^B_{i,j} + C_1\right) dx^B_{i,j}/dt \\ + \left(A_2 {x^B_{i,j}}^2 + B_2 x^B_{i,j} + C_2\right) x^B_{i,j} = D^B_{i,j}, \end{aligned} \tag{2.1c}$$

where $D^R_{i,j}$, $D^G_{i,j}$, and $D^B_{i,j}$ are the inputs for each of the oscillators, and $x^R_{i,j}(t)$, $x^G_{i,j}(t)$, and $x^B_{i,j}(t)$ are the activities of the oscillators. These inputs are then given by

$$D^R_{i,j} = {D_0}^R_{i,j} + \partial D^R_{i,j}, \tag{2.2a}$$

$$D^G_{i,j} = {D_0}^G_{i,j} + \partial D^G_{i,j}, \tag{2.2b}$$

$$D^B_{i,j} = {D_0}^B_{i,j} + \partial D^B_{i,j}, \tag{2.2c}$$

where ${D_0}^R_{i,j}$, ${D_0}^G_{i,j}$, and ${D_0}^B_{i,j}$ are the fundamental inputs for determining the fundamental frequencies according to the RGB values, and $\partial D^R_{i,j}$, $\partial D^G_{i,j}$, and $\partial D^B_{i,j}$ are the inputs from the neighbors. Each of ${D_0}^R_{i,j}$, ${D_0}^G_{i,j}$, and ${D_0}^B_{i,j}$ corresponds to the brightness for each wavelength as pointed out by Land [49–53]. The fundamental inputs are given by

$${D_0}^R_{i,j} = \rho + \lambda r_{i,j}, \tag{2.3a}$$

$${D_0}^G_{i,j} = \rho + \lambda g_{i,j}, \tag{2.3b}$$

$$D_{0}{}^{B}_{i,j} = \rho + \lambda b_{i,j}, \tag{2.3c}$$

where ρ and λ are the constants, and $r_{i,j}$, $g_{i,j}$, and $b_{i,j}$ are the R, G, and B values for position (i,j), which are normalized into 0 to 1 values. The inputs from the neighbors are then given by

$$\partial D^{R}_{i,j} = \sum_{u,v\in\mu} s_{i,j,u,v} W_{i,j,u,v} (x^{R}_{u,v} - x^{R}_{i,j}), \tag{2.4a}$$

$$\partial D^{G}_{i,j} = \sum_{u,v\in\mu} s_{i,j,u,v} W_{i,j,u,v} (x^{G}_{u,v} - x^{G}_{i,j}), \tag{2.4b}$$

$$\partial D^{B}_{i,j} = \sum_{u,v\in\mu} s_{i,j,u,v} W_{i,j,u,v} (x^{B}_{u,v} - x^{B}_{i,j}), \tag{2.4c}$$

where μ is a set of neighbors for position (i,j), $s_{i,j,u,v}$ is a link determined by the higher layer, and $W_{i,j,u,v}$ is the connection strength between the sets of oscillators $\left(x^{R}_{i,j}, x^{G}_{i,j}, x^{B}_{i,j}\right)$ and $\left(x^{R}_{u,v}, x^{G}_{u,v}, x^{B}_{u,v}\right)$. We apply the mirror reflection for the boundary condition. Therefore, μ for position (i,j) includes (i,j) itself if (i,j) is located on the boundary. The dynamics for $W_{i,j,u,v}$ are determined by using the following equation.

$$\begin{aligned} dW_{i,j,u,v}/dt = {} & \psi(x^{R}_{i,j} - x_{th}, x^{R}_{u,v} - x_{th}) \\ & + \psi(x^{G}_{i,j} - x_{th}, x^{G}_{u,v} - x_{th}) \\ & + \psi(x^{B}_{i,j} - x_{th}, x^{B}_{u,v} - x_{th}), \\ & for\ W_{min} \le W_{i,j,u,v} \le W_{max} \end{aligned} \tag{2.5}$$

where x_{th} denotes the threshold for firing, W_{min} and W_{max} are the lower and upper limits for $W_{i,j,u,v}$, and $\psi(\zeta,\xi)$ is given by

$$\psi(\zeta,\xi) = \begin{cases} w & (\zeta > 0 \cap \xi > 0) \\ -w & (\zeta\xi < 0) \\ 0 & (otherwise), \end{cases} \tag{2.6}$$

which is a simple expression of the synaptic learning rules. The activities of the oscillator and the connection strength are shown in Figs. 2.3 and 2.4.

The higher layer, which describes groups of pixels, is a set of multiple vectors. On this layer, the groups, which are sets of pixels that are strongly connected with each other, are formed according to the connection strength among the pixels. The pixel at position (i,j) and that at position (u,v) are joined together into the same group when

$$W_{i,j,u,v} - w_{th} > 0, \tag{2.7}$$

where w_{th} denotes the threshold for the connection strength. The k-th group is described using vector I_k, which denotes the average RGB values of the component pixels, and is given by

$$I_k = \frac{1}{N_k} \sum_{i,j\in\phi_k} \begin{pmatrix} r_{i,j} \\ g_{i,j} \\ b_{i,j} \end{pmatrix}, \tag{2.8}$$

where ϕ_k is a set of the component pixels for the k-th group, N_k is the number of pixels included in ϕ_k, and $r_{i,j}$, $g_{i,j}$, and $b_{i,j}$ are the RGB values of the pixel at position

(i, j). According to the values of vector I_k, the links among the component pixels are reformed as follows:

$$s_{i,j,u,v} = \begin{cases} 1 & (|I_k - I_{i,j}| < I_{th}) \\ 0 & (otherwise) \end{cases} for \forall (u, v) \in \phi_k, \tag{2.9}$$

where

$$I_{i,j} = \begin{pmatrix} r_{i,j} \\ g_{i,j} \\ b_{i,j} \end{pmatrix}, \tag{2.10}$$

and I_{th} is a threshold for determining the component pixels. The activity of the group formation is shown in Fig. 2.5.

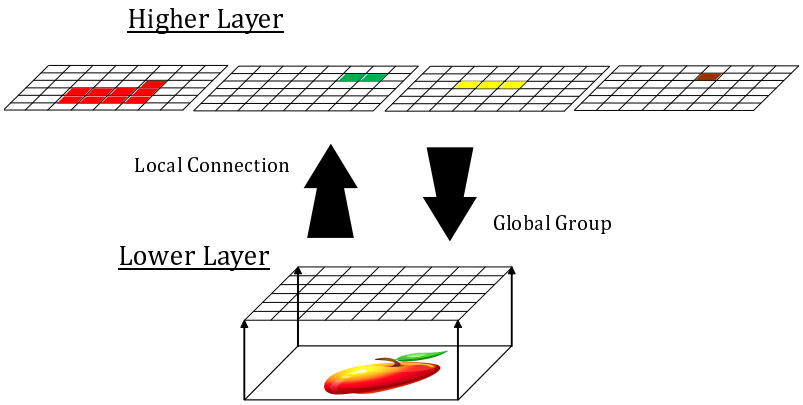


Fig. 2.1 Model construction, which consists of two hierarchical layers. The lower layer forms the local connections by representing the RGB values of each pixel in an input image. Referring to the locally formed connections, the higher layer forms groups of pixels. Subsequently, the lower layer updates the local connections according to the groups formed by the higher layer. In this example, the higher layer forms four groups, each of which is colored red, green, yellow, and brown by referring to the locally formed connections. Note that the number of groups is previously unknown, and is autonomously formed.

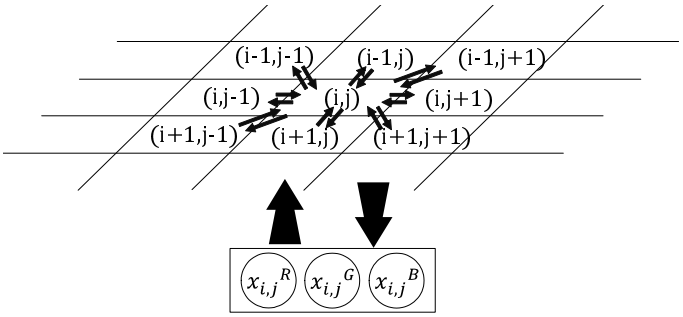


Fig. 2.2 Construction of lower layer. The pixel in image at position (i, j) is described by the activities of a set of oscillators as $\left(x^R_{i,j}(t), x^G_{i,j}(t), x^B_{i,j}(t)\right)$, which correspond to the R, G, and B values of the pixel. The pixel at position (i, j) is connected to each of the neighboring pixels at positions $(i-1, j-1), (i-1, j), (i-1, j+1), (i, j-1), (i, j+1), (i+1, j-1), (i+1, j), and (i+1, j+1)$ using $W_{i,j,u,v}$ ((u, v) is the position of each of the neighboring pixels).

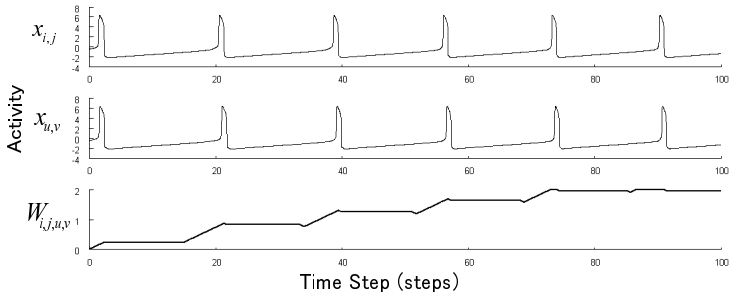


Fig. 2.3 Activities of oscillators and connection strength between them when oscillators are synchronized. The first row denotes the activity of oscillator $x_{i,j}$, the second row denotes that of oscillator $x_{u,v}$, and the third row denotes that of the connection strength $W_{i,j,u,v}$ between oscillators $x_{i,j}$ and $x_{u,v}$. When both $x_{i,j}$ and $x_{u,v}$ fire ($x_{i,j} > x_{th}$ and $x_{u,v} > x_{th}$), the $W_{i,j,u,v}$ value increases until $W_{i,j,u,v} = W_{max}$ so that the connection is strengthened. In this case, $W_{max} = 2$.

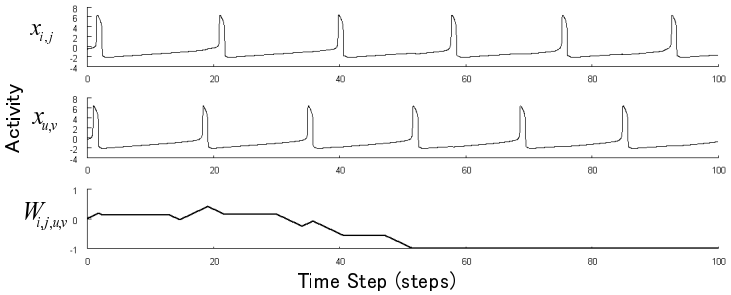


Fig. 2.4 Activities of oscillators and connection strength between them when oscillators are not synchronized. The first row denotes the activity of oscillator $x_{i,j}$, the second row denotes that of oscillator $x_{u,v}$, and the third row denotes that of the connection strength $W_{i,j,u,v}$ between oscillators $x_{i,j}$ and $x_{u,v}$. When either $x_{i,j}$ or $x_{u,v}$ fires (but not both), the $W_{i,j,u,v}$ value decreases until $W_{i,j,u,v} = W_{min}$ so that the connection is weakened. In this case, $W_{min} = -1$.

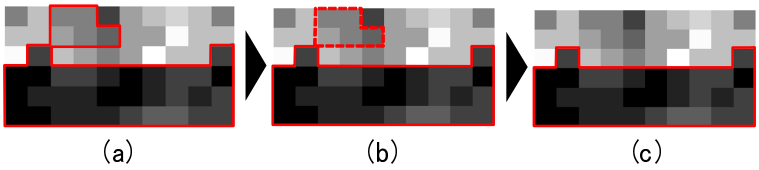


Fig. 2.5 Activity of group formation. The solid red line in (a) denotes the elements of the group formed using the local connections formed by the lower layer. The broken red line in (b) denotes the elements excluded from the group. The solid red line in (b) and (c) denotes the elements of the group reformed using the higher layer.

2.3 Results

In this section, we evaluate the effectiveness of our proposed model. For this evaluation, we present the experimental results using the images selected from Microsoft clip art shown in Fig. 2.6. Referring to the compressed images, the states of the oscillators corresponding to the RGB values of each pixel are updated using the differential equations shown in Section 2.2. The connections among the oscillators and the groups of oscillators are, then, also accordingly updated.

For this evaluation, the images shown in Fig. 2.6 are compressed so that the image size is smaller than 60×60 px. The compressed images in Fig. 2.7 are constructed using an averaging filter. The oscillators are updated with the relevant parameter values, $A_1 = 3.0$, $B_1 = -20.0$, $C_1 = 1.0$, $A_2 = 1.0$, $B_2 = -0.0$, $C_2 = -1.0$, $\rho = 5.0$, $\lambda = 2.0$, $x_{th} = -1.13$, $W_{min} = -1.0$, $x_{max} = 2.0$, $w = 0.1$, and $I_{th} = 0.5$. Then, the equations presented in Section 2.2 are solved using the fourth order Runge-Kutta method with a micro-sampling time interval of $dt = 0.02$. Accordingly, the connections among the oscillators and the groups are updated, as shown in Figs. 2.8 and 2.9. The connection strength between each oscillator, which is initially set to zero, gradually increases as time evolves if the dynamics of its edges synchronize. As a result of the increase in connection strength, the groups of oscillators are formed. We define the following evaluation indices in order to visualize the group forming status.

$$\epsilon_{intra} = \sum_{i,j,u,v\in\psi_{intra}} |I_{i,j} - I_{u,v}|/N_{intra}, \tag{2.11a}$$

$$\epsilon_{inter} = \sum_{i,j,u,v\in\psi_{inter}} |I_{i,j} - I_{u,v}|/N_{inter}, \tag{2.11b}$$

$$\epsilon_{difference} = \epsilon_{inter} - \epsilon_{intra}, \tag{2.11c}$$

where ψ_{intra} is a set of connections both of which edges (i, j) and (u, v) are located in the same group, ψ_{inter} is a set of the connections both of which edges (i, j) and (u, v) are located in different groups, N_{intra} is the number of ψ_{intra} elements, and N_{inter} is the number of ψ_{inter} elements. $\epsilon_{difference}$, the subtraction of ϵ_{intra} from ϵ_{inter}, tells us how well each oscillator is divided into suitable groups. As shown in Fig. 2.10, the evaluation indices, which are unstable in the early stage, become stabilized after about 10000 time steps. This tells us that our model repeats the trial and error process in the early stage by changing the local connections until suitable groups are formed. As a result of the trial and error process shown in Fig. 2.8, pixels with similar colors are classified into the same clusters if they are not geometrically intermittent, as shown in Fig. 2.9.

We then make a comparison between our model and a model without the higher layer in order to confirm the role that the higher layer plays. In particular, we set the threshold for determining the component of groups I_{th} to infinity ($I_{th} = 999999$). As a result, $s_{i,j,u,v}$, the link between positions (i, j) and (u, v), is stably set to 1. With this configuration, the states of the oscillators, connections, and groups are updated in the same manner. The dynamics of the connection strength are shown in Fig. 2.11. Unlike when the I_{th} is suitably set, these results show that the pixel groups can never be formed using this configuration. As a consequence, we confirmed that the inhibition from the higher layer is indispensable for forming the pixel groups.

Table 2.1 Relationship among D, vibration period, and frequency.

D	Vibration Period (sec.)	Frequency (Hz)
0	55.70	0.01795
1	29.32	0.03411
2	21.70	0.04608
3	17.54	0.05701
4	14.86	0.06729
5	12.92	0.07740
6	11.46	0.08726
7	10.32	0.09690
8	9.380	0.1066
9	8.600	0.1163
10	7.940	0.1259

We finally give an interpretation of our results from the neuropsychological viewpoint. Our model is based on the KYS oscillator, which was originally introduced to simulate the neuronal cells dynamics [31, 48]:

$$\begin{aligned} d^2x/dt^2 + \left(A_1x^2 + B_1x + C_1\right) dx/dt \\ + \left(A_2x^2 + B_2x + C_2\right) x = D, \end{aligned} \tag{2.12}$$

where x denotes the activity of an oscillator and D denotes the input from outside. The relationship among value D, the vibration period, and frequency as a result of the activity of the single oscillator is summarized in Table 2.1. The relationship between value D and the vibration frequency can then be linearly approximated, as shown in Fig. 2.12, using

$$f = \alpha D + \beta, \tag{2.13}$$

where f denotes the frequency and $\alpha = 0.0105$ and $\beta = 0.0234$. With this relationship, we try to explain how our model recognizes the image colors. Figs. 2.13-2.16 show the RGB values of the original images and those simulated using the observed values for the frequency of the oscillators. These results tell us that the color recognized using the neuronal oscillators may be different from that of the original images, as shown using the various optical illusions examples [54, 55]. Interestingly, even pixels that belong to the same group can describe different frequencies. This fact implies that the human visual cortex may recognize the area and color simultaneously so that multiple colors in the same area can be investigated.

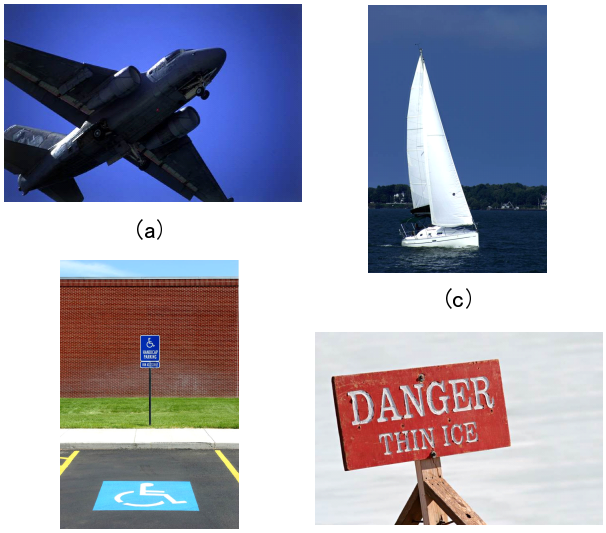


Fig. 2.6 Original images for evaluation, which are selected from Microsoft clip art.

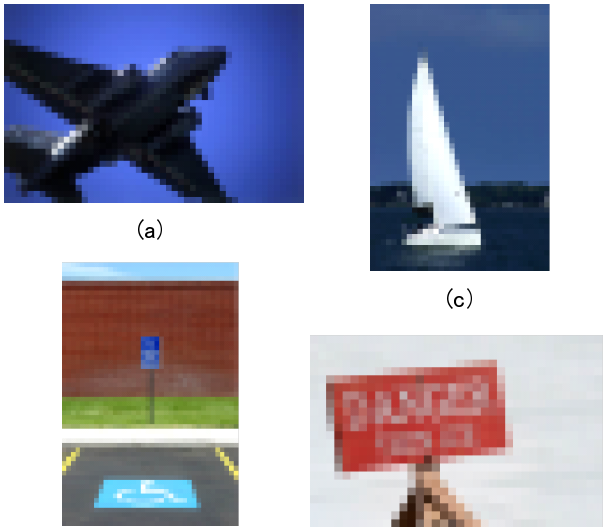


Fig. 2.7 Compressed images based on original images shown in Fig. 2.6. The pixel sizes of (a)-(d) are 33×50 (a), 57×38 (b), 50×34 (c), and 24×36 (d).

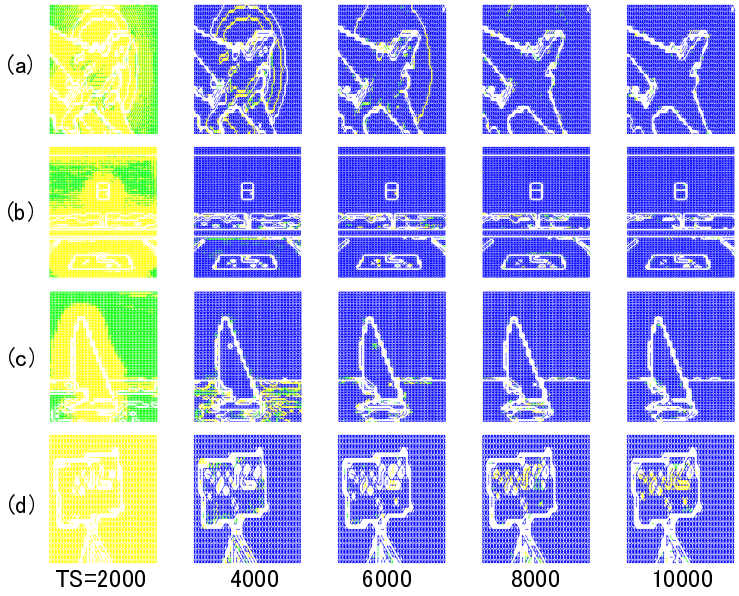


Fig. 2.8 Evolution of connection strength between pixels at time steps (TS) 2000, 4000, 6000, 8000, and 10000. The groups as a result of the connected pixels are gradually formed as the connection growth. (a)-(d) represent Figs. 2.7(a)-(d).

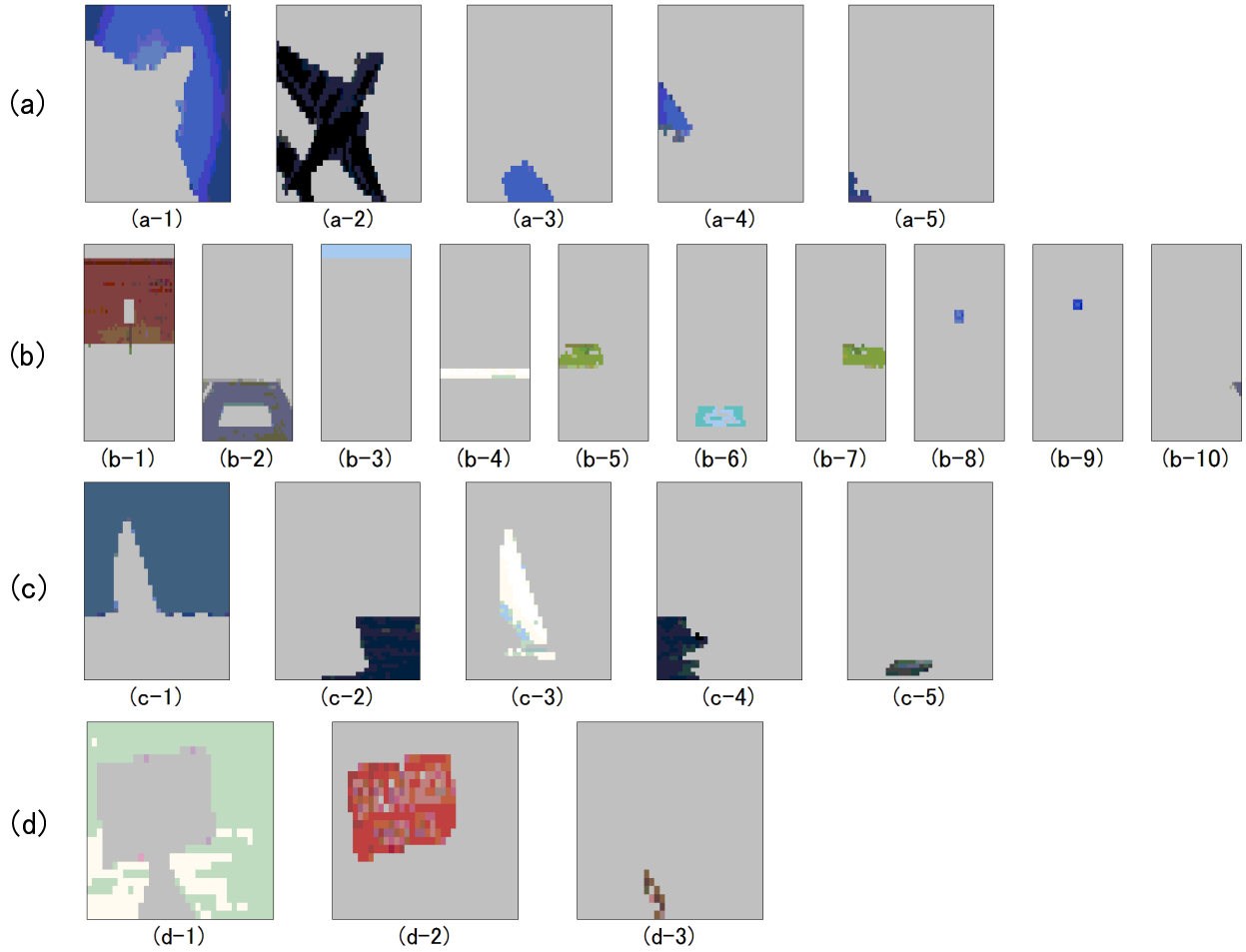


Fig. 2.9 Pixel groups formed at time step (TS) 50000. (a)-(d) represent Figs. 2.7(a)-(d).

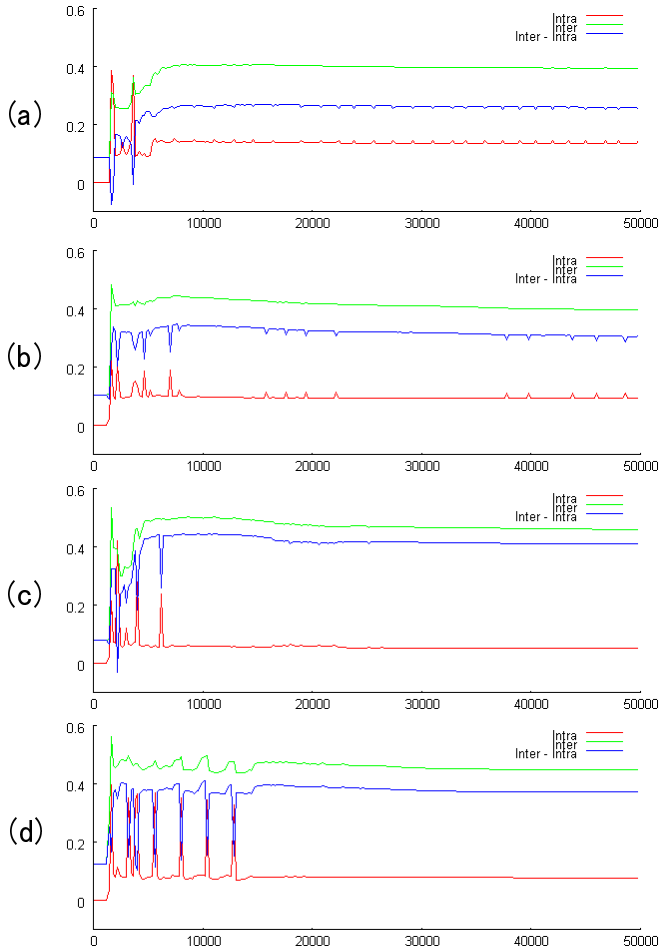


Fig. 2.10 Evolution of evaluation indices. The red line denotes the average difference in the RGB values of the elements in the same group ϵ_{intra}, the green line denotes the average difference in the RGB values of the elements in different groups ϵ_{intra}, and the blue line denotes the subtraction of ϵ_{intra} from ϵ_{inter}. (a)-(d) shows these indices corresponding to Figs. 2.7(a)-(d).

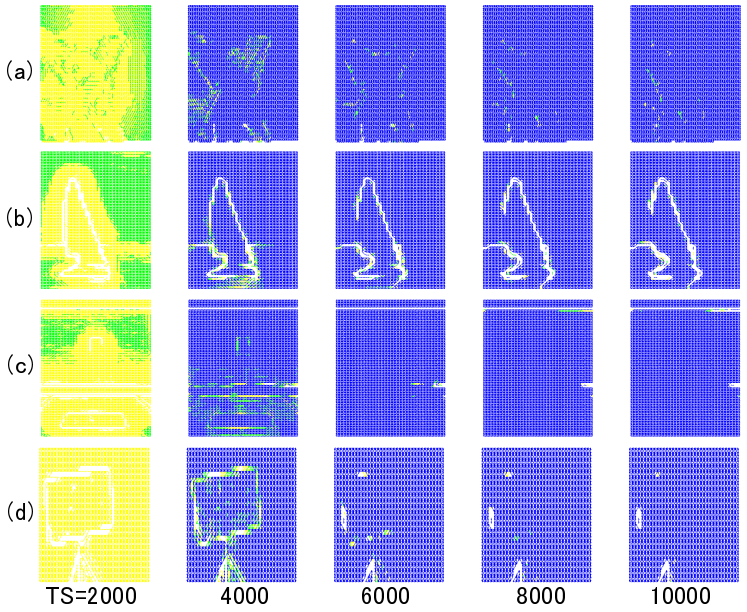


Fig. 2.11 Evolution of connection strength between pixels at time steps (TS) 2000, 4000, 6000, 8000, and 10000 when $I_{th} = 999999$. Unlike in Fig. 2.8, the connection strength between the groups perceived when $I_{th} = 0.5$ (original value) is getting stronger, so the borders of the groups are becoming more vague. (a)-(d) represent Figs. 2.7(a)-(d).

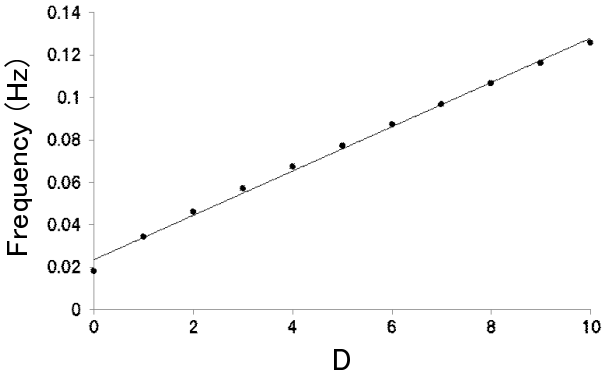


Fig. 2.12 Relationship between value D and vibration frequency (f) (experimental results) as result of activity of single oscillator classified in Table 2.1. Their relationship is linearly approximated as $f = 0.00105D + 0.0234$.

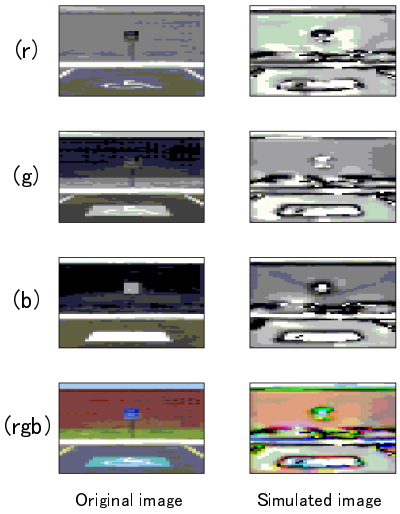


Fig. 2.13 Comparison of RGB values of original image Fig. 2.7(a) and those simulated with observed values for frequency of oscillators. The value in red is shown as (r), green is shown as (g), blue is shown as (b), and RGB value is shown as (rgb).

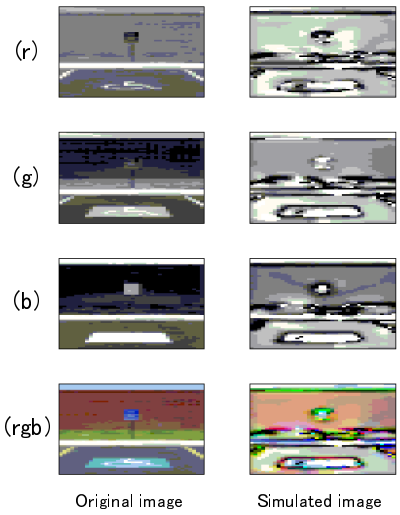


Fig. 2.14 Comparison of RGB values of original image in Fig. 2.7(b) and those simulated using observed values for frequency of oscillators. The red value is shown as (r), green is shown as (g), blue is shown as (b), and the RGB value is shown as (rgb).

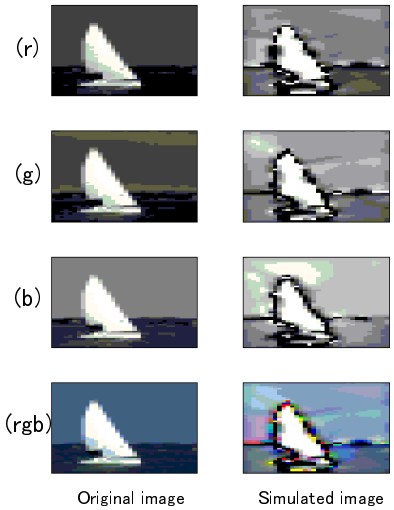


Fig. 2.15 Comparison of RGB values of original image in Fig. 2.7(c) and those simulated using observed values for frequency of oscillators. The red value is shown as (r), green is shown as (g), blue is shown as (b), and the RGB value is shown as (rgb).

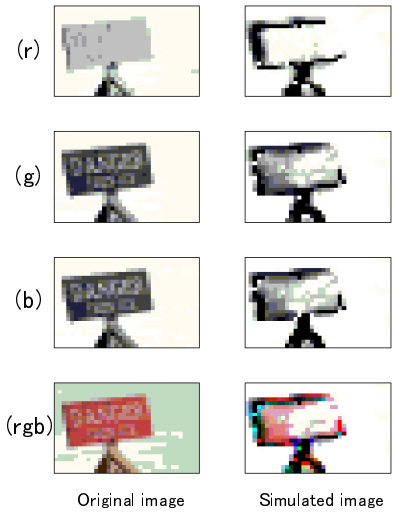


Fig. 2.16 Comparison of RGB values of original image in Fig. 2.7(d) and those simulated using observed values for frequency of oscillators. The red value is shown as (r), green is shown as (g), blue is shown as (b), and the RGB value is shown as (rgb).

2.4 Discussion and Conclusion

We proposed a model of the perceptual grouping in color images using the interaction between the globally formed groups and locally formed connections in this chapter. We then evaluated the effectiveness of our proposed model using color images. As a result, we found that our model successfully forms groups and segregates these spatial regions. In this section, we discuss five topics: the mechanism for the convergence of the grouping, the difference between our model and the previously proposed clustering approaches, the difference between our model and the previously proposed models from the view of the role that the higher layer plays, the meaning of the resulting clusters, and the future work to be done with respect to our approach.

We first discuss the mechanism for the convergence of the grouping. The role of the connection strength $W_{i,j,u,v}$ between oscillators $x_{i,j}$ and $x_{u,v}$ is the excitation when $W_{i,j,u,v} > 0$, and the inhibition otherwise. This mechanism is explained by looking at the equations noted in Eq. 2.4. As a matter of course, the connection strength $W_{i,j,u,v}$ does not work as a neighbor excitor nor inhibitor when $s_{i,j,u,v} = 0$. When $W_{i,j,u,v} > 0$, $x_{i,j}$ gains energy so that $x_{i,j}$ increases its frequency to repel each other, but $x_{i,j}$ loses energy so that $x_{i,j}$ decreases its frequency otherwise. As a result, the activities of $x_{i,j}$ and $x_{u,v}$ are locked together (frequency-locking phenomenon). In contrast, the activities of $x_{i,j}$ and $x_{u,v}$ repel each other when $W_{i,j,u,v} < 0$. For this mechanism, the system is converged to a metastable state once the grouping occurs and excitation occurs between the oscillators belonging to the same group and inhibition occurs between those belonging to different groups. This phenomenon is shown with ψ_{intra}, ψ_{intra}, and $\psi_{difference}$ defined by using Eqs. 2.11. As shown in Fig. 2.10, ψ_{intra}, ψ_{intra}, and $\psi_{difference}$ become stable after vibration in the early stage.

We next discuss the difference between our model and the previously proposed clustering approaches. As described in Section 2.1, the models of the globally formed groups as a result of the locally formed connections have not yet been proposed. Previously proposed clustering approaches are roughly divided into two methods: hierarchical clustering and non-hierarchical clustering. Hierarchical clustering methods such as [56–58] form clusters based on the local connections. With these methods, the clusters are gradually formed in a manner in which the near nodes form clusters according to the distances between them. These methods, therefore, form clusters according to the local connections. Non-hierarchical clustering methods such as [59–61] form clusters based on the global features. For example, the k-means clustering method, which is widely used in the non-hierarchical clustering method, classifies nodes based on the centroids that are randomly determined so that the nodes are classified into k clusters ("k" is pre-determined). For this reason, the clusters formed by using the k-means clustering method are not formed using the local connections (centroids cannot be determined using the local features). Although more sophisticated methods have been proposed, it is still true that the models for globally formed groups as a result of the locally formed connections have not yet been proposed.

Let us now discuss the difference between our model and the previously proposed models from the viewpoint of the role that the higher layer plays. As described in Section 2.3, the role of the higher layer is the inhibition between the oscillators belonging to different groups. Historically, a long-range inhibitory connection has been well investigated and neural network models with a long-range inhibition have been proposed for the perceptual grouping [37–43,62–66]. Since the concept of global

inhibition is to inhibit remote nodes and excite neighboring nodes, globally formed groups cannot influence locally formed connections.

We next discuss the resulting clusters shown in Fig. 2.9. Although the shapes of some clusters are markedly different from those the human visual system perceives, it is still important for our models to autonomously form "centroids" of clusters. As a computational visual system, shapes can be captured using parameter tuning based on the shape representations once the centroid in the color coordinate is obtained. It is also plausible that the human visual system has a mechanism in which the shape and color features interact with each other. In addition, it is also known that the primary visual cortex that detects the contrast and models to improve the edge sharpness have widely been proposed [67–69]. Using these models as a preprocessing for our model, the shapes of the formed clusters are expected to resemble those of the surface perceived by the human visual cortex.

Finally, we discuss the future works to be done with respect to our approach. The first work is for the parameter set (RGB) that drives the neural oscillators. Since it is inferred that two parameters (red/green) are enough to perceive color [49–53], it is plausible that the RGB parameter set is reduced to RG. This parameter reduction is the first subject for our future work. The second work is to use our model for reproducing the pop-out effect [70–72]. It have been pointed out that the stimuli that contain similar perceptual features pop out from the background and these features form a surface. Since the mechanism for the surface formation seems quite similar to that of our model, our model is expected to be used for reproducing the pop-out effect. The final work is to integrate the groups formed by our model. We assume that our model successfully integrates the local pixels into the global groups. These groups are, however, composed of pixels with similar colors. In order to identify the general objects in images, these groups must be integrated as a more global shape. This group integration is the final subject for our future work.

Chapter 3

Shape Representation
Visual shape representation with geometrically characterized contour partitions

3.1 Introduction

It is a challenging issue for computational visual systems to identify general shapes flexibly of real-world scenes. Computational visual systems describe the real-world as images which consist of pixels. In the images, shapes are not described as whole bodies. Unless the shapes are described as whole representations, computational visual systems cannot basically extract and identify them. The human visual system is, on the other hand, assumed to identify shapes flexibly using a suitable shape representation. For this reason, the shape representation described in the human visual system needs to be unveiled. The shape representation is known to be formed through visual ventral pathway which hierarchically integrate the stimuli caught by the retinal cells using the cortical cells which have multiple-sized receptive fields [1–3, 10]. The hierarchical processing models for shape recognition have been proposed based on hierarchical neural network models [73–75].

The conventional hierarchical models, however, have not yet constructed the shape representation suitable for shape matching since the representation they construct does not fulfill all of the constraints that are required for shape matching:

1. The original shape should be represented with a group of partitioned contours in order to retrieve the whole shape (global information) from the partial contour (local information). In case the original shape is not represented with a discrete feature, the whole shape is basically required to be identified. The human visual system is, however, known to identify shapes even when they are partially occluded [12–15]. Therefore, shapes should be represented with a set of the parts. In addition, neurophysiological results support that the shapes are plausibly represented with a group of curves in area V4 [5, 6, 8]. The representation with the group of curves is created through the hierarchical structure from the retina to area V4 and the horizontal structure in each area [76]. For these reasons, it is quite reasonable that the original shape is represented with a group of partitioned contours in human visual system.

2. Coarse and fine structures of the original shapes should be individually represented in order for the visual system to respond to shapes as quick as possible with least number of features of them, and to discriminate a shape from others with the detailed information. Various psychophysical researches support that shapes are represented individually with coarse and fine structures of them [16–19]. For example, researches on face recognition imply that lower spatial frequency contributes to recognize whole faces and higher frequency makes us understood the detailed shapes [77, 78]. In addition, neurophysiological studies also support this coarse and fine structures. For example, Drucker applied shapes defined by radial frequency components (RFCs) and observed a coarse spatial coding of shape features in the lateral occipital complex (LOC) and a fine coding in ventral LOC [79]. These evidences imply that the human visual system identifies coarse and fine structures of shapes individually using a spatial frequency coding.
3. The shape recognition realized with the visual system should be invariant to geometric transformation such as rotation or shear. It is observed that some of Neuronal responses in inferotemporal cortex (IT) are invariant to geometric changes of shapes such as size and position changes [20] or changes of view points [21]. It is, then, quite acceptable that the shape representation is described with geometric parameters since the geometric structure of features are preserved in early visual cortex according to the retinotopy [3]. Based on these two evidences, it is plausibly supposed that the shape representation, which is described with geometric parameters, is transformed from early visual cortex to IT so that the shape recognition is invariant to geometric transformation caused by such as changes of view points.

The typical neural network models do not construct representations which fulfill all of these three constraints. Fukushima's model [73] and Poggio's model [74], for example, reconstruct the neuron's hierarchical receptive fields. Using these neurons, they hierarchically integrate the pixels of input images, and finally recognize the pattern of the images with the neurons of the deepest layer. Although these models reconstruct the hierarchical pattern of the neural network well, they do not represent the global and local structures of input images. Grossberg's model [75], on the other hand, focuses on the global and local structures of images, and makes it possible to classify multiple scenes. With these neural network models, however, the geometric relationships of the integrated features are steadily fixed, hence their models are not invariant under geometrical transform.

Computational recognition approaches, on the other hand, are challenging to develop representations suitable for shape recognition especially in the context of template matching [80, 81]. The template matching is psychophysically plausible to be applied for the recognition process of the human visual system [82]. Although most of the computation-based representations are not based on neurophysiological nor psychophysical evidence, some representation approaches are proposed in a psychophysical context [83]. These approaches, however, have not yet solved the three problems each of which are corresponding to the three constraints for the shape representation:

1. The original shapes are required to be retrieved from the partial contour information.
2. The process to pay attention to shapes and that to analyze them are required to be individually done.
3. The representation is required to be geometrically invariant.

A good example is curvature scale space (CSS), which is a shape representation using the curvature zero-crossing points of the closed contour at varying levels of detail [84–87]. With CSS, the shape contour is partitioned with the curvature zero-crossing points, coarse and fine structures are independently described at varying levels of detail, and the geometrical transform is fixed with the modification of the contour coordinate. CSS does not, however, solve the first and third problems especially when partially occluded shapes are represented. CSS cannot be described unless the whole of the original shape is obtained. The original shapes are, therefore, cannot be retrieved from partial information. Although the revised representations of CSS are proposed [88, 89], they have not solved all of three problems yet. In this way, a biologically plausible shape representation that solves these problems has not yet been proposed.

In this chapter, we propose a biologically plausible representation for matching general shapes with their contour curvature information. Although our representation is not based on hierarchical neural network models, it reflects the constraints that are required for shape matching. We firstly define a "curvature partition" as a descriptor of a group of partitioned contours. We, then, implement a Gaussian blur in multiple scales in order to obtain a coarse and a fine spatial coding. Finally, we apply the geometric transformation in order to achieve the invariant shape recognition. Section 3.2 describes the overview of our method. Section 3.3 describes our proposed method for constructing the shape representation with geometric parameters. Section 3.4 describes a similarity measurement that is applied for the representation. Section 3.5 shows experimental results obtained using MPEG-7 CE-Shape-1 part B data set for evaluating for general shapes and numeral characters as examples of shapes that previous contour based representations have a difficulty in. Section 3.6 presents a discussion based on these results and concludes this chapter.

3.2 Overview of the Method

We propose a new representation (called a "curvature partition") of shapes and a method that uses curvature partitions to measure the similarity of two shapes. The curvature partition is defined as a set of vectors of sub-vectors each of which consists of five elements in multiple scales of smoothing. The process of constructing a curvature partition is shown in Fig. 3.1. Let us assume that two planar shapes A and B are compared to estimate their similarity (Fig. 3.1(a)).

For each shape, the curvature partitions are constructed. Shape contours are expressed as sample points on the outer boundary of shapes. Each contour is expressed as a two-dimensional vector C:

$$C = (X(t), Y(t)), \tag{3.1}$$

where t is the accumulated path length from a starting point on the contour. Using vector C in multiple scales of smoothing, curvature partitions are constructed as partial curves of which both edges are curvature zero-crossing points (Fig. 3.1(b)). Curvature partitions are expressed as $P_i (i = 1, 2, \ldots, n)$, where n is the number of curvature partitions that the contour has. Curvature partitions are then divided into N sub-partitions (Fig. 3.1(c)). Each sub-partition p_{ij} is expressed with its average curvature κ_{ij}, length l_{ij}, normal direction θ_{ij}, and location $o(x_{ij}, y_{ij})$ (Fig. 3.1(d)):

$$p_{ij} = \begin{pmatrix} \kappa_{ij} \\ l_{ij} \\ \theta_{ij} \\ x_{ij} \\ y_{ij} \end{pmatrix}, \tag{3.2}$$

where $j = 1, 2, \ldots, N$. Therefore, curvature partition P_i has $N \times 5$ elements:

$$P_i = \begin{pmatrix} \kappa_{i1} & \kappa_{i2} & \ldots & \kappa_{iN} \\ l_{i1} & l_{i2} & \ldots & l_{iN} \\ \theta_{i1} & \theta_{i2} & \ldots & \theta_{iN} \\ x_{i1} & x_{i2} & \ldots & x_{iN} \\ y_{i1} & y_{i2} & \ldots & y_{iN} \end{pmatrix}. \tag{3.3}$$

In this way, each curve on a contour (which is a partial curve whose edges are both curvature zero-crossing points) is expressed by its curvature, length, normal direction, and location. The curvature partitions $P_1, P_2, \ldots, P_n$ in multiple scales of smoothing are stored as representation of the shape (Fig. 3.1(e)). The relationship between the partition P and the sub-partition p is shown in Fig. 3.2.

To derive the similarity of shapes A and B, we first compare their curvature partitions with each other in multiple scales (Section 3.4). Then, we choose the best similarity from those of multiple scales as the similarity of shapes A and B. The partitioning enables the correspondence between two points on A and B to be determined much more easily than with point matching methods [90, 91]. The least-squares estimation method is then used to perform linear transformation. Finally, the similarity of the two shapes is derived as the sum of the distances of curvature partition features.

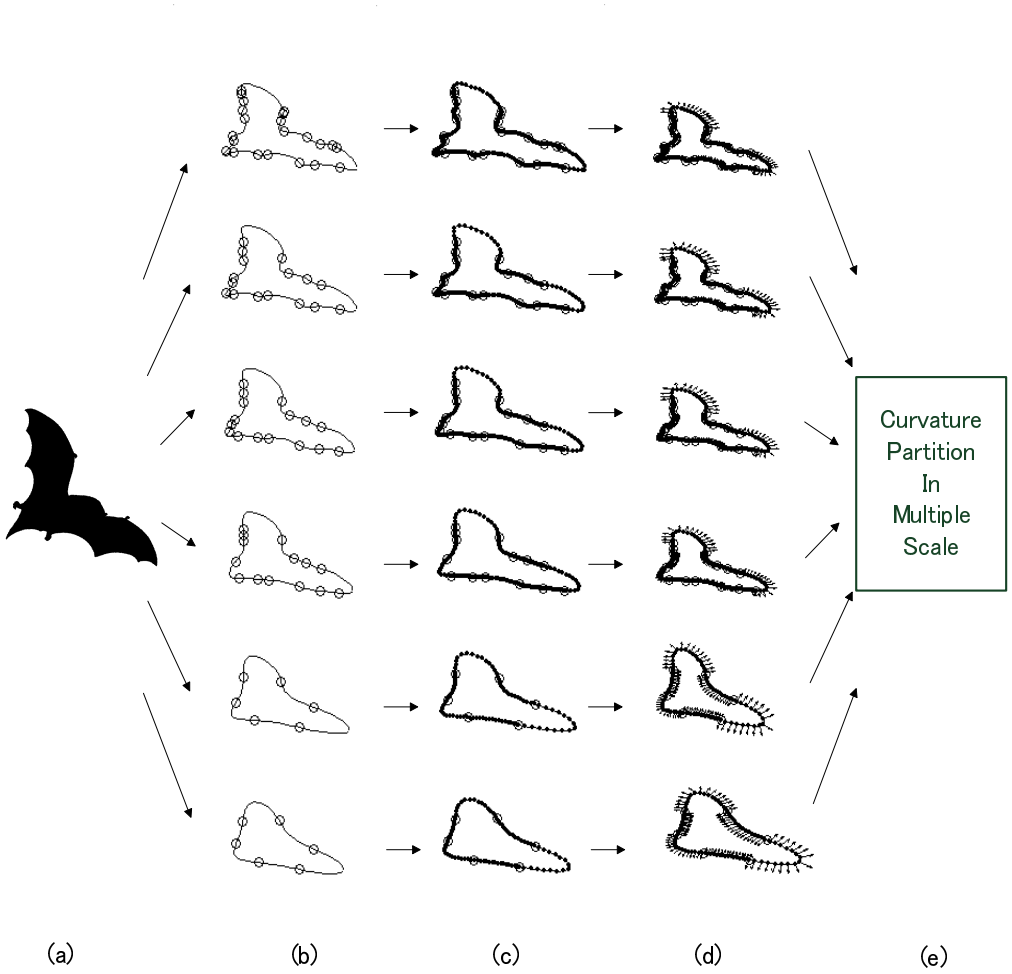


Fig. 3.1 Process for constructing the vector of a curvature partition. The curvature partition is defined as a set of vectors of sub-vectors that has five elements in multiple scales of smoothing. First (a), the contour C is extracted from an image. Then (b), the contour C is smoothed in multiple scales, and curvature zero-crossing points are detected on each smoothed contour. The curve between two curvature zero-crossing points is called a "curvature partition". Next (c), the curvature partitions are divided into N sub-partitions. Then (d), all sub-partitions are specified with their average curvature, length, normal direction, and location. Finally (e), the curvature partition vectors expressed with sub-partitions in multiple scales are stored as a representation of the shape. Here, examples when $\sigma = 2, 4, 8, 16, 32$, and 64 are shown.

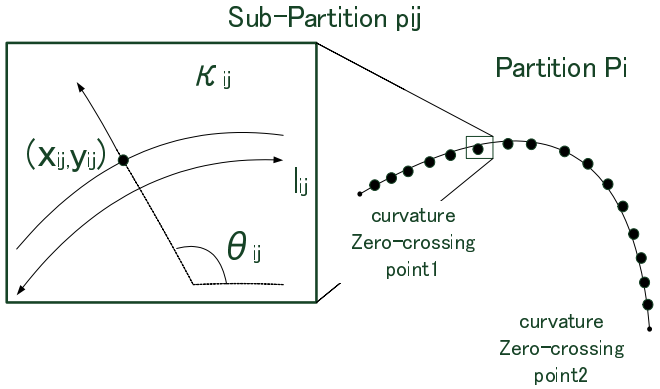


Fig. 3.2 The relationship between the partition and the sub-partition. The partition is the partitioned smoothed contour segregated with curvature zero-crossing point. The sub-partition is, then, the segregated partition represented with its curvature, length, normal direction, and location.

3.3 Curvature Segment Representation

3.3.1 Curvature Partition Construction

Shape contours, which are expressed as vector C, are basic coordinates for delineating partition borders. CSS representation uses the hierarchical smoothing approach to extract the boundary points as the curvature zero-crossing points [84–87]. CSS representation, known to be robust to noise and geometrical transforms, is constructed with two steps [84] as shown in Fig. 3.3. The first step is smoothing for a shape's outer contour (Fig. 3.3(a) to (b)). Convolved with a one-dimensional Gaussian kernel $g(t,\sigma)$ of width σ, the contour C evolves to the smoother contour $C(X(t,\sigma),Y(t,\sigma))$, where $X(t,\sigma)$ and $Y(t,\sigma)$ are the convolved expressions of $X(t)$ and $Y(t)$ with $g(t,\sigma)$. At the second step, curvature zero-crossing points are detected on the smoothed contour. On each sample point $(X(t,\sigma),Y(t,\sigma))$, curvature $\kappa(t,\sigma)$ is derived in the manner given in [84]. In this chapter, a curvature zero-crossing point is defined as the point where the value of the curvature κ is 0. These curvature zero-crossing points $C_0(t,\sigma)$, for which curvature $\kappa = 0$, are then detected and plotted as the CSS representation, where the x-axis is t and the y-axis is σ (Fig. 3.3 (c)).

CSS representation divides the shape contour into multiple partitions. These partitions (called "curvature partitions" in this chapter), are units for the template matching. This partitioning method is known to be robust for partially occluded shapes with their average curvature and location information [88,89]. For a fixed value of σ, curvature zero-crossing points $C_0(t,\sigma)$ are extracted. These points are defined as the boundary points for curvature partitions. An example curvature partition is shown in Fig. 3.1. In this chapter, the values of σ are fixed at $2, 4, 8, 16, 32$, and 64, which are approximate multiples of 2.

3.3.2 Sub-Partition Construction

Curvature partitions, i.e., contour parts whose edges are both curvature zero-crossing points, are divided into N sub-partitions each of which has equal arc-length in a partition. In this chapter, $N = 16$ is set. These sub-partitions are expressed with their average curvature, length, normal direction, and location (x, y). Based on the hypothesis that shapes are coded with their curvature and angular position in area V4 [5,6], we describe partial contours as arcs. Use of the radius estimation enables our sub-partition expression to describe these arcs as shown in Fig. 3.4. This figure also shows that our representation reconstructs the partial contour of occluded shapes in the same manner.

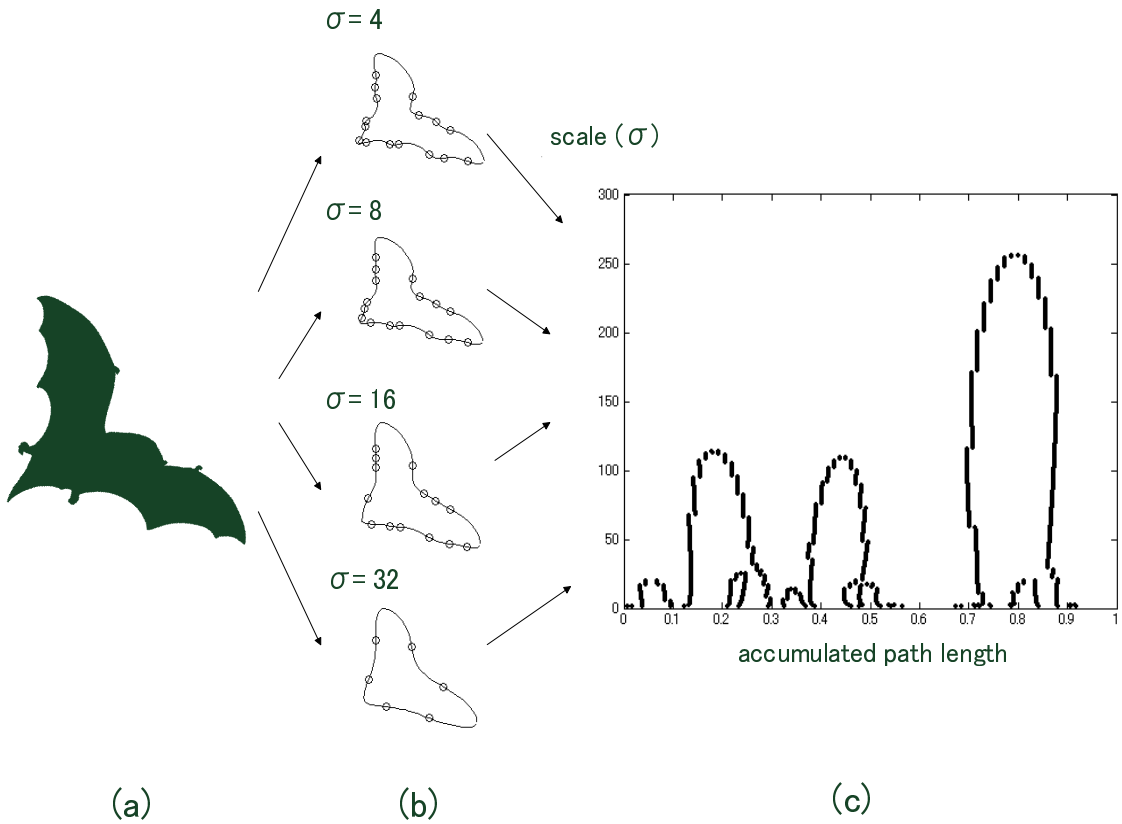


Fig. 3.3 Example of CSS representation. Parts (a), (b), and (c) respectively show the contour of an original shape, the smoothed contour in multiple scales, and the CSS representation computed from the contour. After being extracted from an image, the contour (left) is smoothed in multiple continuous scales. Curvature zero-crossing points are then detected on the smoothed contours. These points are shown as circled dots and the smooth contours are shown as dots (center). Finally, the sets of the scale and the location of these curvature zero-crossing points are plotted on a normalized accumulated path length as a CSS representation (right).

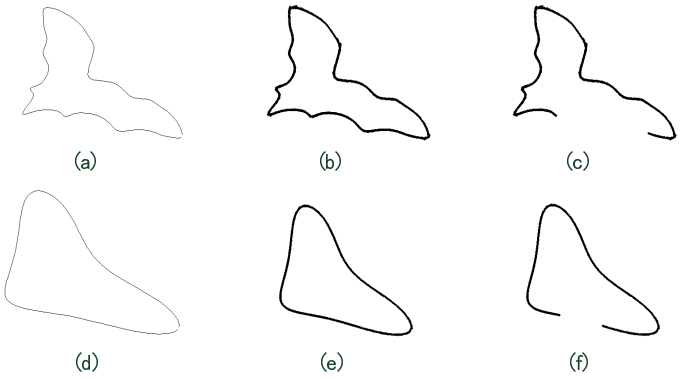


Fig. 3.4 Shape contours, their estimated shapes, and those in case some partitions are occluded. A shape contour in a small scale is shown in (a), and that in a large scale is shown in (d). The estimated shape in a small scale is shown in (b), and that in a large scale is shown in (e). Finally, the estimation of occluded shape in a small scale is shown in (c), and that in a large scale is shown in (f). The figures (a)-(f) show that occluded shapes can be represented and estimated in the same manner that the whole shapes are estimated.

3.4 Similarity Measure

3.4.1 Overview of Similarity Measurement Algorithm

The similarity measurement algorithm with the curvature partition representation consists of four steps (shown in Fig. 3.5). The first step is for the partition alignment (first rows of Fig. 3.5). For closed shape contours, the similarity is measured only when the number of the partitions of each shape is equal. For open curves, on the other hand, the similarity is measured in any case since the number of the partitions of the whole of the original shape is unknown. In this chapter, we show the case of the closed shape contours for simplicity. The second step is for geometric transform based on this alignment (second row of Fig. 3.5). On the basis of previous research [89], we propose a transform process to solve the problem of linear deformation (including translation, scaling, rotation, and shear). The third step is for evaluating parameter distances (third row of Fig. 3.5). Since the alignment is done for all combinations, parameter distances are derived for all of them. Finally, in the fourth step, the best combination, i.e., the one that minimizes the parameter distance, is selected (fourth row of Fig. 3.5).

3.4.2 Segment Alignment

The similarity measurement algorithm begins with the curvature partition alignment. This alignment corresponds to the curve partition mapping of the CSS representation-based approach [89]. Suppose that two planar shapes A and B are compared and that their contours are composed of a sequence of ordered curvature partitions respectively as: $CP_a = (CP(i_a))$, for $i_a = 1, 2, \ldots n_a$, and $CP_b = (CP(i_b))$, for $i_b = 1, 2, \ldots n_b$, where CP denotes a vector of the curvature partition. The basis of the method is that the first partition of A, $i_a = 1$, and a partition of B, $i_b = j$, construct a pair. Similarly, the following partitions construct pairs as shown in the second row of Fig. 3.5. Therefore, all candidate pairs appear in this step. The following steps extract the best pair from them. In this chapter, a pair is extracted with all parameters of the curvature partition, while the CSS representation-based approach applies the ratios of average curvature and length of partitions to the mapping index [89].

This pair extraction concept is based on a biological background. According to neurophysiology, each neuron responds to a certain direction, location, and curvature of curves that are applied to it [6, 8]. Further, this response seems to be based on the coordinates on contours [6]. The alignment of the curvature partitions seems to be simple, but point alignment is usually a major issue when there is no focus on the features of wide areas on shapes. For example, a local value such as the curvature value of each sample point on contours is applied when open two-dimensional curves are matched in the methods cited in [80, 90, 91].

3.4.3 Geometric Transform

The second step is for geometric transform based on this alignment. On the basis of previous research [89], we propose a transform process to solve the problem of linear deformation (including translation, scaling, rotation, and shear) in x and y. Note that (x^a_{ij}, y^a_{ij}) is a set of the sub-partition locations on shape A and (x^b_{ij}, y^b_{ij}) is that

on shape B. The estimated relationship between them is

$$\begin{pmatrix} x_{ij}^a \\ y_{ij}^a \end{pmatrix} = \begin{pmatrix} a & b \\ e & f \end{pmatrix} \begin{pmatrix} x_{ij}^b \\ y_{ij}^b \end{pmatrix} + \begin{pmatrix} c \\ d \end{pmatrix}. \tag{3.4}$$

To estimate the value of $\mathcal{T} = (a, b, c, d, e, f)$, the least-squares estimation method is used in [89]. For this method, the dissimilarity measure Ω, which denotes the sum of distances of sub-partitions, is defined as

$$\Omega = \sum_{i=1}^{n} \sum_{j=1}^{N} (\tilde{x}_{ij}^a - x_{ij}^a)^2 + (\tilde{y}_{ij}^a - y_{ij}^a)^2. \tag{3.5}$$

where (x_{ij}^a, y_{ij}^a) is the observed location on shape A, while $(\tilde{x}_{ij}^a, \tilde{y}_{ij}^a)$ is the computed value with the observed location (x_{ij}^b, y_{ij}^b) on shape B. Therefore, the computed value $(\tilde{x}_{ij}^a, \tilde{y}_{ij}^a)$ yields

$$\Omega = \sum_{i=1}^{n} \sum_{j=1}^{N} (a x_{ij}^b + b y_{ij}^b + c - x_{ij}^a)^2 + (e x_{ij}^b + f y_{ij}^b + d - y_{ij}^a)^2. \tag{3.6}$$

Then, the value of $\mathcal{T} = (a, b, c, d, e, f)$ is determined with

$$\frac{d\Omega}{d\mathcal{T}} = 0. \tag{3.7}$$

In Eq. 3.7, the relationship among the values of $\mathcal{T} = (a, b, c, d, e, f)$ is described as

$$\Pi \mathcal{T}^{\mathrm{T}} = \Phi, \tag{3.8}$$

where Π and Φ are defined as a result of the solution of Eq. 3.7 in Eq. 3.9 and Eq. 3.10, respectively.

$$\Pi = \begin{pmatrix} \sum_{i=1}^{n}\sum_{j=1}^{N} (x_{ij}^b{}^2) & \sum_{i=1}^{n}\sum_{j=1}^{N} (x_{ij}^b y_{ij}^b) & \sum_{i=1}^{n}\sum_{j=1}^{N} (x_{ij}^b) & 0 & 0 & 0 \\ \sum_{i=1}^{n}\sum_{j=1}^{N} (x_{ij}^b y_{ij}^b) & \sum_{i=1}^{n}\sum_{j=1}^{N} (y_{ij}^b{}^2) & \sum_{i=1}^{n}\sum_{j=1}^{N} (y_{ij}^b) & 0 & 0 & 0 \\ \sum_{i=1}^{n}\sum_{j=1}^{N} (x_{ij}^b) & \sum_{i=1}^{n}\sum_{j=1}^{N} (y_{ij}^b) & nN & 0 & 0 & 0 \\ 0 & 0 & 0 & nN & \sum_{i=1}^{n}\sum_{j=1}^{N} (x_{ij}^b) & \sum_{i=1}^{n}\sum_{j=1}^{N} (y_{ij}^b) \\ 0 & 0 & 0 & \sum_{i=1}^{n}\sum_{j=1}^{N} (x_{ij}^b) & \sum_{i=1}^{n}\sum_{j=1}^{N} (x_{ij}^b{}^2) & \sum_{i=1}^{n}\sum_{j=1}^{N} (x_{ij}^b y_{ij}^b) \\ 0 & 0 & 0 & \sum_{i=1}^{n}\sum_{j=1}^{N} (y_{ij}^b) & \sum_{i=1}^{n}\sum_{j=1}^{N} (x_{ij}^b y_{ij}^b) & \sum_{i=1}^{n}\sum_{j=1}^{N} (y_{ij}^b{}^2) \end{pmatrix} \tag{3.9}$$

$$\Phi = \begin{pmatrix} \sum_{i=1}^{n}\sum_{j=1}^{N} (x_{ij}^b x_{ij}^a) \\ \sum_{i=1}^{n}\sum_{j=1}^{N} (x_{ij}^a x_{ij}^b) \\ \sum_{i=1}^{n}\sum_{j=1}^{N} (x_{ij}^a) \\ \sum_{i=1}^{n}\sum_{j=1}^{N} (y_{ij}^a) \\ \sum_{i=1}^{n}\sum_{j=1}^{N} (x_{ij}^b y_{ij}^a) \\ \sum_{i=1}^{n}\sum_{j=1}^{N} (y_{ij}^b y_{ij}^a) \end{pmatrix}. \tag{3.10}$$

Therefore, the estimate of the value of $\mathcal{T}$ is

$$\mathcal{T} = \Pi^{\mathrm{T}} \Phi^{-1}. \tag{3.11}$$

Because the value of $\mathcal{T}$ is composed of six independent variables, this transform (Eq. 3.4) can solve many types of geometric transform problems, such as rotation or shear. In previous research, certain constraints have been applied to the value of $\mathcal{T}$. For example, Mokhtarian applied the relation shown in Eq. 3.12 to the transform given in Eq. 3.4 [89].

$$\begin{pmatrix} e \\ f \end{pmatrix} = \begin{pmatrix} -b \\ a \end{pmatrix}. \tag{3.12}$$

While Mokhtarian applied his method to searching model shapes which are precisely known, our research focuses on the intra-class variability. Therefore, more types of geometric transforms are conducted.

3.4.4 Parameter Distance Evaluation

The third step is for evaluating parameter distances. Since the alignment is done for all combinations, parameter distances are derived for all of them. The parameter distance $dist_{ij}$ for sub-partition p_{ij} is defined with the distance of its length dl_{ij}, curvature $d\kappa_{ij}$, location do_{ij}, and normal direction $d\theta_{ij}$ as

$$dist_{ij} = w_l dl_{ij} + w_\kappa d\kappa_{ij} + w_o do_{ij} + w_\theta d\theta_{ij}, \tag{3.13}$$

where

$$dl_{ij} = |l^a_{ij} - l^b_{ij}|/dl_{max}, \tag{3.14}$$

$$d\kappa_{ij} = |\kappa^a_{ij} l^a_{ij} - \kappa^b_{ij} l^b_{ij}|, \tag{3.15}$$

$$do_{ij} = \sqrt{(x^a_{ij} - x^b_{ij})^2 + (y^a_{ij} - y^b_{ij})^2}/do_{max}, \tag{3.16}$$

$$d\theta_{ij} = |\theta^a_{ij} - \theta^b_{ij}|, \tag{3.17}$$

where

$$dl_{max} = \max_{i,j} |l^\alpha_i - l^\alpha_{ij}|, \tag{3.18}$$

$$do_{max} = \max_{i,j} \sqrt{(x^\alpha_i - x^\alpha_{ij})^2 + (y^\alpha_i - y^\alpha_{ij})^2}, \tag{3.19}$$

where (i, j) denotes an arbitrary sub-partition and α denotes a or b. Here, w_l, w_κ, w_o, and w_θ are weight parameters for the length, curvature, location, and normal direction of each sub-partition, respectively. In this research, we set $w_l = 0.1$, $w_\kappa = 0.1$, $w_o = 0.2$, and $w_\theta = 0.2/N$. With these parameters, the parameter distance for shape A $dist_a$ is defined as

$$dist_a = \sum_{i=1}^{n} \sum_{j=1}^{N} dl_{ij} dist_{ij}/nN, \tag{3.20}$$

where the number of sub-partitions included in each curvature partition is N and that of partitions in the contour of shape A is n. While Mokhtarian applied the curvature and length of partitions for solving the partition matching problem, the parameters we apply are the length, curvature, location, and normal direction of sub-partitions [89]. In order to solve the partition matching problem for similar shapes, these four parameters are necessary as shown in Section 3.5.

3.4.5 Combination Selection

The fourth step is for best combination selection. The best combination is defined to be the one that minimizes the parameter distance. This step is shown in the fourth and fifth rows of Fig. 3.5. The best combination of similar shapes minimizes the parameter distance much more than that of non-similar ones.

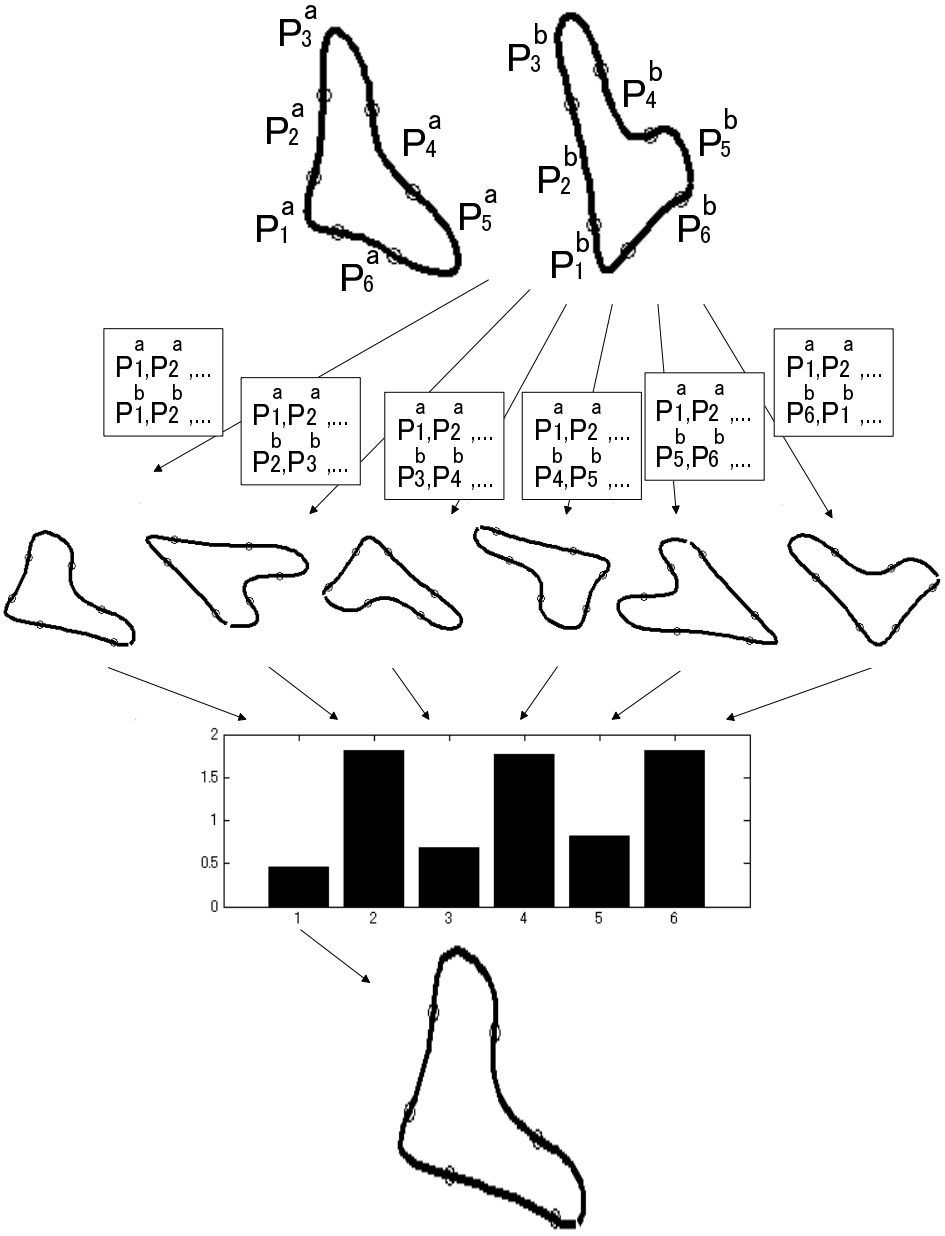


Fig. 3.5 An example of the similarity measurement algorithm. When two shapes are compared with each other, the partitions of one shape are arbitrarily numbered clockwise (first row). In this case the partitions of the left shape are numbered $P_1^a, P_2^a, \ldots, P_6^a$, and those of the right shape are numbered $P_1^b, P_2^b, \ldots, P_6^b$. Geometric transform is then performed with the matching pairs (second row). After this transform, the parameter distances are evaluated with each matching pair (third row). Finally, the optimal matching pair is found with these parameter distances. The similarity is defined to be the inverse of the parameter distance of the optimal matching pair.

3.5 Evaluation Results

In this section, we evaluate the effectiveness of our proposed method. For this evaluation, we firstly show the experimental results on a standard data set, MPEG-7 CE-Shape-1 part B data set as an example of general shape matching. We, then, show the results on geometrically deformed numeral characters written in two different font types which are applied as an example of shape matching that clearly shows the difference between our methods and previous ones. As a result of the evaluation, we show that our representation is promising for describing general shapes including the ones that contours are roughly partitioned.

First we show the results on the MPEG-7 data set [92]. MPEG-7, is widely used to test the similarity of shape representations. It consists of 1400 images, which are classified as 70 shape classes of 20 images each as shown in Fig. 3.6. Each shape in the images is compared to all other shapes, and the number of similar shapes is counted in the top 40 matches. This evaluation is called the bull's-eye test. Here, we applied a simpler version of the bull's-eye test in order to observe the tendency in the result. We constructed a data set, which consists of 350 images, which are a set of 5 images from each of 70 classes. Then, the number of similar shapes is counted in the top 10 matches. The evaluation results of the simpler version of the bull's eye test are shown in Fig. 3.7, which shows the average score of each shape image (shown in percentage). There is a clear tendency in the results, that the shape scores well when it has simple outer contour and vice versa. An example that has simple outer contour is "butterfly", which is located on the first row of the ninth column in Fig. 3.6. Since the structures of the contours of 5 images from "butterfly" have a lot in common, the similarity of them is resulted in high score. On the contrary, shapes that have complex contours do not keep intra-class invariance with our representation. An example of these shapes is "device6", which is located on the third row of the ninth column in Fig. 3.6. Although the spatial structures of the images from "device6" have a pentagon in common, the contours do not describe the spatial pattern well. Therefore, the similarity of them is resulted in low score. The total score of 70 shape classes is 73.68%, which is almost comparable to a previous representation such as CSS 75.44% [93]. Although some representations score up to 93.32% [94, 95], none of them are described with partitioned contours with geometrical parameters which the human visual system is plausibly used. This result shows that our representation is effectively applied for a general shape descriptor.

Second we show the results on geometrically deformed numeral characters written in two different font types, Arial and HGSoeiKakupoptai (see Fig. 3.8). Characters written in HGSoeiKakupoptai are geometrically deformed (translated, expanded, rotated, and sheared). We compared these 20 shapes with each other and computed their similarities. In this case, characters having similar patterns (e.g., "1" and "7" and "6" and "9") should be similarly represented because they are regarded as shapes. In order to distinguish "6" from "9", contexts such as the direction in which characters are written should be considered in addition to shape representations. These contexts are outside the scope of this evaluation, however, and the focus is on the shape similarity. Fig. 3.9 shows the results obtained in choosing the shortest distance, which corresponds to the best similarity, from among those acquired using six smoothing parameters σ simultaneously for each numerical pair. The horizontal and vertical axes in the figure denote the same numerals. In both axes, the numerals written in Arial and HGSoeiKakupoptai are arranged alternately in the order of similarity determined

by the authors. For example, since "9" is similar in shape to an inverted "6", the two numerals are arranged next to each other. Brightness is used to show parameter distances, with brighter colors corresponding to smaller distances and darker ones corresponding to better similarity. Therefore, with appropriate shape representations that describe shape features, the diagonal elements and their adjacent ones should have darker colors. From the figure it can be seen that our proposed representation expresses similar shapes "similarly" since the diagonal elements and their adjacent ones have darker colors. The results we obtained also indicate that one particular parameter is not sufficient to represent similar shapes with similar representations and that the four parameters we applied must be used to achieve accurate shape representations.

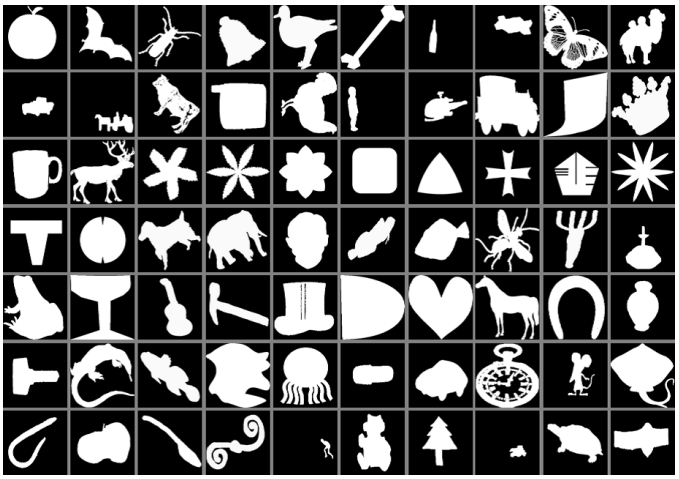


Fig. 3.6 MPEG-7 CE-Shape-1 part B data set is shown. One of the shapes each of which is chosen from each of 70 classes is shown.

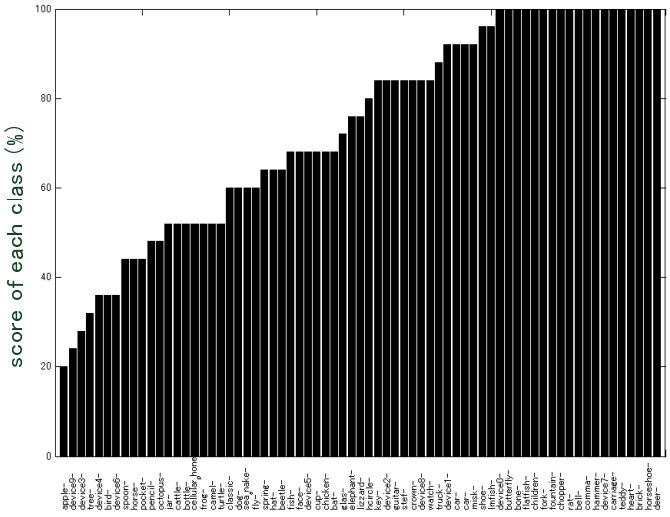


Fig. 3.7 The average score for each shape (shown in percentage). The tendency of the scores is that the shape scores well when it has simple outer contour and vice versa.

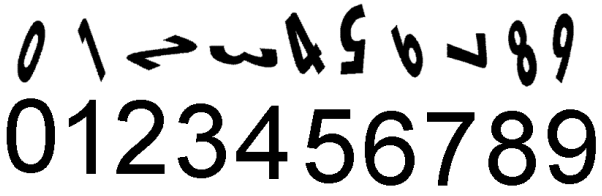


Fig. 3.8 Sample numerals. The top-row shapes are geometrically deformed numeral characters 0-9 written in HGSoeiKakupoptai. The bottom-row shapes are the same characters written in ARIAL.

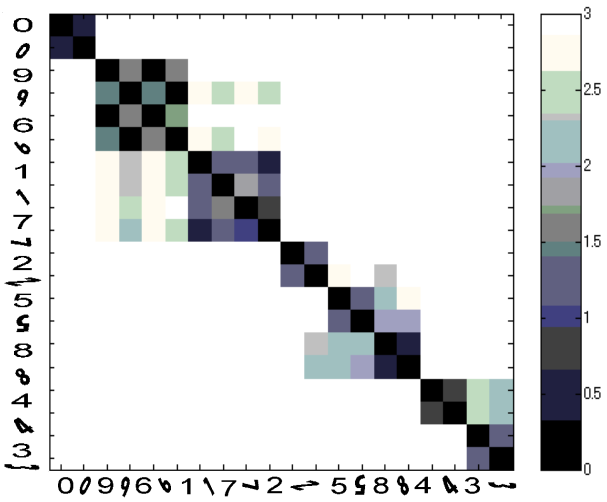


Fig. 3.9 Distance matrix is shown. Brightness is used to show parameter distances (only the 0-3 range is shown). The horizontal and vertical axes denote the same numerals. In both axes, the numerals written in Arial and HGSoeiKakupoptai are arranged alternately. The results obtained show that our approach expresses numerals with similar shapes (such as "1" and "7") with similar representations.

3.6 Discussion and Conclusion

In the above sections of this chapter, we proposed a new computational model for representing general shapes that plausibly the human visual pathway describes. We, then, showed that our model generally represents shapes keeping intra-class invariance using a simpler version of Bull's eye test with MPEG-7 CE-Shape-1 part B data set, which is widely used to test the similarity of shape representations, and numeral recognition test, which is used to show the difference between our methods and previous ones clearly. In this section, we review the progress of and the future work needed for our approach.

In order to confirm the progress we have made with our approach, we review the three constraints required for the shape representation of the visual system. The first constraint, which points out the representation with a group of partitioned contours, was satisfied with our two-step partitioning approach. With our approach, shape contours were basically partitioned with the curvature zero-crossing points for the first step. Each partitioned contour was, then, partitioned equally with the curve length for the second step. The second constraint, which defines coarse and fine structures of the shapes, was satisfied with the closed contour at varying levels of detail that CSS applies. The third constraint, which supports the invariance of the shape recognition, was satisfied with our geometric parameter and geometric transform with the partitioned contour alignment.

Conventional contour representations, on the other hand, have either of the three difference with the representation that the human visual system plausibly describes.

1. The contour partitions are not independent, so the whole shape cannot be retrieved.
2. The contour is not partitioned, so the template matching cannot be done with the representations.
3. The contour partitions do not keep sufficient information to reconstruct the original shapes.

Examples that have the first difference are CSS [84], and a partition based on the center of the shape [94]. CSS requires the full length of the shape contour, which is dependent on the whole shape. The partition based on the center of shapes is also dependent on the whole shape. Examples that have the second difference are representations that are not partitioned [90,91], and those whose partitions are not reliable so that the template matching cannot be applied [96]. Although applied for matching occluded shapes, non-partitioned representations cannot be used for template matching. Unreliable partitions also require matching methods, such as relaxation, besides template matching. Examples that have the last difference are, as shown in Fig. 3.10, the CSS and the curvature [90, 97] of "2" and "5" written in two font types. As the figure shows, the CSS representations of "2" and "5" are quite similar; this is because the information between the curvature zero-crossing points is defective. These examples show that partitioned contours represented with CSS do not keep the features that the original shapes hold. Besides, The curvature representations of them are also quite similar. These examples show that the curvature do not always keep information sufficient for pattern matching, and geometric parameters are required.

Finally, we discuss future work to be done with respect to our approach. We assume that our proposed representation can potentially be applied for retrieving occluded shapes. Since occluded shapes have not yet represented with partitioned contours

with geometric parameters, they are not retrieved with template matching. This being the case, applying our representation to occluded shapes is a subject for future work.

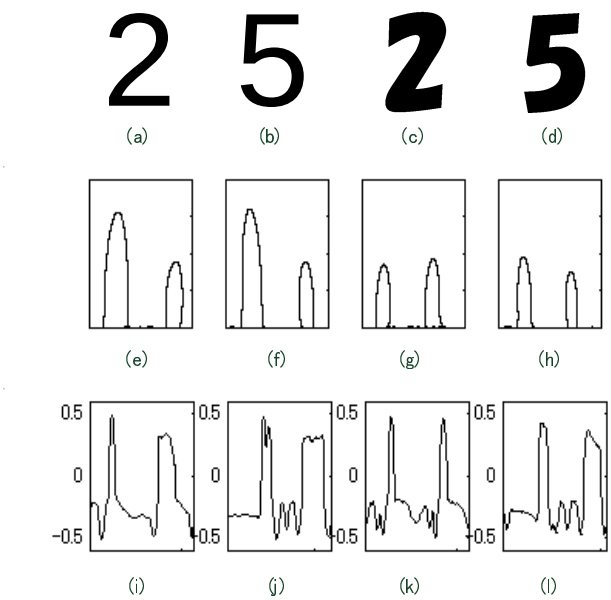


Fig. 3.10 An example of why CSS is not always suitable for recognizing shapes with a small number of the curvature zero-crossing points. (a)-(d) are original shapes, (e)-(h) are the CSS representations of their contours, where the x-axis shows sample point locations on contours and the y-axis shows the scale (see sub-section 3.3.1 and (i)-(l) are the curvature profile on their contours. (a)-(b) are "2" and (c)-(d) are "5" written in Arial and HGSoeiKakupoptai, respectively. It is clear that the CSS representations, whose images describe the shape features, make it difficult to distinguish "2"s from "5"s. So do the curvature representataions.

Chapter 4

Shape Mapping (Shape Retrieval) Shape retrieval with geometrically characterized contour partitions

4.1 Introduction

The real world is an unpredictably and dynamically changing environment. We, as living things, create certainty from uncertainty in order to adapt to such uncertainties. The human visual system always encounters uncertainty, since most visual shapes are incomplete. This incompleteness concerning visual shapes derives from (1) occlusion, (2) geometric distortion, and (3) differences in the resolution of images. However, even if occlusions and distortions such as expansion or rotation occur in shapes with low resolution, humans can still predict the original shapes by using their knowledge and memory. With the ability to predict complete shapes with the incomplete information, we detect and classify objects such as animals, plants, and landmarks [91,98,99]. Complete shapes must be retrieved from incompletely described shapes affected by occlusion, geometric distortion, and differences in image resolution. Such a retrieval will be successful if the shape representations appropriately describe the incompletely observed shapes and if the observed shapes are appropriately mapped to shapes in memory (the point correspondences are appropriately solved). Thereby, the incompletely observed shapes can be appropriately matched to shapes in memory and it can be decided if they are alike or unlike. For this reason, the task of shape retrieval will require appropriate shape-representation and shape-mapping methods.

We assume that the shape retrieval methods should appropriately normalize the shape representations in the mapping process. Shape representations can easily be normalized when the representations are based on completely described whole shapes. In such situations, there are easy normalizations using geometric transformations based on the centroid [94, 100–103] or the whole contour of the shape [84–87]. Here, completely observed shapes are described with closed contours, whereas shapes affected by occlusion, geometric distortion, and differences in image resolution are shown as roughly described open contours. The previous approaches have tried to reconstruct whole shapes from incompletely described ones. For example, some methods use T-shaped intersections, simplicity of the hidden figure, symmetry, or good continuations as "cues" for reconstructing shapes [7, 13, 104–107]. Although these completion mechanisms are innate in our human visual system, they are still sometimes inadequate for representations of incompletely described shapes since incompletely observed shapes do not always have enough "cues" for our visual system to recon-

struct complete shapes. Moreover, a normalization for representations of incomplete shapes observed under unpredictable conditions such as when occlusion, geometric distortion, and differences in resolution of images occur simultaneously has not yet been established.

Hierarchical processing models for shape retrieval have been proposed based on hierarchical neural network models [7,15,73–75,105–109]. These models rely on the fact that the shape representation is formed through a visual ventral pathway that hierarchically integrates the stimuli caught by the retinal cells by using cortical cells which have receptive fields of different sizes [1,3,10] [2,5,6,110–114]. These neural network models, however, do not accomplish suitable normalizations for the constructed shape representations. Fukushima's model and Poggio's model, for example, reconstruct the neuron's hierarchical receptive fields [7, 73, 74, 106]. Using these neurons, they hierarchically integrate the pixels of input images, and they recognize the pattern of the images with the neurons of the deepest layer. Although these models reconstruct the hierarchical pattern of the neural network well, they do not include any normalization process for the shape representations. Besides these methods, Grossberg's model exploits both global and local structures of images [75]. Global and local structures are keys to the normalization since locally described structures can be normalized with local information. In these neural network models, however, the geometric relationships of the integrated features are fixed, and they do not have normalizations for the shape representations as the result of a geometric transformation.

On the other hand, there are many computational shape retrieval methods [83, 97, 115–122]. These can roughly be classified into two streams, one of which uses "shortest path searches" for finding the correspondences between a series of points on the contours of partial shapes and those of the whole shape [98, 123–125] [126–129], the other of which uses "geometric consistency checks" for finding the correspondences between feature points on the partial shapes and those of the whole shape [80, 130–135]. Using the order of the points or the geometric relationship among the feature points, these methods excel in finding the point correspondences (the point-to-point mapping) between two shapes. However, the previous studies only focused on solving the point correspondences; they did not deal with the normalization for the shape representation. Unless the features of the shapes are well normalized, the similarity of the corresponding points cannot be determined even if the shortest path is found. Moreover, unless the features of the shapes are well extracted, the geometric relationship between two shapes would be vague even if they are geometrically consistent. For example, the curvature representation is widely used to describe a series of contours on shapes [7, 84, 88, 90, 97, 105, 124, 127, 136]. Curvature itself, however, varies when expansion occurs, since it is defined as the rate of change of angle per unit length. It is true that the curvature representation can be normalized if the whole body is completely observed. However, the curve length cannot be used as the normalization parameter if it is not known whether the observed shape is the whole body or part of one. Although the previous studies discussed this problem [90, 124], none of them describe a normalization of the shape description.

In this chapter we propose a shape retrieval method using a "curvature partition" shape representation and a shape mapping including a normalization using an "angle-length profile". The curvature partition, which is composed of geometrically characterized contours partitioned by zero-crossing points on the contours with coarse and fine shapes, describes general shapes well [22]. Although the curvature partition excels at representing shapes, the previous methods do not have a mapping process including a normalization that works when there are occlusions, geometric distortions, and differences in image resolution. Here, we describe such a normalization process

using the angle-length profile. By exploiting the fact that the angle variation is Euclidean invariant even under occlusions, we construct an angle-length profile as a basis for the normalization of the curvature partition. As a result, unlike the previously proposed methods, our method can be applied in situations with occlusion, geometric distortion, and differences in image resolution. Section 4.2-4.4 describes the three main steps to our approach (see Fig. 4.1). Briefly, in the first step, the feature points are extracted as the bases for the point-to-point mapping on the shapes (Section 4.2). In the second step, the point-to-point mapping is done with the angle-length profile and the sizes of the two shapes are normalized (Section 4.3). In the final step, the normalized shapes are represented with the curvature partition and the shapes are matched (Section 4.4). Section 4.5 presents experimental results, and Section 4.6 discusses them and concludes the chapter. Appendix A presents empirical studies showing how our method performs under noisy conditions and with affine transforms. Appendix B presents experimental results for typical previous approaches, i.e., dynamic programming and geometric hashing, to serve as benchmarks of the evaluation of our method.

4.2 Feature Point Extraction

This section explains how feature points are extracted on the contours of shapes. Let us suppose that shape A, a complete shape in memory, is matched to shape B, an incompletely observed shape. Shape A can be described with a closed contour (the whole shape), whereas shape B can be described with an open contour (a partial shape), as shown in Figs. 4.1(a-1) and 4.1(b-1). We use both the zero-crossing points on lines of curvature and the curvature peak points on contours on multiple scales for smoothing the feature points. The feature point extraction consists of two steps.

1. Construction of smoothed contours on multiple scales
2. Extraction of feature points on multiple scales

The smoothed contours are constructed on multiple scales on the basis of the idea of curvature partitions [22]. When convolved with a one-dimensional Gaussian kernel $g(t, \sigma)$ of width σ, a contour C evolves into a smoother contour $C(x(t, \sigma), y(t, \sigma))$, where $x(t, \sigma)$ and $y(t, \sigma)$ are $x(t)$ and $y(t)$ convolved with $g(t, \sigma)$. This convolution is done for closed and open contours after the contours for the open curves are completed with curvature approximation.

The feature points are extracted on multiple scales, as shown in Fig. 4.2. These are the zero-crossing points and the peak points on lines of curvature on the contours. The curvature, $\kappa(t, \sigma)$, for each sample point on the contours on multiple scales is expressed as

$$\kappa(t, \sigma) = \frac{\dot{x}\ddot{y} - \dot{y}\ddot{x}}{(\dot{x}^2 + \dot{y}^2)^{3/2}}, \tag{4.1}$$

where

$$\dot{x} = \frac{dx}{dt}, \dot{y} = \frac{dy}{dt}, \ddot{x} = \frac{d^2x}{dt^2}, \ddot{y} = \frac{d^2y}{dt^2}, \tag{4.2}$$

where x and y are abbreviations of $x(t, \sigma)$ and $y(t, \sigma)$. Then, the zero-crossing points and the peak points on $\kappa(t, \sigma)$ are found. Although the zero-crossing point is invariant to affine transformations [86], it cannot be detected in shapes without concave/convex areas. For this reason, we use both the zero-crossing points and peak points on the curvatures on the contours as feature points.

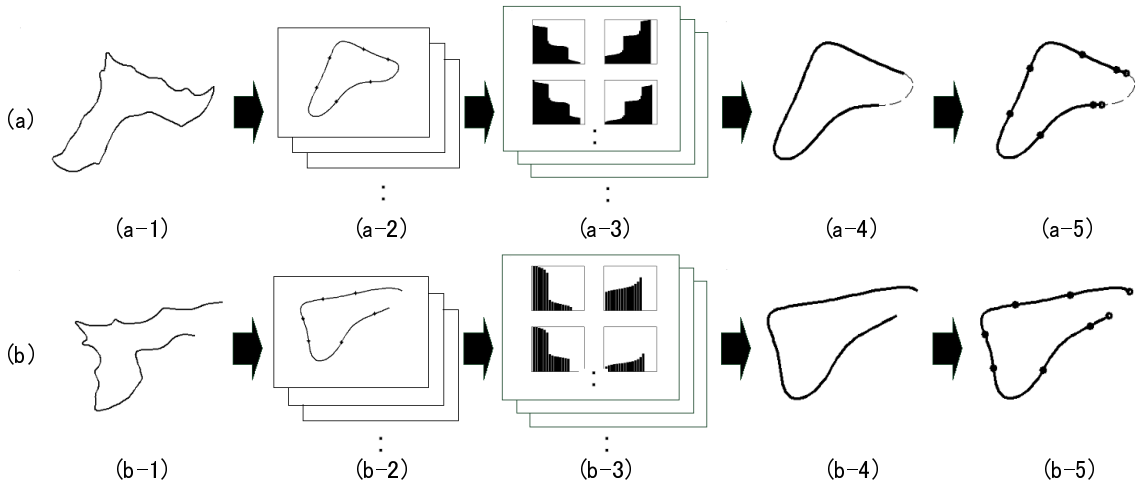


Fig. 4.1 Our approach to shape retrieval. Here, shape A, which is a complete shape in memory is matched to shape B, which is an incompletely observed shape. Shape A is described with a closed contour (a), and shape B is described with an open contour (b). The closed (a) and open contour (b) are specified in (a-1) and (b-1), respectively. In order to find the feature points, the closed (a) and open contour (b) are smoothed in (a-2) and (b-2) on multiple scales ($\sigma = 2, 4, 8, 16, 32$, and 64). Then, their feature points are extracted (zero-crossing points on lines of curvature are shown for simplicity). In order to find the point correspondences between the shapes, the angle-length profiles are constructed in (a-3) and (b-3) for both sides (left/right sides) of each feature point. The angle-length profiles, the closed (a-3) and the open contour (b-3), and the points on the closed contour corresponding to that on the open contour are found. Then, the two shapes are normalized on the basis of the correspondences. The normalized open contour (b-4) and the contour area (points) on the original closed contour corresponding to the open contour (a-4) are presented. After the curvature partitions are constructed as shown in (a-5) and (b-5), the two shapes are finally matched by comparing the curvature partitions for both shapes.

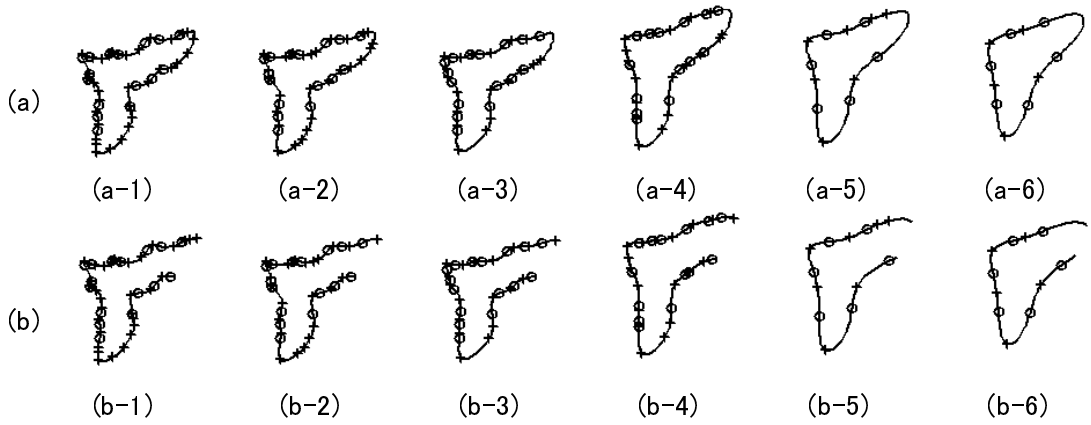


Fig. 4.2 Convolved contours and feature points on multiple scales. (a) is a closed contour (shape A), and (b) is an open contour (shape B). (a-1)-(a-6) and (b-1)-(b-6) correspond to contours convolved with $\sigma = 2, 4, 8, 16, 32$, and 64. Curvature zero-crossing points are indicated as circles, and curvature peak points are indicated as crosses.

4.3 Angle-Length Profiles

This section explains how to normalize the size of shapes A and B by comparing their angle-length profiles and how to find the correspondences between them. Angle-length profiles are constructed from the viewpoint of the feature points extracted with the procedure described in Section 4.2. These profiles describe the change in the curve length per unit of angle from the viewpoint of the feature points. These profiles correspond to those in Fig. 4.1(a-3)(b-3). After the sizes of shapes A and B are normalized, the correspondences are found. This process roughly consists of three steps.

1. Construction of angle-length profiles
2. Derivation of normalization parameter
3. Detection of the corresponding areas (points)

4.3.1 Construction of Angle-Length Profiles

An angle-length profile is constructed for each feature point on shapes A and B. The angle-length profile (AL) is defined as the change in curve length per unit angle from the viewpoint of a feature point. The angular unit is denoted as ϵ (ϵ is set to 10 degrees in this chapter). To construct AL, the contour vector, C, of the original or the smoothed contour of shapes A and B, is resampled with an angle-based contour vector (AC). Each element of AC is the curve length from the initial point to a point at which the difference in tangents from that of the initial point is a certain value. The initial point of AC is defined as a feature point of a contour of each of shapes A and B. AC is described with a vector Z defined as

$$Z = (\zeta(\delta\theta = \omega), \zeta(\delta\theta = 2\omega), \ldots, \zeta(\delta\theta = m\omega)), \tag{4.3}$$

where ω denotes a unit difference in tangents, $\zeta(\delta\theta)$ denotes the curve length from the initial point to the point at which angle difference from the initial point is $\delta\theta$, and

$$m = \lceil \delta\theta_{max} \rceil, \tag{4.4}$$

where $\lceil x \rceil$ denotes the ceiling function for x, and $\delta\theta_{max}$ denotes the difference in tangents to the end point on an open contour. On a closed contour, on the other hand, $\delta\theta_{max}$ denotes 360 degrees. AC is defined on both the left and right of the initial point. Examples of angle-based contour vector (AC) are given in Figs. 4.3, 4.4(i), and 4.5(i). The angle-length profile (AL) is constructed by referring to AC. AL is a vector Ξ defined as

$$\Xi = (\xi(1), \xi(2), \ldots, \xi(\mu)), \tag{4.5}$$

where

$$\xi(i) = \zeta(i\epsilon), \tag{4.6}$$

and

$$\mu = \lceil \delta\theta_{max}/\epsilon \rceil. \tag{4.7}$$

Fig. 4.4(ii) and Fig. 4.5(ii) show examples of angle-length profile ALs.

4.3.2 Derivation of Normalization Parameter

A normalization parameter is derived for each feature point pair on shapes A and B by comparing the angle-length profiles (ALs) of shapes A and B. Supposing that shape A is described with a closed contour and shape B is described with an open contour, their ALs are expressed as

$$\Xi^A = \{\xi(1), \xi(2), \ldots, \xi(\mu^A)\}, \tag{4.8}$$

and

$$\Xi^B = \{\xi(1), \xi(2), \ldots, \xi(\mu^B)\}, \tag{4.9}$$

where Ξ^A and Ξ^B are vectors representing the ALs of shapes A and B, and μ^A and μ^B are derived using Eq. 4.7. In this case, it is assured that

$$\mu^B \leq \mu^A. \tag{4.10}$$

Therefore, the closed contour (shape A) is converted into an open contour if the number of elements of its AL is limited to μ^B. The corresponding AL is

$$\Xi^A = \{\xi(1), \xi(2), \ldots, \xi(\mu^B)\}. \tag{4.11}$$

Fig. 4.5 illustrates the process of finding the correspondences between shapes A and B. Now that two ALs are described with the same number of elements, the normalization parameter can simply be derived as the average expansion r of the elements of ALs.

$$r = \sum_{i=1}^{n^B} \frac{\xi^A(i)}{\xi^B(i)}. \tag{4.12}$$

Finally, we use the normalization parameter r to define the certainty parameter c that reflects the level of certainty about the feature point pair used to derive the normalization parameter.

$$c = -\sum_{i=1}^{\mu^B} |\xi^A(i) - r\xi^B(i)|. \tag{4.13}$$

The best certainty parameter c is selected from those of all pairs of feature points of shapes A and B.

4.3.3 Detection of the Corresponding Areas

The process described in the last sub-section derives a certainty parameter c for each pair of feature points of shapes A and B. Then, the best certainty parameter is selected from those of all pairs of feature points of shapes A and B. Assume that (i_A, i_B) are a pair of feature points of shapes A and B. Here, i_A denotes the i_Ath feature points on shape A, and i_B denotes the i_Bth feature points on shape B. The best pair (i_A^*, i_B^*) is derived from

$$\begin{aligned} (i_A^*, i_B^*) &= \arg\max_{i_A, i_B} c_{i_A, i_B}, \\ for \;\; i_A &= 1, 2, \ldots, M_A, \\ i_B &= 1, 2, \ldots, M_B, \end{aligned} \tag{4.14}$$

where shape A has M_A feature points, shape B has M_B feature points, and c_{i_A,i_B} denotes the certainty parameter c derived with the pair (i_A, i_B). As a result, the normalization parameter r^* for shapes A and B is the one derived for the best feature point pair (i_A^*, i_B^*) by using Eq. 4.12. Similarly, the contour area on shape A (closed contour) corresponding to the contour of shape B (open contour) can be derived using the best feature point pair (i_A^*, i_B^*), as a result of the AL conversion shown with Eq. 4.11. The open contour resulting from this process is shown in Fig. 4.6.

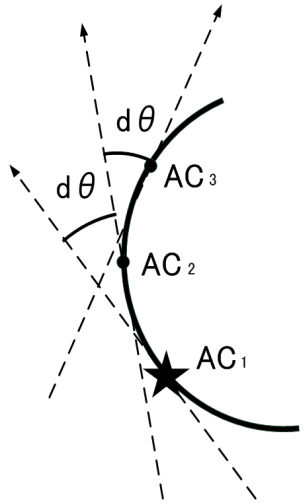


Fig. 4.3 Angle-based contour vector (AC). An AC on the left of a feature point (indicated by the star) on the contour of the shape and the elements of AC (AC_1, AC_2, and AC_3) are shown. The tangent difference from that of the initial element is an integral multiple of angular units $d\theta$.

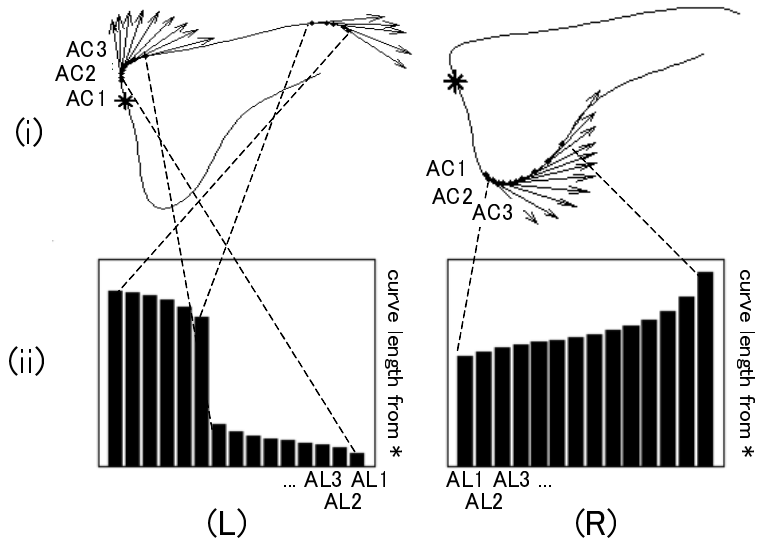


Fig. 4.4 Angle-based contour vectors and angle-length profiles. The angle-based contour vectors (AC) on the left (L) and right (R) of a feature point (indicated by the asterisk) on an open contour and the tangents (indicated by the arrows) are shown in (i). The angle-length profiles (AL) on each side of the feature point ($*$) are shown in (ii). The x-axis indicates the order of AC, and the y-axis indicates the curve length from the feature point.

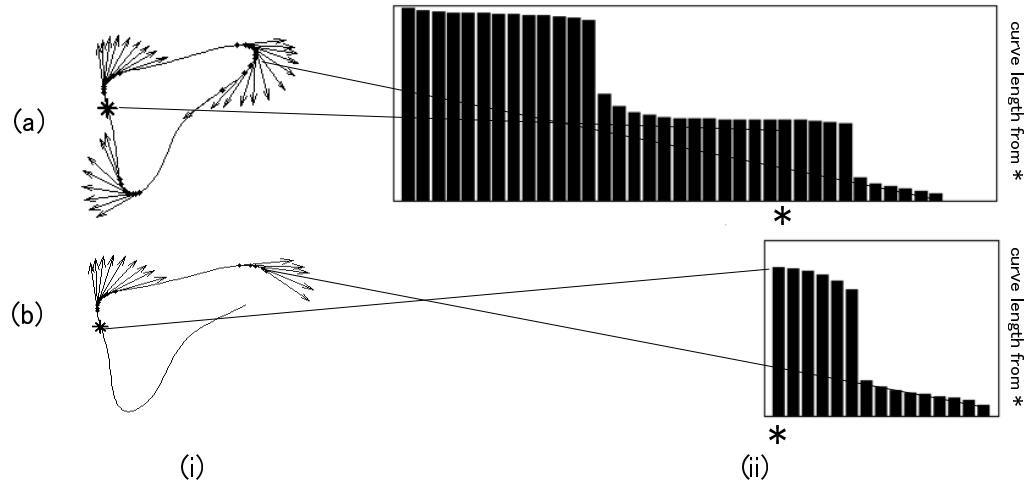


Fig. 4.5 Comparison of angle-length profiles of closed and open contours. In (i), the angle-length profiles on the left of a feature point (indicated by an asterisk on each contour) in (ii) are shown for a closed contour (a) and an open contour (b). Because angle-length profiles of closed and open contours have different number of elements, this comparison is done after the same numbers of elements are selected. The number is the minimum of the two contours.

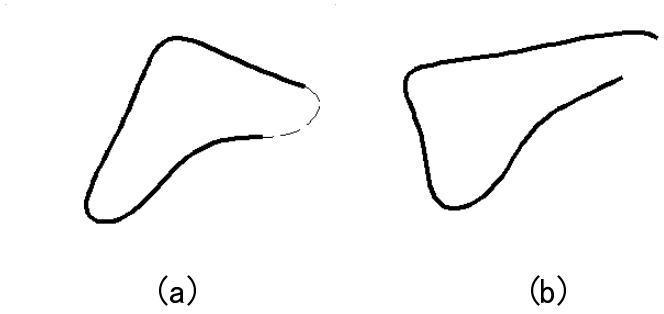


Fig. 4.6 Area enclosed by a closed contour (a) corresponding to the area of an open contour (b). In (a), the original contour is indicated by the dashed line, and the area enclosed by the original contour corresponding to the open contour (b) is indicated by the bold line.

4.4 Curvature Partitions

This section explains how two shapes are represented and matched in order to accomplish the final step of our shape retrieval. The shape representation is obtained with the process described in the last section. The shapes represented with the curvature partition (explained in Chapter 3) are compared in order to calculate the similarity. The shape representation and matching consists of two steps.

1. Construction of curvature partitions (Sub-section 4.4.1)
2. Evaluation of parameter distance with geometric transformation (Sub-section 4.4.2)

Because this matching is done with all partitions, the edges of which are the feature points or the edges of a contour, it is only done for open contour pairs having the same numbers of feature points.

4.4.1 Construction of Curvature Partitions

Curvature partitions are constructed as partial curves whose edges are the feature points or the edges of contour C on multiple scales of smoothing. They are expressed as $P_i(i = 1, 2, \ldots, n)$, where n is the number of curvature partitions that the contour has. Once they are constructed, they are divided into N sub-partitions. Each sub-partition p_{ij} is expressed in terms of its average curvature κ_{ij}, length l_{ij}, normal direction θ_{ij}, and location $o(x_{ij}, y_{ij})$:

$$p_{ij} = \begin{pmatrix} \kappa_{ij} \\ l_{ij} \\ \theta_{ij} \\ x_{ij} \\ y_{ij} \end{pmatrix}, \tag{4.15}$$

where $j = 1, 2, \ldots, N$. Therefore, curvature partition P_i has $N \times 5$ elements:

$$P_i = \begin{pmatrix} \kappa_{i1} & \kappa_{i2} & \ldots & \kappa_{iN} \\ l_{i1} & l_{i2} & \ldots & l_{iN} \\ \theta_{i1} & \theta_{i2} & \ldots & \theta_{iN} \\ x_{i1} & x_{i2} & \ldots & x_{iN} \\ y_{i1} & y_{i2} & \ldots & y_{iN} \end{pmatrix}. \tag{4.16}$$

Each curve on a contour (which is a partial curve whose edges are both feature points) is expressed by its curvature, length, normal direction, and location. The curvature partitions, (P_1, P_2, ..., P_n), on multiple scales of smoothing are stored as representations of the shape.

4.4.2 Evaluation of Parameter Distance

The curvature partitions of shapes A and B are compared with each other on multiple scales in order to derive similarities in the manner described in [22]. The geometric transformation is done by estimating the relationship between shapes A and B,

$$\begin{pmatrix} x^a \\ y^a \end{pmatrix} = \begin{pmatrix} a & b \\ e & f \end{pmatrix} \begin{pmatrix} x^b \\ y^b \end{pmatrix} + \begin{pmatrix} c \\ d \end{pmatrix}, \tag{4.17}$$

where (x^a, y^a) is a set of the sub-partition locations on shape A and (x^b, y^b) is that on shape B. The values of a, b, c, d, e, f, are estimated with the least-squares estimation method. Each parameter of curvature partition P_i is deformed using Eq. (4.17).

Finally, the similarity between the two shapes is taken to be the smallest parameter distance on the scales. The parameter distance $dist_{ij}$ for sub-partition p_{ij} is defined with its length dl_{ij}, curvature $d\kappa_{ij}$, location do_{ij}, and normal direction $d\theta_{ij}$ as follows:

$$dist_{ij} = w_l dl_{ij} + w_\kappa d\kappa_{ij} + w_o do_{ij} + w_\theta d\theta_{ij}, \tag{4.18}$$

where w_l, w_κ, w_o, and w_θ are weight parameters for the length, curvature, location, and normal direction of each sub-partition, respectively. Here, we apply $w_l = 0.1$, $w_\kappa = 0.1$, $w_o = 0.2$, and $w_\theta = 0.2/N$ in the manner described by [22]. After the average $dist_{ij}$ is derived for each scale, the smallest average value is chosen as the similarity between the two shapes.

4.5 Results of Evaluation

This section evaluates the effectiveness of our method by presenting the experimental results on an open contour classification test as an example of shape retrieval. The results show that our method, which includes the normalization process, effectively retrieves shapes from open contours especially under unpredictable conditions in which occlusion, geometric distortion, and differences in image resolution occur simultaneously.

Our classification test is based on the standard MPEG-7 CE-Shape-1 part B data set. This data set is widely used to test the similarities in shape representations [92]. It consists of 1400 images classified into 70 shape classes of 20 images each, as shown in Fig. 4.7. Although they do not have occlusion, the classes of this data set include quite a few rotated, sheared, and expanded examples. For this reason, this data set was used to investigate the classification of shapes with differences in resolution as well as geometric distortion. However, as it does not include any examples of occluded shapes, we constructed a data set of open contours from it, which consisted of 350 images, i.e., five images from each of the 70 classes. The open contours in our data set were of a certain ratio $(99\%, 90\%, 80\%, 70\%, 60\%, 50\%)$ of the length of the original closed contours. Fig. 4.8 shows some examples of these open contours. We supposed that the shapes would be correctly retrieved if the open contours were effectively classified on the basis of closed contours. For this evaluation, the number of similar open contours was counted for each closed contour in the top 10 matches. The open contours belonging to the same class as those of the closed contour included in the top 10 matches were counted as a score showing the effectiveness of the shape retrieval. This sort of evaluation is called a bull's-eye test.

The results of the evaluation for contour length ratios (%) of $99\% - 50\%$ are presented in Fig. 4.9, where the average scores of all image shapes are indicated as percentages. The average scores of the 70 shape classes are 71.94% (ratio of 99%), 70.29% (ratio of 90%), 59.83% (ratio of 80%), 44.69% (ratio of 70%), 31.03% (ratio of 60%), and 11.66% (ratio of 50%). Fig. 4.10 compares these results with those of promising previous approaches. As shown in this figure, our method classifies open contours more accurately for the contours whose length ratios are more than 50 % of the whole contour, whereas the previous approaches classify them more accurately for open contours equal to 50 % of the whole contour.

Our experimental results do not show that our method perfectly retrieves shapes but rather that it is suitable for retrieving shapes under unpredictable conditions. We believe that this is the effect of the normalization process. We can better explain the effect of the normalization process by examining the difference between the results for the 99% and 100% ratios of contour length. Although some of the previous methods are effective at classifying shapes, some of whose scores in the bull's eye test with MPEG-7 CE-Shape-1 part B data set are up to 93.32% [94, 137], the scores of our experiments with dynamic programming and geometric hashing are respectively 31.26% and 25.60% for the contour length ratio of 99%. This is caused by the effect of the normalization which much more significantly affects occluded contours than closed contours. Needless to say, the sizes are easily normalized for closed contours since the centroids or the whole length of the contours are used as the bases. However, the contour size cannot be deterministically normalized for a contour that does not completely show the whole shape since it is not predetermined what percentage of the whole shape is shown. Even contours with 99% contour length ratios are not

exceptions, since that ratio cannot be determined beforehand. Nevertheless, the score 71.94% with our method for the 99% contour length ratio is almost the same score as 73.68% for the closed contour (100% contour length ratio) with the curvature partition [22]. This result shows that our normalization process works well on incomplete open contours.

In addition, the scores for the 99% and 60% contour length ratios plausibly show the effect of the normalization process. It is true that the lower the ratio goes, the more the score decreases. Because open contours with smaller ratios have fewer features, it is more difficult for the human visual system to classify shorter contours. The computational models are not exceptions, and all of the scores of our models and previous methods are correlated to the ratio of the contour length. Nonetheless, Fig. 4.10 clearly shows that the scores of our method are much better than those of the promising previous methods. These results are plausible evidence of the suitability of our normalization process.

As a supplement, let us explain why the scores of our method for the 50% contour length ratio are less than those of the previous methods, as shown in Fig. 4.10. As we can see in Fig. 4.8(f), it is usually quite difficult to retrieve the original shapes from open contours with such a small ratio. However, the open contour data set for our bull's eye test is composed of classes, each of which includes five open contours. Because each open contour is classified with a closed contour, one of the five open contours is classified with exactly the original closed contour. When an open contour is matched and classified with the original closed contour, the normalization process is not necessary since their sizes are exactly the same. On the other hand, while neither the "curvature partition" shape representation nor "the angle-length profile" shape mapping is suitable for exactly representing and matching shapes, both are suitable for roughly representing and matching shapes in the same class. For this reason, the scores of our method for the 50% contour length ratio are less than those of the previous methods.

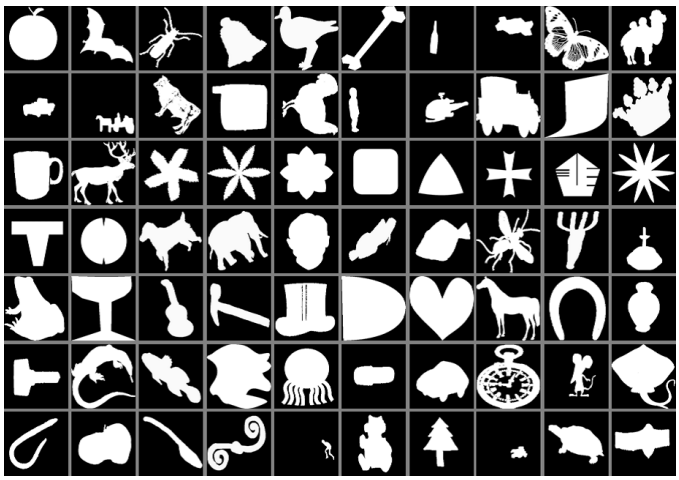
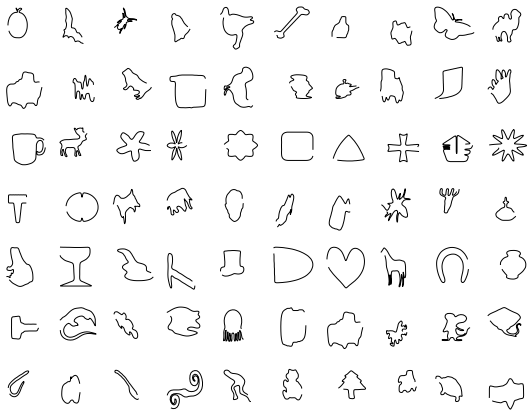
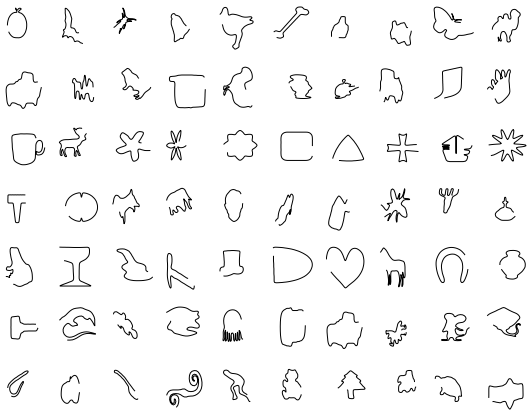


Fig. 4.7 MPEG-7 CE-Shape-1 part B data set. Shapes were chosen from each of 70 classes.



(a)

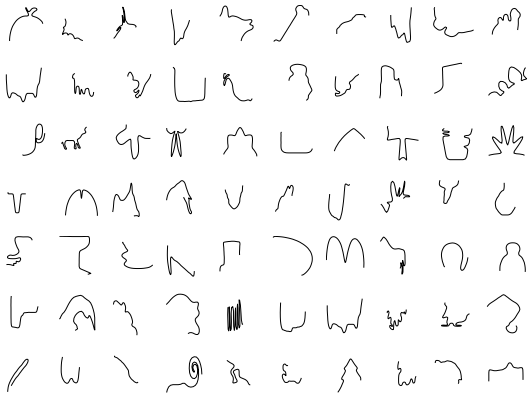


(b)

(c)

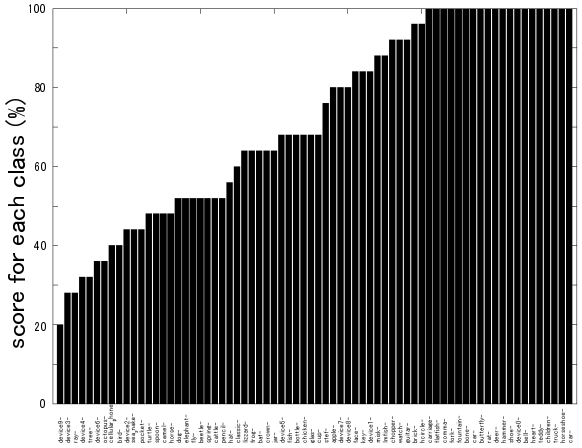
(d)

(e)

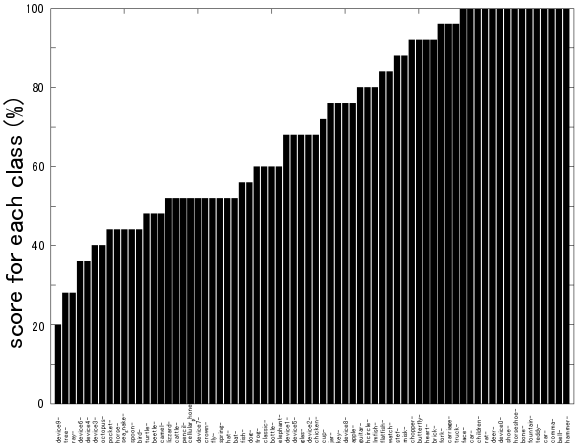


(f)

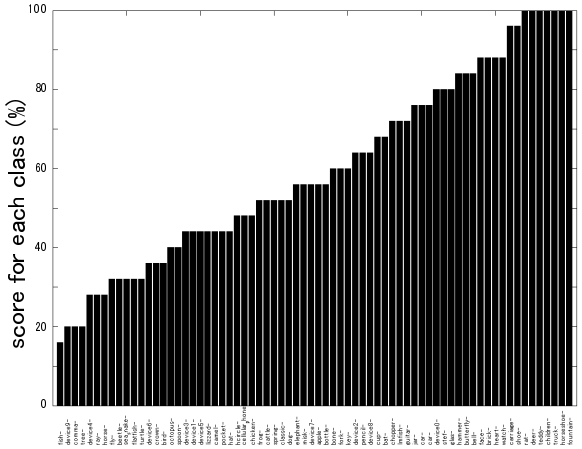
Fig. 4.8 Examples of open contours constructed with the MPEG-7 CE-Shape-1 part B data set shown in Fig. 4.7. The open contours are certain ratios $(99\%, 90\%, 80\%, 70\%, 60\%, 50\%)$ of the length of the closed contours of the original shapes in the data set. Examples of open contours for the ratios of $99\%, 90\%, 80\%, 70\%, 60\%, and 50\%$ are shown in (a)-(e).



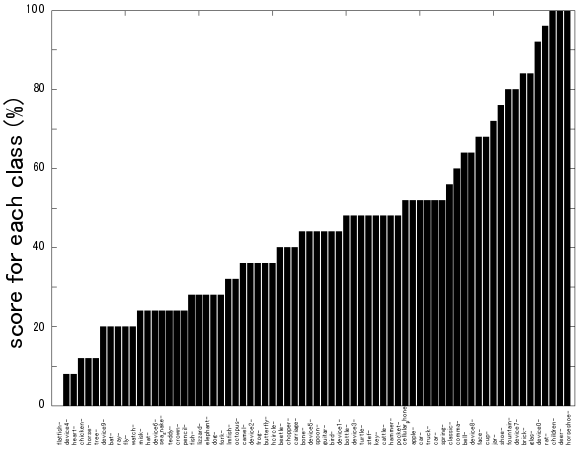
(a)



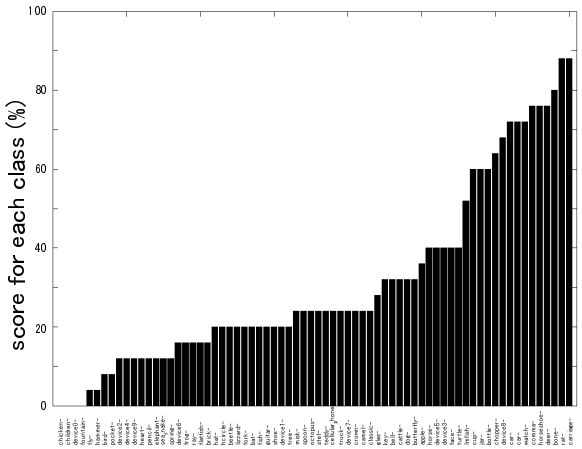
(b)



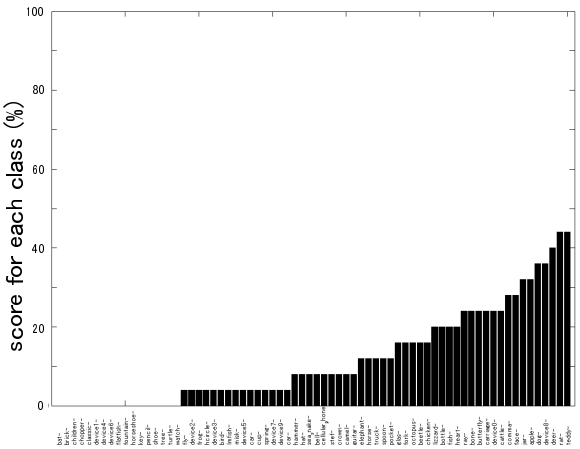
(c)



(d)



(e)



(f)

Fig. 4.9 Average scores for each class (as percentages) for ratios of (a) 99%, (*b*)90%(*c*)80%, (*d*)70%, (*e*)60%, and (f) 50%. The average scores of 70 shape classes are 71.94, 70.28, 59.83, 44.69, 31.03, and 11.66% for ratios of 99%, 90%, 80%, 70%, 60%, and 50%.

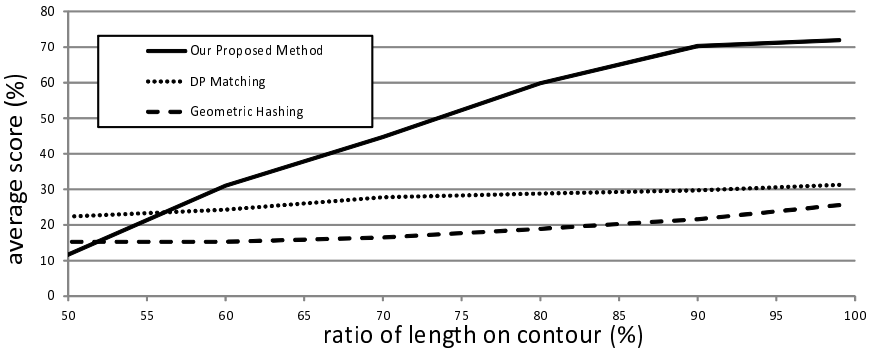


Fig. 4.10 Comparison of our method and typical previous methods. The average scores of 70 shapes for contour length ratios from 99% to 50% are shown. The bold line shows the results for our method, the dashed line those for dynamic programming, and the dotted line those for geometric hashing.

4.6 Discussion and Conclusion

We described a new method of computation for retrieving shapes under the conditions of occlusion, geometric distortion, and differences in resolution. We demonstrated that our method effectively retrieves shapes through experiments on an open contour classification test. This section reviews the progress we made with our approach.

Let us review the conditions that cause shapes to appear incomplete. They are occlusion, geometric distortion, and differences in the resolution of images. Our method deals with occlusion by conducting normalization on the shape representation with the angle-length profile. This normalization can be stably conducted because the angle-length profile is Euclidean invariant even under occlusion. The angle-length profile slightly varies as the base contour is smoothed through the smoothing process shown in Section 4.2. In this smoothing process, the variations are similar for two angle-length profiles of similar shape, as can be checked with the smoothed contours shown in Fig. 4.2. Geometric distortion can be treated by using the curvature partition and the angle-length profile. Because the shape representation that we use is a curvature partition composed of geometric parameters, the geometric transform can be easily done if a clear criterion is obtained. The criterion consists of the normalization of the sizes of the shapes and the point-to-point mapping acquired by the angle-length profile. Differences in image resolution can be dealt with by conducting the smoothing process on multiple scales in the feature point extraction step. Because multiple scales of smoothing results in coarse and fine structures of shapes, the contours are represented at varying levels of detail. This absorbs the effect of differences in resolution.

Now let us discuss how our method performs under noisy conditions and affine transforms. Our method's performance (in Section 4.5) deteriorated, although only slightly, for the occluded shape retrievals compared with whole shape retrievals using the method described in [22]. We assume this is caused by the angle-length profile (the normalization and mapping step), which is not exactly invariant under affine transformations. However, the deterioration in accuracy is so small that we believe our method is still applicable to affine transformed shapes. In addition, our method can be used in noisy conditions, since the contour smoothing (feature point extraction step) eliminates the noise. Studies showing this is are described in Appendix A.

Now let us compare our method with two previous approaches, one based on a shortest path search [98,123–125] [126,127], the other a geometrical consistency check [80,130] [131,132]. For this comparison, we point out the qualitative and quantitative differences between our method and the two approaches.

The first approach, based on a shortest path search, is different from our approach especially in the shape representation and the normalization process. While our representation is geometrically invariant with coarse and fine structures, the representations used with the shortest path search are, in most cases, not invariant under geometric transformations. For example, shortest path search methods often use curvature as a descriptor of contours of shapes [127], but curvature is inversely proportional to the contour size and should be normalized before comparison. For this reason, the accuracy of our method is approximately twice that of a shortest path search without normalization when the contour length ratio is more than 70%. However, shortest path searches still excel in finding the point correspondences between contours because they use the order of points on the contour coordinates. Therefore, the similarity can be well represented (normalized) for shape contours on which the order

of points identifies the shapes. The performance of DP matching (a typical shortest path search) is discussed in Appendix 4.8.1. Although the scores for this method are much smaller than those of our method, the tendency for the scores to decrease as the ratio of the contour length decreases is small. In fact, the scores of DP matching for contour length ratios of 50% are larger than those of our method. This result implies that the scores of DP matching would be better for specific shapes. Compared with the shortest path searches, our approach better represents and normalizes shapes especially under unpredictable conditions such as when occlusion, geometric distortion, and differences in resolution occur simultaneously.

The second approach, based on a geometric consistency check, is also different from our approach, especially in the shape representation and the normalization process. While our method represents the order of curvature partitions with an index and the partitioned areas with geometric parameters, the methods based on geometric consistency checks only represent the geometric relationships between features. For this reason, geometric consistency checks are only half as accurate as our method when the contour length ratio is more than 60%. However, they excel in detecting occluded shapes since the geometric relationships are invariant for occlusion. Therefore, the similarities in shape can be well represented for occluded shapes. Geometrical hashing (a typical geometric consistency check method) is described in Appendix 4.8.2. Although the scores for this method are much smaller than those of our method, the tendency for the scores to decrease as the ratio of the contour length decreases is small. For this reason, the score of geometrical hashing for the contour length ratio 50% is a bit larger than that of our method. This result implies that the scores of geometrical hashing would be better for specific shapes with occlusion. Compared with the geometric consistency checks, our approach better represents and normalizes shapes especially under unpredictable conditions such as when occlusion, geometric distortion, and differences in resolution occur simultaneously.

Finally, we should point out the consequences of our development. The experimental results showed that our method effectively retrieves shapes under unpredictable conditions. However, the classification accuracies of our results (the average scores for the classification test) seem far from 100% and do not seem sufficient for shape retrieval. Nonetheless, these accuracies are high enough for accurate shape retrieval. For instance, at an accuracy around 70%, "candidates" can be effectively nominated, and these candidates will be well normalized with our method. Therefore, we can use the previously proposed shape recognition methods for closed contours (such as [94, 137]) to select the most similar shape from the candidates. In summary, our method is effective enough for realizing accurate shape retrieval by combining it with previously proposed shape recognition methods for closed contours. An evaluation of open contour classification by using previously proposed methods in combination with ours will therefore be a topic of future work.

4.7 Appendix I
Performance of method under noisy conditions and affine transformations

This section describes empirical studies showing how our method performs under noisy conditions and affine transforms.

First, we studied how our method performed for affine transformations. This study used the examples shown in Fig. 4.11(a)(b) and the transformed shapes shown in Fig. 4.12(a-1)(b-1). Each transformed shape shown in Fig. 4.12 was made with

$$\begin{pmatrix} x_a & y_a & 1 \end{pmatrix} = \begin{pmatrix} x_o & y_o & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ \alpha & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \tag{4.19}$$

where (x_o, y_o) is a coordinate of the original shape (Fig. 4.11) and (x_a, y_a) is that of the transformed shape (4.12), and $\alpha = 0.1, 0.2, \cdot, 1.0$. For each shape in Fig. 4.12, our method calculates the similarity to Fig. 4.11. The results are shown in Fig. 4.13 with the parameter distances between Figs. 4.11 and 4.12. They show that the similarity slightly deteriorated when the shape was transformed (although only slightly). Despite that, they show that our method still works well for transformed shapes.

The performance of our method for the occluded shape retrievals (see Section 4.5) slightly deteriorated in comparison with that for the whole shape retrievals using the method described in [22]. This deterioration is presumably caused by the angle-length profile (the mapping step), which is not exactly invariant under affine transformations.

We studied the performances using shapes affected by salt and pepper noise. This study was done with the examples shown in Fig. 4.11(a)(b) and shapes with salt and pepper noise shown in Fig. 4.14(a-1)(b-1). Each noisy shape in Fig. 4.14 was obtained by adding salt and pepper noise quantified using the ratio of the number of pixels of the original shapes to that of noise $\alpha = 0.1, 0.2, \cdots, 1.0$. Our method calculated the similarity to Fig. 4.11 for each shape in Fig. 4.12 and for each shape in Fig. 4.14. The results are shown in Fig. 4.15. These results show that the similarity only slightly deteriorated as the noise increased. This suggests that our method works well since the contour smoothing (the feature extraction step) removes the noise from the extracted contours.

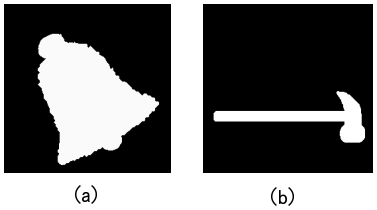


Fig. 4.11 Shapes used for empirical studies under noisy conditions and affine transforms. (a) is "bell-1", and (b) is "hammer-1" in the standard MPEG-7 CE-Shape-1 part B.

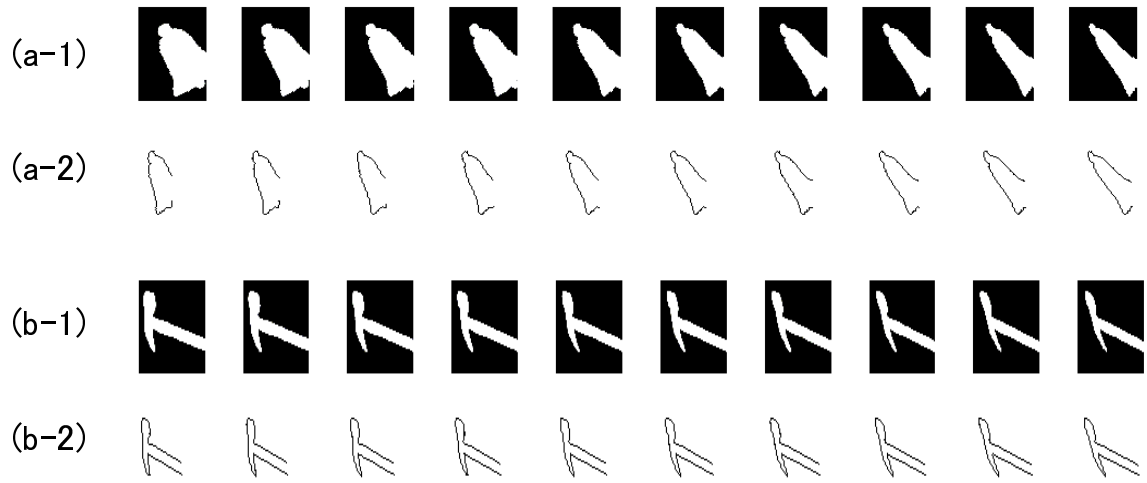


Fig. 4.12 Transformed shapes of Fig. 4.11(a)(b) with Eq. 4.19. From the left side, the shapes are obtained with $\alpha = 0.1, 0.2, \cdots, 1.0$. (a-1) is transformed from Fig. 4.11(a), and (b-1) is transformed from Fig. 4.11(b). (a-2) and (b-2) are contours obtained with (a-1) and (b-1) simultaneously.

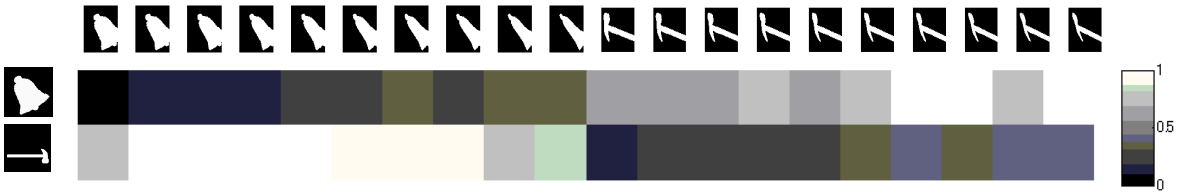


Fig. 4.13 Parameter distances between Fig. 4.11 and Fig. 4.12. The row is Fig. 4.11(a)(b), and the column is Fig. 4.12(a-1)(b-1). Each parameter distance is shown with a brightness (the less bright, the more similar).

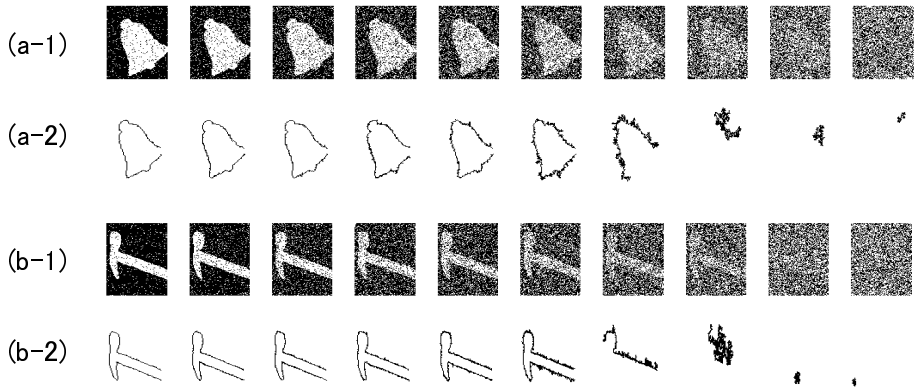


Fig. 4.14 Shapes of Fig. 4.11 with salt and pepper noise. From the left side, the shapes are obtained with the ratio of the number of pixels of noise to that of the pixels of the original shapes $\alpha = 0.1, 0.2, \cdot, 1.0$. (a-1) is for Fig. 4.11(a), and (b-1) is for Fig. 4.11(b). (a-2) and (b-2) are contours obtained with (a-1) and (b-1) simultaneously. Each noisy shape shown in Fig. 4.14 was obtained by adding salt and pepper noise with the ratios corresponding to noise $\alpha = 0.1, 0.2, \cdot, 1.0$.

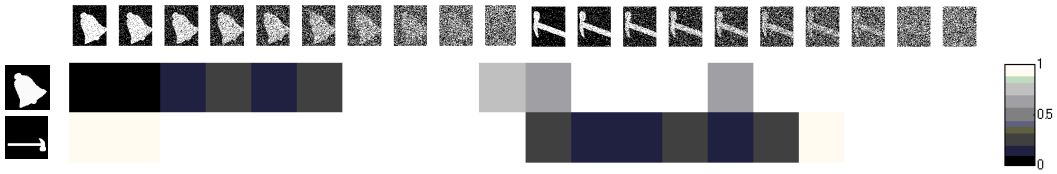


Fig. 4.15 Parameter distances between Fig. 4.11 and Fig. 4.14. The row is Fig. 4.11(a)(b) and the column is Fig. 4.14(a-1)(b-1). Each parameter distance is shown with a brightness (the less bright, the more similar).

4.8 Appendix II
Previous Methods for Comparison

This section evaluates the performance of previous whole-to-part shape matching approaches in order to compare them with our method. As mentioned in Section 4.1, the previous methods seem to have difficulty in identifying similar shapes while they excel in finding the point correspondences on shapes. In order to verify this supposition, we evaluated the effectiveness of a shortest path search and a geometric consistency check. The evaluations used the same standard MPEG-7 CE-Shape-1 part B data set that was used to evaluate our method in Section 4.5.

4.8.1 Shortest Path Search

Dynamic programming (DP) matching was selected as a representative shortest path search. There are more efficient calculation methods for finding shortest paths, such as Dynamic Time Warping [127]). However, we chose DP matching as a benchmark since we wanted to focus on shape matching accuracy, and DP matching is a pattern matching algorithm with high accuracy [98, 123–126].

The DP matching illustrated in Fig. 4.16. Before applying DP matching, each sample point should be described as a feature. As the descriptor, we used the curvature of the contour on the shapes. Therefore, each contour is represented as a vector C:

$$C = C(\kappa(t)), \tag{4.20}$$

where t denotes the contour coordinate from the starting point on the contour. Fig. 4.16(b) shows examples of curvatures of Fig. 4.16(a). Given two shapes A (a closed contour) and B (an open contour), represented as $C_A(\kappa(t_A))$ and $C_B(\kappa(t_B))$, we derive the map (t_A, t_B) that minimizes the parameter distance between $\kappa(t_A)$ and $\kappa(t_B)$. This map is searched with the graph shown in Fig. 4.16(c). The map (t_A, t_B) is found by taking a path on the graph whose y-axis denotes the contour coordinate of shape B (t_B) and whose x-axis denotes the duplicated (repeated) contour coordinate of shape A (t_A). The parameter distance between $C_A(\kappa(t_A))$ and $C_B(\kappa(t_B))$ is then allocated to each point on the graph. Finally, the map (t_A, t_B) is derived with DP as the path on which the accumulated parameter distance is minimized. Fig. 4.16(d) shows the point correspondences between two shapes. The path is investigated in the clockwise and counterclockwise directions on shape B. The minimum accumulated parameter distance describes the similarity of the two shapes.

Now let us turn to the results of applying this approach to the standard MPEG-7 CE-Shape-1 part B data set. This evaluation shows quite different performance from that of our method (Fig. 4.17). The average scores of the 70 shape classes are 31.26% (ratio of 99%), 29.71% (ratio of 90%), 28.80% (ratio of 80%), 27.77% (ratio of 70%), 24.80% (ratio of 60%), and 22.34% (ratio of 50%). These results clearly show that the matching accuracy of the DP matching is much worse than that of our method, especially for larger ratios. As a successful example (Fig. 4.18) and a failure example (Fig. 4.19) show, the DP matching is successful only when the point correspondences between mostly the same two contours are found. It appears that DP matching has difficulty in describing "global" similarity, since it is sensitive to local differences. In contrast, our method, which describes contours on multiple scales, can deal with global similarity.

4.8.2 Geometric Consistency Check

Geometric hashing was selected as a representative geometric consistency check. geometric hashing, originally introduced in [131], is one of the traditional methods for checking geometric consistency, and it is utilized in many fields [133,138,139] because of its efficiency and accuracy in identifying geometric relationships.

The geometric hashing we use here is illustrated Fig. 4.20. Before applying geometric hashing, the feature points should be extracted from the shapes. As features, we chose curvature extrema (peak points) of the contour on the shapes, in the same manner as [131] (points of sharp convexities and deep concavities). Each feature point is represented as a vector P:

$$P = P(x(i), y(i)), \tag{4.21}$$

where i denotes the index. Fig. 4.20(b) shows examples of feature points of Fig. 4.20(a). Given two shapes A (a closed contour) and B (an open contour), represented with feature points as $P_A(x(i_A), y(i_A))$ and $P_B(x(i_B), y(i_B))$, we derive the map (i_A, i_B) that minimizes the distance between P_A and P_B. For P (P_A and P_B), each set of three, non-collinear points (p_1, p_2, p_3) is chosen as a basis triplet. Examples of basis triplets are shown in Fig. 4.20(c). An affine transformation from (p_1, p_2, p_3) to $(0, 0)$, $(1, 0)$, $(0, 1)$ is performed on each point of P. Fig. 4.20(d) shows examples of the transformed points. After this transformation has been performed with all possible basis triplets, the best triplet is selected from each of shapes A and B by voting. After this transformation, a grid coordinate system is prepared, and each point of P is located on it, as shown in Fig. 4.20(e). After superimposing the grid coordinate system, we find how many votes each point gained. The basis triplet (p_1, p_2, p_3) with the most votes for P is selected for each of P_A and P_B, and the map (i_A, i_B) is finally obtained. Examples of the point correspondences with the map are shown in Fig. 4.20(f). Here, the similarity of two shapes is determined by the number of points located in the same bin.

Finally, let us describe the results of applying this approach to the standard MPEG-7 CE-Shape-1 part B data set. The performance in this case is quite different from that of our method shown in Fig. 4.21. The average scores of the 70 shape classes are 25.60% (ratio of 99%), 21.60% (ratio of 90%), 18.91% (ratio of 80%), 16.46% (ratio of 70%), 15.26% (ratio of 60%), and 15.26% (ratio of 50%). The results clearly show that the matching accuracy of geometric hashing is much worse than that of our method, especially for larger contour ratios. As a successful example (Fig. 4.22(a)) and a failure example (Fig. 4.22(b)) show, the geometric hashing is successful only when the features are located in the "same manner". Although it is strong against local noise, geometric hashing neglects information between points. In contrast, our method takes into account information between points by using geometric parameters. As this comparison shows, using geometric parameters in a suitable manner is necessary for describing shapes.

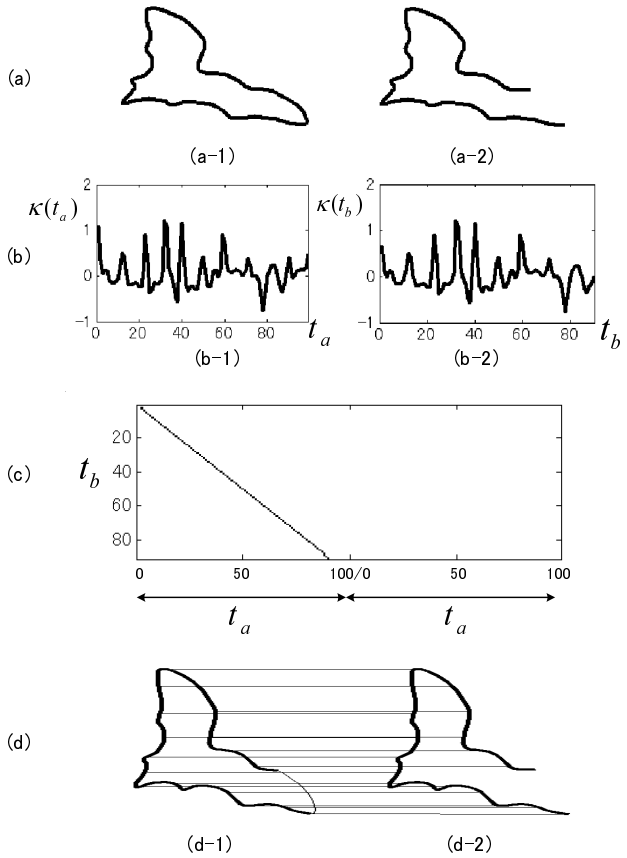
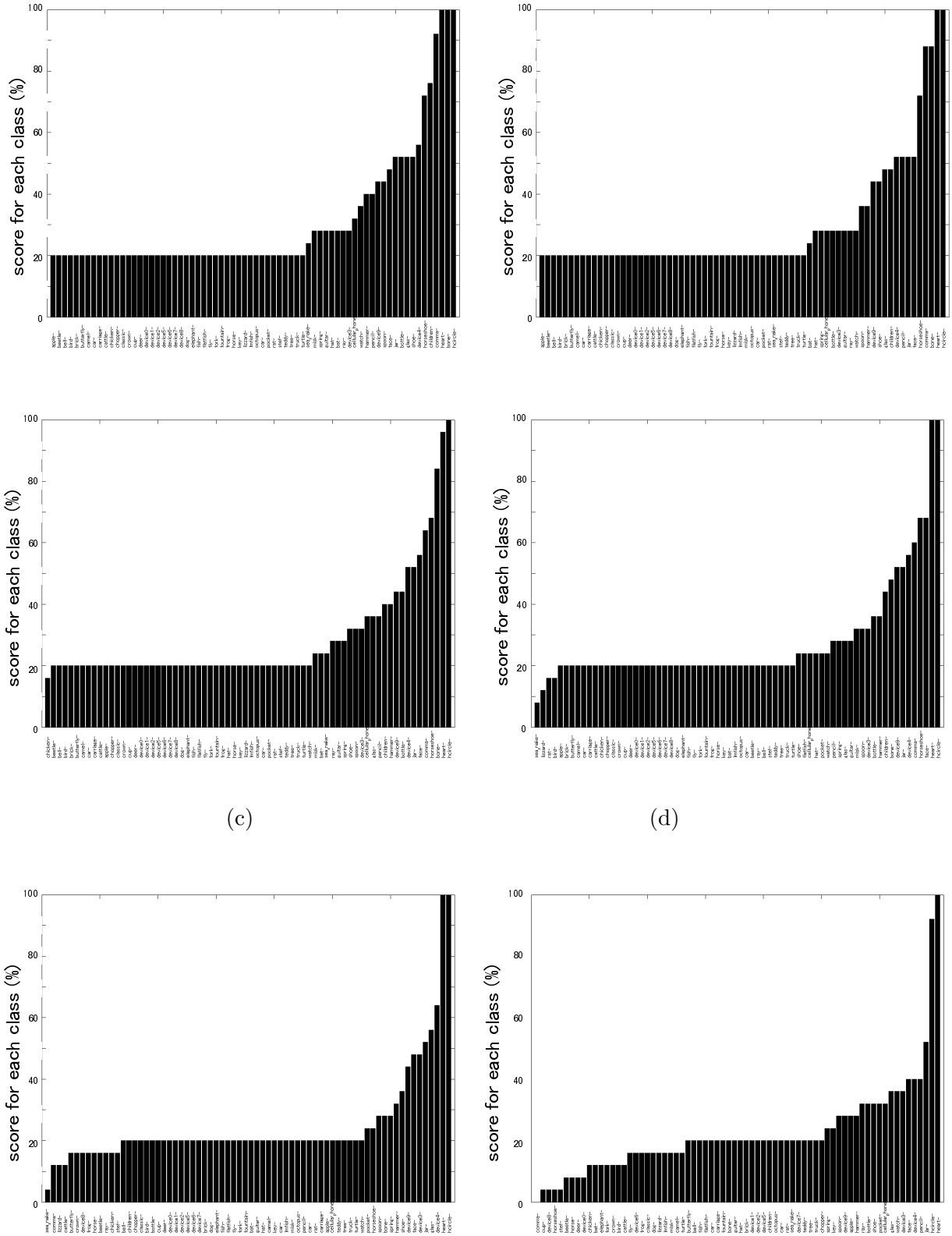


Fig. 4.16 Process of DP matching. An example of contours on a shape is shown in (a-1), and a fragment of (a-1) is shown in (a-2). The curvature of (a-1) and (a-2) are shown in (b-1) and (b-2). The x-axis of (b-1) and (b-2) is the contour coordinate, and the y-axis is curvature. The graph to search the map (t_A, t_B) is shown in (c). The y-axis of (c) is the contour coordinate of shape (a-2) (t_B), and the x-axis is the duplicated (repeated) contour coordinate of the shape (a-1) (t_A).The shortest path, which has the minimum parameter distance, is plotted in (c). The point correspondences are shown in (d). The selected corresponding points in (a-1) are shown with a thick solid line, and the remnant of the contour is shown with a thin line.



(c) (d)

(e) (f)

Fig. 4.17 Average score for each shape (shown in percentage) obtained by DP matching for ratios of (a) 99%, (b) 90%, (c) 80%, (d) 70%, (e) 60%, and (f) 50%. The average scores of 70 shape classes are $31.26, 29.71, 28.80, 27.77, 24.80,$ and 22.34% for ratios of $99\%, 90\%, 80\%, 70\%, 60\%,$ and 50%.

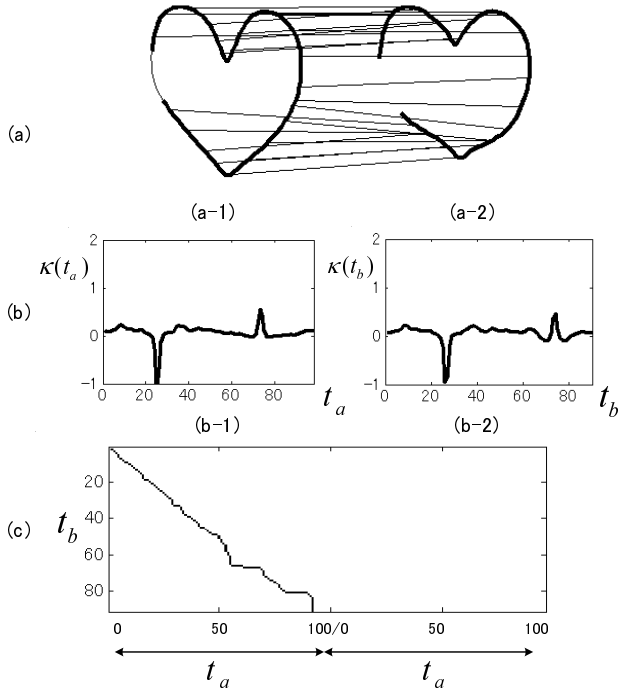


Fig. 4.18 Example of the contours in which the point correspondences are successfully found. The selected corresponding point pair is shown in (a). The curvature of (a-1) is shown in (b-1), and that of (a-2) is shown in (b-2). The shortest path is shown in (c).

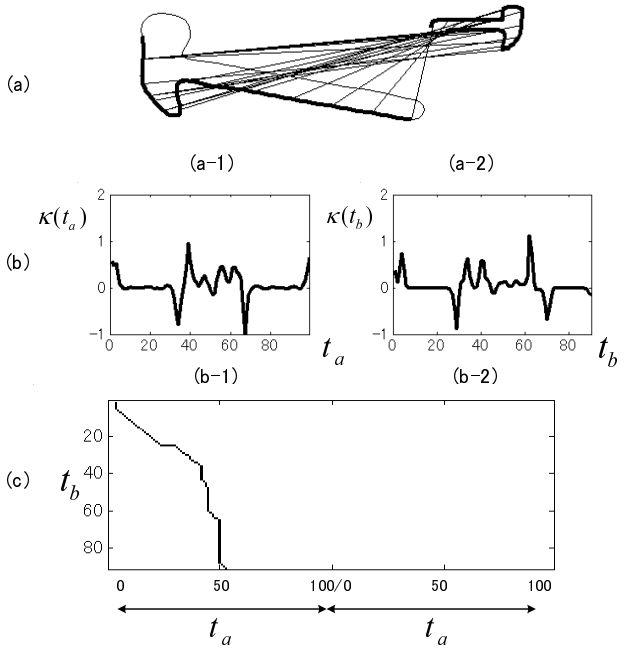


Fig. 4.19 Example of the contours which the point correspondences are not found. The selected corresponding point pair is shown in (a). The curvature of (a-1) is shown in (b-1), and that of (a-2) is shown in (b-2). The shortest path is shown in (c).

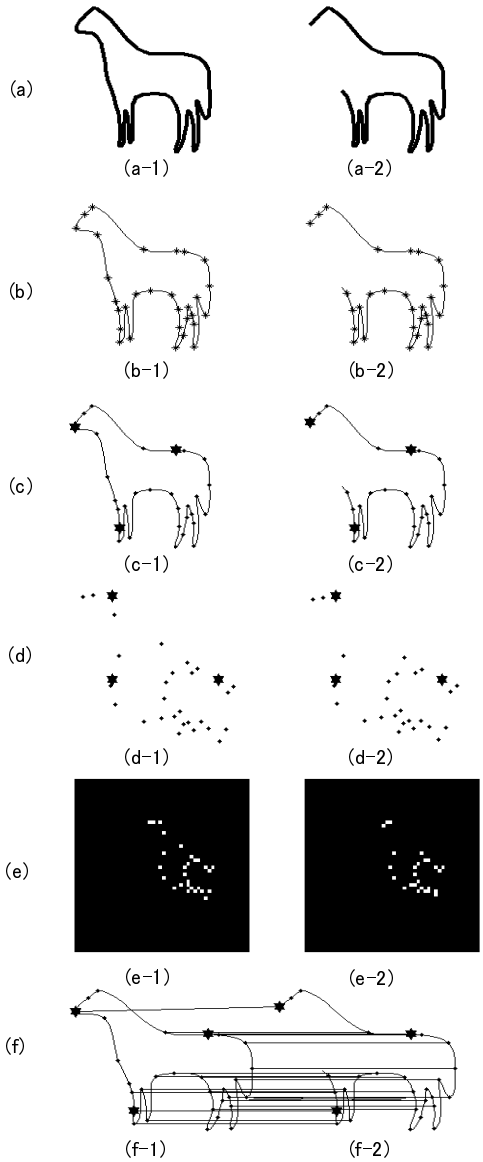
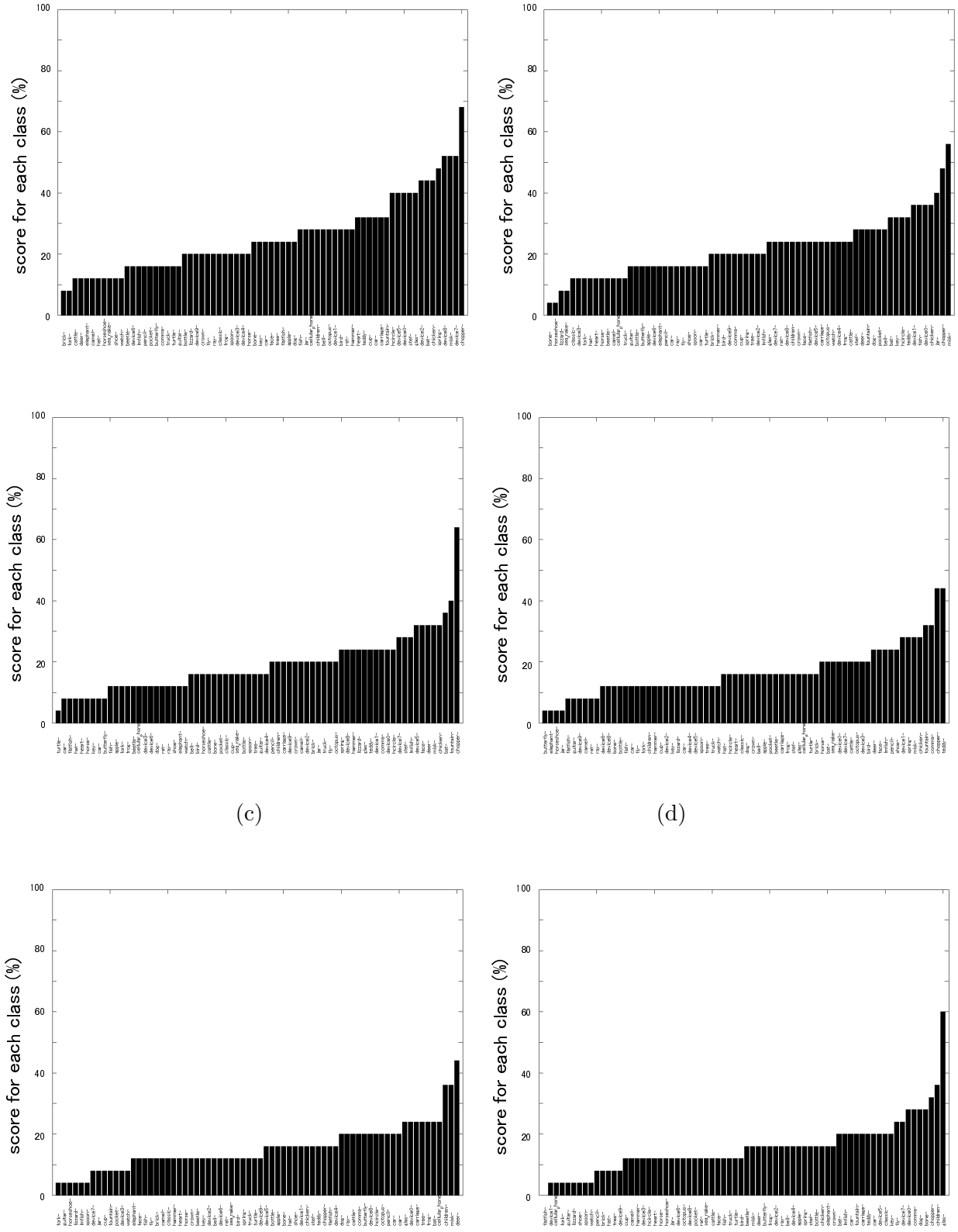


Fig. 4.20 Process of geometric hashing. An example of the contours on a shape is shown in (a-1), and a fragment of (a-1) is shown in (a-2). Feature points on (a-1) and (a-2) are shown in (b-1) and (b-2) with asterisks on the contours (thin lines). Examples of extracted basis triplets from the feature points are shown in (c-1) and (c-2) with black stars. The feature points after an affine transformation that transformed the basis triplets to $(0,0)$, $(1,0)$, $(0,1)$ are shown in (d-1) and (d-2). The grid coordinate system obtained by the transformed feature points are shown in (e-1) and (e-2). Examples of the point correspondences obtained by voting on the possible triplets are shown in (f-1) and (f-2).



(c) (d)

(e) (f)

Fig. 4.21 Average score for each shape (shown as a percentage) obtained by geometric hashing for ratios of (a) 99%, (b) 90%, (c) 80%, (d) 70%, (e) 60%, and (f) 50%. The average scores of 70 shape classes are $25.60, 21.60, 18.91, 16.46, 15.26,$ and 15.26% for ratios of $99\%, 90\%, 80\%, 70\%, 60\%,$ and 50%.

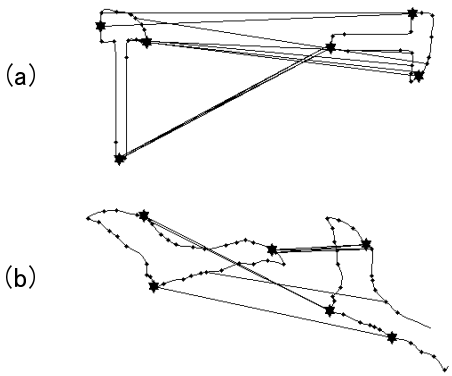


Fig. 4.22 Example of the contours for which the point correspondences are successfully found (a), and example of the contours for which the point correspondences are not successfully found (b).

Chapter 5

Application I
System of detecting underwater threats in side scan sonar images

5.1 Introduction

For object detection systems in the maritime environment, high reliability is quite challenging to be accomplished since the maritime environment changes unpredictably and dynamically. Underwater threats are, for example, buried in noisy images with occlusion. Generally speaking, high reliability for object detection systems is accomplished by learning a huge number of the samples under various different environments. However, samples of underwater objects are quite hard to be collected. Underwater threats are no exception.

For this reason, we propose a promising method of recognizing objects called "One Shot Detector (OSD)". OSD is a method of recognizing objects with their shapes included in color images without machine learning procedure. OSD is, then, based on the human visual system, which performs well under unpredictably and dynamically changing environment. OSD is composed of two main processes, one of which is to extract object area, and the other of which is to represent the area.

1. **Object Area Extraction**
   It has been quite challenging for computational visual systems to extract object area. Objects are represented with pixels in images. In the same time, background is also represented with pixels in images. How do computational visual systems distinguish the pixels of objects from those of the background? Pixels in images are usually quite hard to be automatically distinguished since RGB values of pixels are not usually exactly the same. Our approach to extract object area is, on the other hand, to represent pixels as neural oscillators, which are synchronized with neighbors when their rhythms are similar. With this mechanism, object areas are "autonomously" constructed.
2. **Shape Representation**
   Object areas are required to be described with shape representations in order to match other object areas. However, it is usually quite hard to robustly describe objects in real world scenes since they are observed under the conditions of noise/occlusion/geometric deformation. On the other hand, our shape representation, called "curvature partition", describes shapes with multiple resolution contours, each of which is divided as partitions with curvature zero-crossing

points [22]. These partitions are, then, describes with geometric parameters. With our representation, noises described with high resolutions are ignored in low resolutions, occluded parts are ignored as occluded partitions, and geometric deformations can be transformed with geometric parameters. Therefore, our representation can robustly describe objects in real world scenes.

In this chapter, we show an approach to detect underwater threats in side scan sonar (SSS) images using OSD, which combines our object area extraction process and shape representation process. We, then, show that OSD effectively detects underwater objects included in wide-area SSS images. As a result, we show the effectiveness of our approach. Finally, through the comparison to a promising previous approach, we show that our approach is suitable under the condition that any machine learning procedure cannot be used. .

5.2 Method

In this section, we propose "One Shot Detector" (OSD), a promising method of recognizing objects. We, then, explain how to detect underwater threats on side scan sonar images using OSD. Our proposed method OSD comprises four steps as shown in Fig. 6.1, image acquisition, image segmentation (explained in Chapter 2), shape representation (explained in Chapter 3), and shape matching (explained in Chapter 4).

In the image acquisition step, input images are divided into small grids. In this chapter, the grid size is limited to no more than 60×60 pixels.

In the image segmentation step, pixels on each image grid are segmented with nonlinear oscillators that describe color relationships among their neighbors. Each pixel is described by a set of oscillators for their color R, G, and B. The oscillation pattern of them determines the link strength between each two pixels, and the strongly connected pixels form a segment that corresponds to a shape on the image. The construction of the oscillator network is shown in Fig. 5.2, and the dynamics of the interaction between two pixels u and v are given by

$$\partial D_{u,v} = \sum_{c \in C} s_{u,v} W_{u,v} (x_v^c - x_u^c), \tag{5.1}$$

where C denotes a set vector elements which represent color (R, G, and B), x_v^c and x_u^c denote nonlinear oscillators, $s_{u,v}$ denotes the first connection strength determined by globally formed groups, and $W_{u,v}$ denotes the second connection strength determined by the synchronicity of the two oscillators given by

$$dW_{u,v}/dt = \sum_{c \in C} \psi(x_v^c - x_u^c) \quad for \;\; W_{min} \leq W_{u,v} \leq W_{max}, \tag{5.2}$$

where W_{min} and W_{max} denote the lower and upper limits for $W_{u,v}$, and $\psi(\zeta, \xi)$ denotes an equation determined by using the following expression:

$$\psi(\zeta, \xi) = \begin{cases} w & (\zeta > 0 \cap \xi > 0) \\ -w & (\zeta\xi < 0) \\ 0 & (\text{otherwise}). \end{cases} \tag{5.3}$$

The first connection strength $s_{u,v}$ is, then, determined by

$$s_{u,v} = \begin{cases} 1 & (|I_k - I_u| < I_{th}) \\ 0 & (\text{otherwise}) \end{cases} \quad for \quad \forall(v) \in \phi_k, \tag{5.4}$$

where I_u denotes the vector for the color of pixel u, I_k denotes the vector for the average color of pixels included in the segment k which u belongs to, I_{th} denotes the threshold, and ϕ_k denotes a set of pixels belonging to the segment k.

In the shape representation step, shapes extracted as segments are described with a shape representation "curvature partition" as shown in Fig. 5.3.

The curvature partitions are constructed as partial curves whose edges are the curvature zero-crossing points on contour C on multiple scales of smoothing. They are expressed as $P_i(i = 1, 2, \ldots, n)$, where n is the number of curvature partitions that the contour has. Once they are constructed, they are divided into N sub-partitions. Each sub-partition p_{ij} is expressed in terms of its average curvature κ_{ij}, length l_{ij}, normal direction θ_{ij}, and location $o(x_{ij}, y_{ij})$:

$$p_{ij} = \begin{pmatrix} \kappa_{ij} \\ l_{ij} \\ \theta_{ij} \\ x_{ij} \\ y_{ij} \end{pmatrix}, \tag{5.5}$$

where $j = 1, 2, \ldots, N$. Therefore, curvature partition P_i has $N \times 5$ elements:

$$P_i = (p_{i1}, p_{i2}, \ldots, p_{iN}). \tag{5.6}$$

Each curve on a contour (which is a partial curve whose edges are both the curvature zero-crossing points) is expressed by its curvature, length, normal direction, and location. The curvature partitions, (P_1, P_2, ..., P_n), on multiple scales of smoothing are stored as representations of the shape.

In the shape matching step, the curvature partitions of shapes A and B are compared with each other on multiple scales after geometric transformation. The comparison is done with the derivation of the similarity between shapes A and B, $S_{A,B}$ which is given by

$$S_{A,B} = S_{th} - \sum_{i=1}^{n} \sum_{j=1}^{N} d_{ij}, \tag{5.7}$$

where S_{th} denotes the threshold, d_{ij} denotes the parameter distance for sub-partition p_{ij} of shapes A and B defined with its curvature $d\kappa_{ij}$, length dl_{ij}, location do_{ij}, and normal direction $d\theta_{ij}$ as follows:

$$d_{ij} = w_l dl_{ij} + w_\kappa d\kappa_{ij} + w_o do_{ij} + w_\theta d\theta_{ij}, \tag{5.8}$$

where w_l, w_κ, w_o, and w_θ are weight parameters for the curvature, length, location and normal direction of each sub-partition. In this chapter, shapes A and B are defined as similar shapes if the similarity between the two shapes $S_{A,B}$ fulfills:

$$S_{A,B} > 0. \tag{5.9}$$

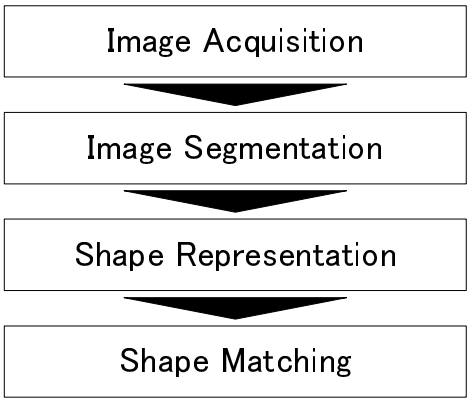


Fig. 5.1 Overview of OSD.

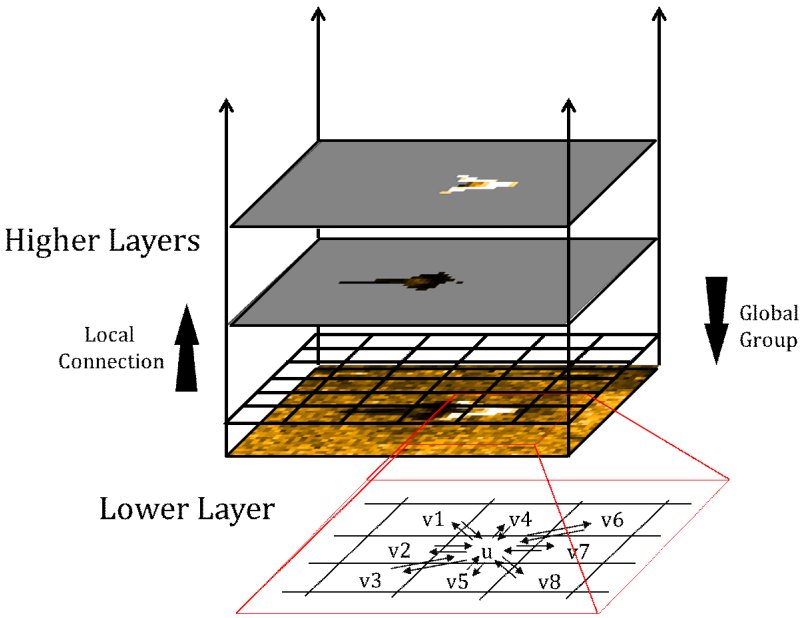


Fig. 5.2 Image Segmentation.

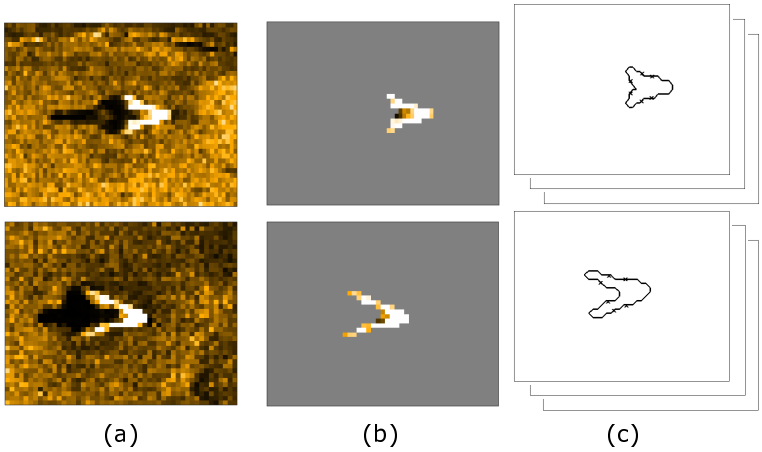


Fig. 5.3 Shape Representation and Matching.

5.3 Experimental Results

In this section, we conduct two types of experiments using SSS images acquired by a high resolution SSS system, Klein System 3000 [140], which has frequencies of 100 kHz and 500 kHz, an angular resolution (beam width) of 0.70° at 100 kHz or 0.21° at 500 kHz, and maximum range 600 m at 100 kHz or 150 m at 500 kHz. First experiment is "object detection test", in which "target objects" included in one "training image" are detected in a wide-area SSS image. The target object included in the training data is a tetrapod. Second experiment is, then, "accuracy evaluation", in which 20 small grid SSS images, some of which includes target objects (tetrapods) and the others of which do not, are classified as those with the target objects and those without them. Both experiments are conducted with SSS images acquired by a frequency of 500 kHz.

5.3.1 Object Detection Test

In this experiment, target objects (tetrapods) are detected in a wide-area SSS image which is acquired by Klein System 3000. Firstly, the target object image is given as a training image, a small grid SSS image as shown in Fig. 5.4(a). Then, a wide-area SSS image is given as shown in Fig. 5.4(b). The wide-area SSS image is, then, divided into small grids as shown in Fig. 5.4(c). Each grid image is matched to the training data, and the similarities between two segments included in the grid image and the training image are calculated. Finally, segments are chosen if their similarities with the segments included in the training image fulfill the Eq. 6.9. In this way, the grids including the target objects (tetrapods) are selected and shown with a green square as shown in Fig. 5.4(d). As a result, three tetrapods are detected as shown in Fig. 5.4(e). This result shows that our "One Shot Detector" has the ability to detect objects with "one shot" (one training image).

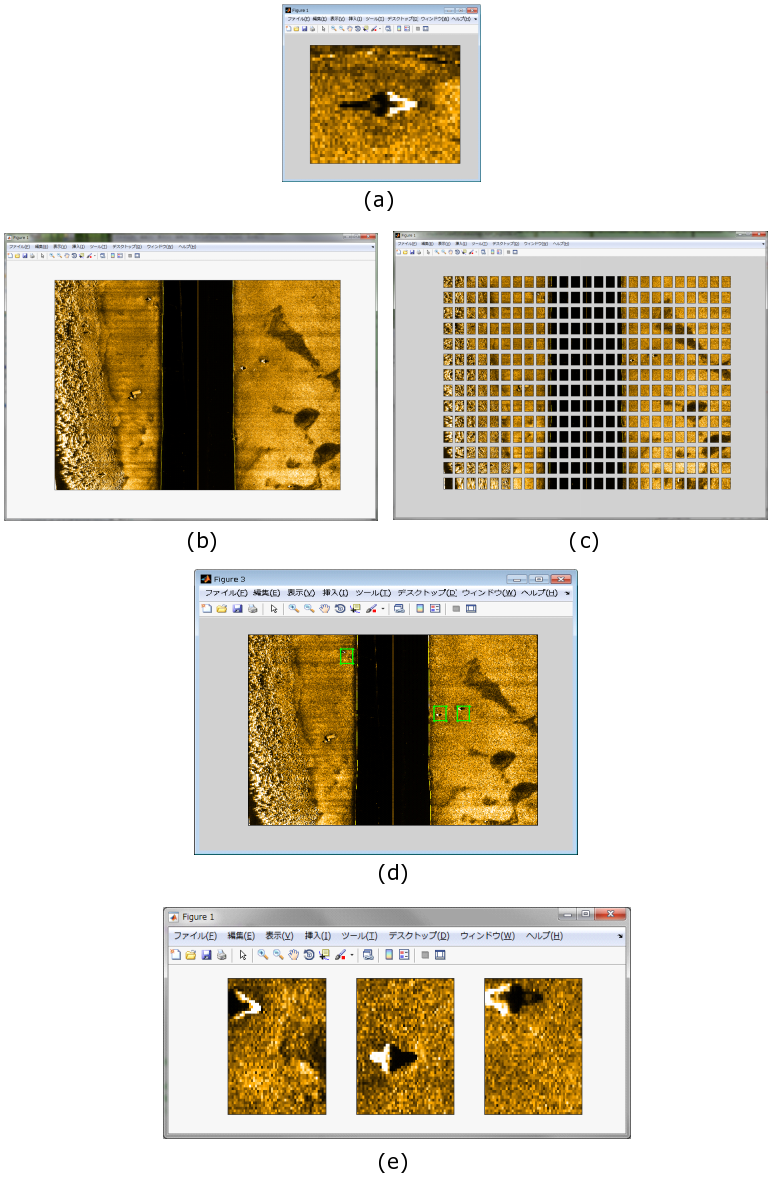


Fig. 5.4 Results of Object Recognition Test

5.3.2 Accuracy Evaluation

In this experiment, we show the effectiveness of our OSD compared to a promising previous method Geometric Hough Transformation (GHT). Our evaluation is done with 20 grid images shown in Fig. 5.5. Figs. 5.4 (1)-(10) include the target objects (tetrapods). For the evaluation of the classification accuracy of our method, we compare every image that includes one tetrapod with all other images, and calculate the similarities between each two images.

For the evaluation of this experiment, we express similarity matrices. In the similarity matrices, the similarities of the detected shapes with the comparison of images are expressed. The similarity matrix for OSD is shown in Fig. 5.6(a) and that for GHT is

shown in Fig. 5.6(c). Then, in order to evaluate the classification accuracy, we firstly choose the thresholds which maximize the classification accuracy, and we derive the classification accuracy with the numbers of true positive, false positive, true negative, and false negative. In this manner, we choose the threshold for OSD (S_{th} in Eq. 6.7) as 0.8 and that for GHT as 80. Then, the elements of the similarity matrices over the thresholds are shown with white square and those under the thresholds are shown with black square in in Fig. 5.6(b) for OSD and in Fig. 5.6(d) for GHT. Although OSD seems successful for deriving almost all similarities, GHT does not.

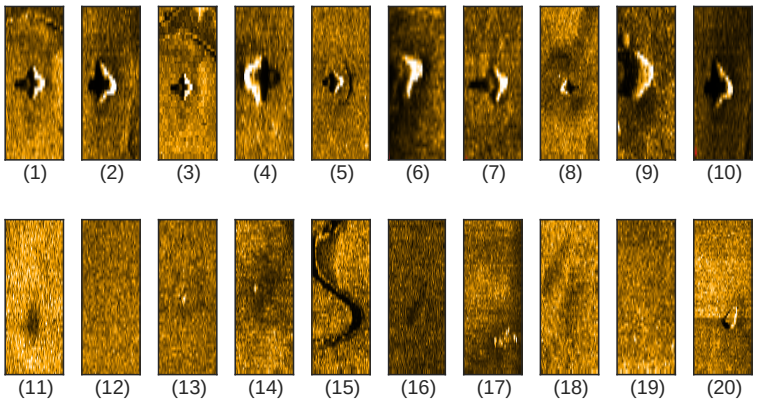


Fig. 5.5 Test Images for Accuracy Evaluation

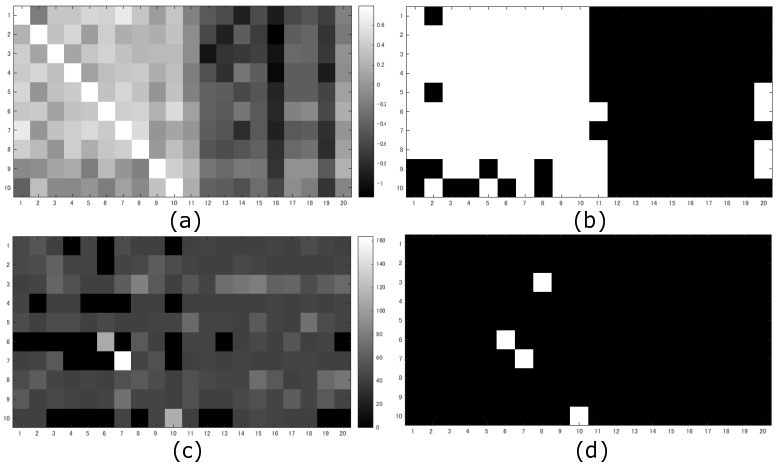


Fig. 5.6 Results of Accuracy Evaluation

Table 5.1 Confusion Matrix for OSD

		Predicted Class	
		Positive	Negative
Actual Class	Positive	89	11
	Negative	8	92

Table 5.2 Confusion Matrix for GHT

		Predicted Class	
		Positive	Negative
Actual Class	Positive	4	96
	Negative	0	100

The accuracy is quantitatively evaluated with confusion matrices in Table 5.1 and Table 5.2. In the confusion matrices, the number of "true positives" (actual tetrapods that are correctly detected as tetrapods), "false negatives" (tetrapods that are not detected), "false positives" (other objects that are incorrectly detected as tetrapods), and "true negatives" (non-tetrapods that are correctly classified as non-tetrapods) are shown. The rows of the confusion matrices denote the actual class ("positive" denotes that the tetrapods are included in the images, and "negative" denotes that they are not), and the columns denote the predicted class ("positive" denotes that tetrapods are detected with OSD or GHT, and "negative" denotes that they are not). As a result, 89.0% of all tetrapods are detected with OSD while 4.0% of them are detected with GHT. Then, OSD exactly recognizes 92.0% of all images which do not include tetrapods while GHT recognizes 100.0% of them. The classification accuracy with OSD is, then, 90.1% while that with GHT is 52.0% (the classification accuracy of even random classification method is still 50%).

These results clearly show that OSD can classify images since it detects shapes included in images and represents and matches shapes. On the other hand, GHT, in principle, is hard to detects shapes since it does not distinguish features belonging to an object from others. In order to distinguish them, object "area" is required to be identified, and in order to regard the area as the object, the shape of the area is represented. For this reason, two processes that OSD is composed of, object area extraction and shape representation, are key factors to successfully recognize objects, and these experimental results show that these processes are helpful to improve object detection accuracy.

5.4 Discussion and Conclusion

We, finally, discuss the applicability of OSD. A distinct feature of OSD is that it clearly represents features of an object included in one sample image. With previous methods, however, representing features of an object requires a large number of samples. Therefore, OSD is applicable for objects located in environment that people are hard to access. For this reason, OSD is quite suitable for detecting underwater threats in SSS images.

In this chapter, we proposed an object recognition approach called "One Shot Detector" (OSD), which is composed of image segmentation and shape representation.

We, then, proposed an approach to detect underwater threats in SSS images using OSD. Our approach, which is suitable for the case without prior knowledge, is shown to accomplish a great performance (the classification accuracy of 90.1% compared to 52.1% of a previous method) in SSS images. The detection tests for various types of objects are the future work.

Chapter 6

Application II
Brain-like Object Recognition on Satellite Images without Machine Learning Scheme

6.1 Introduction

As the use of high resolution satellite images has been widespread, expectations are growing for automatic recognition of objects in these images. However, it is a quite challenging task for computational visual systems to automatically recognize arbitrary objects without learning a huge number of samples observed in real world conditions. Generally speaking, patterns of objects are unpredictable since the real world is dynamically changing. Objects are sometimes occluded, geometrically distorted, or superimposed by noise signals. For this reason, object recognition requires sufficient prior-knowledge in order to predict various patterns of the objects described in the real world condition. High reliability would be accomplished with enough number of samples with conventional methods. However, sample images taken on satellite images cannot always be sufficiently acquired. It is usually not always easy to predict any patterns of the objects described under unpredictable conditions.

On the other hand, the human visual system performs well under unpredictably and dynamically changing environment even when a huge number of patterns are not previously learned. For this reason, neural network models that emulate the hierarchical processing of the human visual system have traditionally been proposed [74, 105]. However, these neural networks basically assume that a number of samples are acquired, and then common features of these samples should be learned. Therefore, object recognition without learning a large number of samples have not been accomplished with neural network models. We assume that computational visual systems for object recognition without machine learning schemes are required to solve two problems: (1) a figure/ground separation, and (2) an invariant shape representation. A figure/ground separation is a basic function of the human visual system that segregates focused objects from the background. Likewise, the human visual system plausibly represent shapes invariantly even under noise, occlusion, or geometric distortion [22]. These two problems are assumed to be hard for conventional computational systems to solve without prior knowledge or machine learning schemes.

In this chapter, we propose an approach to detect object on satellite images using

"One Shot Detector" (OSD) in order to to accomplish object recognition without learning scheme. As shown in Chapter 5, a figure/ground separation is solved with a neuron-like nonlinear oscillator network, and an invariant shape representation is accomplished by a curvature based hierarchical shape descriptor. As a result, our One Shot Detector is shown to achieve much greater performance compared to previous methods for object detection on satellite images.

6.2 Method

In this section, we explain how to detect objects on satellite images using "One Shot Detector" (OSD), which is proposed in Chapter 5.

In the image acquisition step, input images are divided into small grids. In this chapter, the grid size is limited to no more than 50×50 pixels.

In the image segmentation step, pixels on each image grid are segmented with nonlinear oscillators that describe color relationships among their neighbors. Each pixel is described by a set of oscillators for their color R, G, and B. The oscillation pattern of them determines the link strength between each two pixels, and the strongly connected pixels form a segment that corresponds to a shape on the image. The construction of the oscillator network is shown in Fig. 6.2, and the dynamics of the interaction between two pixels u and v are given by

$$\partial D_{u,v} = \sum_{c \in C} s_{u,v} W_{u,v} (x_v^c - x_u^c), \tag{6.1}$$

where C denotes a set vector elements which represent color (R, G, and B), x_v^c and x_u^c denote nonlinear oscillators, $s_{u,v}$ denotes the first connection strength determined by globally formed groups, and $W_{u,v}$ denotes the second connection strength determined by the synchronicity of the two oscillators given by

$$dW_{u,v}/dt = \sum_{c \in C} \psi(x_v^c - x_u^c) \quad for \;\; W_{min} \leq W_{u,v} \leq W_{max}, \tag{6.2}$$

where W_{min} and W_{max} denote the lower and upper limits for $W_{u,v}$, and $\psi(\zeta, \xi)$ denotes an equation determined by using the following expression:

$$\psi(\zeta, \xi) = \begin{cases} w & (\zeta > 0 \cap \xi > 0) \\ -w & (\zeta \xi < 0) \\ 0 & (\text{otherwise}). \end{cases} \tag{6.3}$$

The first connection strength $s_{u,v}$ is, then, determined by

$$s_{u,v} = \begin{cases} 1 & (|I_k - I_u| < I_{th}) \\ 0 & (\text{otherwise}) \end{cases} \quad for \;\; \forall (v) \in \phi_k, \tag{6.4}$$

where I_u denotes the vector for the color of pixel u, I_k denotes the vector for the average color of pixels included in the segment k which u belongs to, I_{th} denotes the threshold, and ϕ_k denotes a set of pixels belonging to the segment k.

In the shape representation step, shapes extracted as segments are described with a shape representation "curvature partition" as shown in Fig. 6.3.

The curvature partitions are constructed as partial curves whose edges are the curvature zero-crossing points on contour C on multiple scales of smoothing. They

are expressed as $P_i(i = 1, 2, \ldots, n)$, where n is the number of curvature partitions that the contour has. Once they are constructed, they are divided into N sub-partitions. Each sub-partition p_{ij} is expressed in terms of its average curvature κ_{ij}, length l_{ij}, normal direction θ_{ij}, and location $o(x_{ij}, y_{ij})$:

$$p_{ij} = \begin{pmatrix} \kappa_{ij} \\ l_{ij} \\ \theta_{ij} \\ x_{ij} \\ y_{ij} \end{pmatrix}, \tag{6.5}$$

where $j = 1, 2, \ldots, N$. Therefore, curvature partition P_i has $N \times 5$ elements:

$$P_i = (p_{i1}, p_{i2}, \ldots, p_{iN}). \tag{6.6}$$

Each curve on a contour (which is a partial curve whose edges are both the curvature zero-crossing points) is expressed by its curvature, length, normal direction, and location. The curvature partitions, (P_1, P_2, ..., P_n), on multiple scales of smoothing are stored as representations of the shape.

In the shape matching step, the curvature partitions of shapes A and B are compared with each other on multiple scales after geometric transformation. The comparison is done with the derivation of the similarity between shapes A and B, $S_{A,B}$ which is given by

$$S_{A,B} = S_{th} - \sum_{i=1}^{n} \sum_{j=1}^{N} d_{ij}, \tag{6.7}$$

where S_{th} denotes the threshold, d_{ij} denotes the parameter distance for sub-partition p_{ij} of shapes A and B defined with its curvature $d\kappa_{ij}$, length dl_{ij}, location do_{ij}, and normal direction $d\theta_{ij}$ as follows:

$$d_{ij} = w_l dl_{ij} + w_\kappa d\kappa_{ij} + w_o do_{ij} + w_\theta d\theta_{ij}, \tag{6.8}$$

where w_l, w_κ, w_o, and w_θ are weight parameters for the curvature, length, location and normal direction of each sub-partition. In this chapter, shapes A and B are defined as similar shapes if the similarity between the two shapes $S_{A,B}$ fulfills:

$$S_{A,B} > 0. \tag{6.9}$$

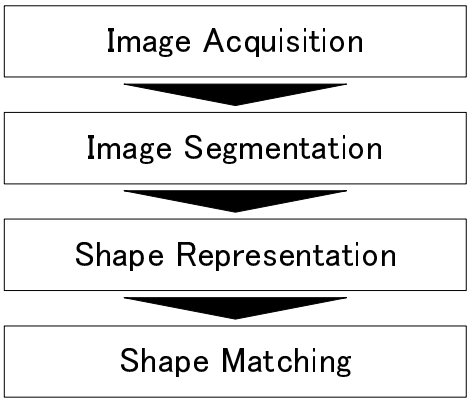


Fig. 6.1 Overview of OSD.

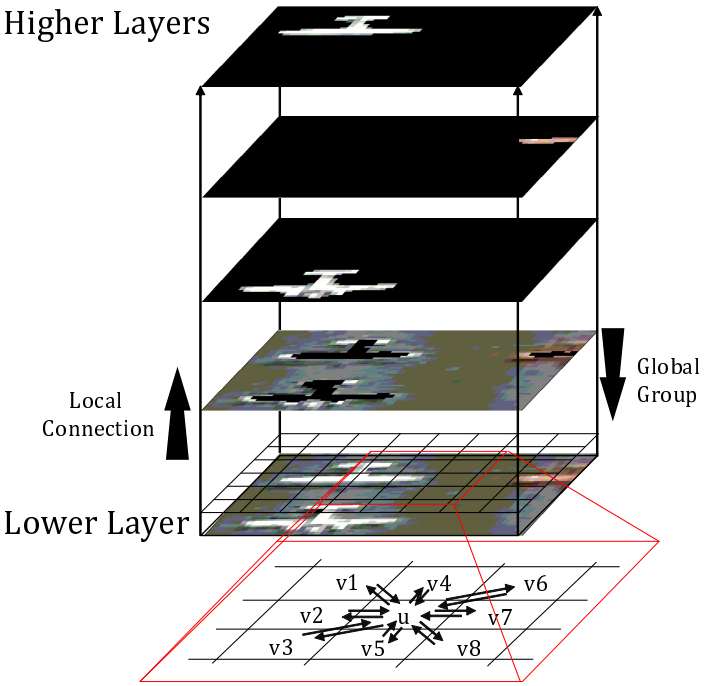


Fig. 6.2 Image Segmentation.

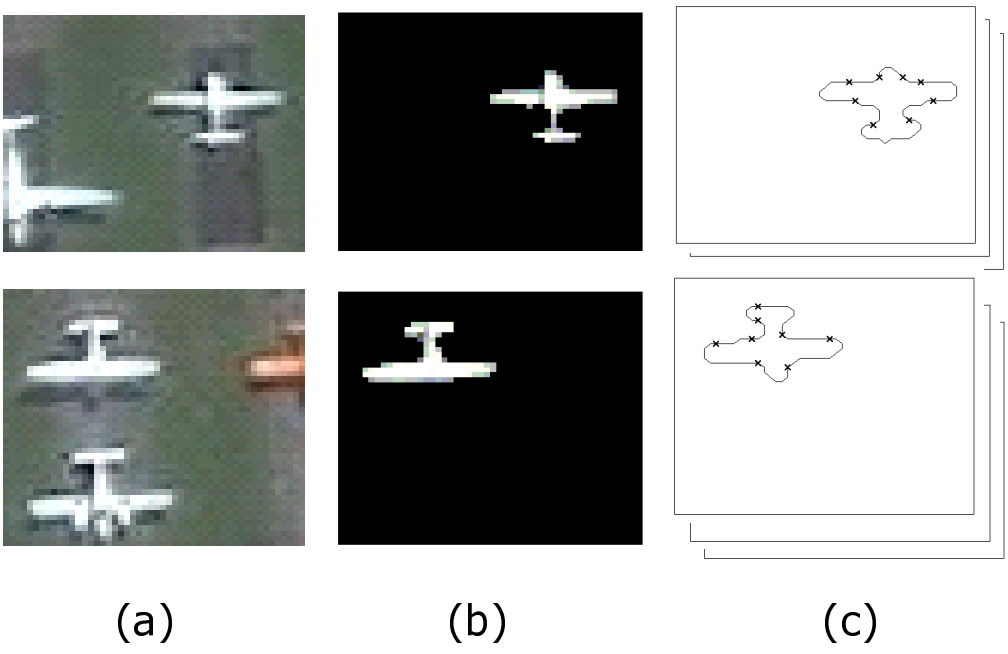


Fig. 6.3 Shape Representation and Matching.

6.3 Experimental Results

In this section, we show our evaluation that shows the effectiveness of our proposed "One Shot Detector (OSD)" compared to a promising previous method Geometric Hough Transformation (GHT). Our evaluation is done with 25 mini-scale images shown in Fig. 6.4 which are extracted as divided grid squares of a satellite image published by MapMart [141] as shown in Fig. 6.5. Figs. 6.4 (1)-(10) and (21)-(25) include the shapes of airplanes. Especially, a multiple number of airplanes included in (7) and (8) and complex backgrounds of (21)-(25) make it difficult to accurately recognize the airplanes included in the images. For the evaluation of the accuracy of our method, we compare every image that includes one or more airplanes with all other images so that airplanes are detected. As a result, airplanes are detected as shown in Fig. 6.6, which column denotes all images (1)-(25), and which row denotes images that include airplanes (1)-(10) and (21)-(25). Therefore, airplanes would be detected in the column of (1)-(10) and (21)-(25), and nothing would be detected in the column of (11)-(20). We call the images chosen in the column "Observed Images" and the images chosen in the row "Sample Images". Then, in order to focus on one type of object (airplane), we choose one pixel in each sample image so that shapes (corresponding to airplanes) including the selected pixels are automatically selected. The shapes selected in this manner are shown in Fig. 6.7. For the evaluation of this experiment, we express the detection accuracy with similarity matrices and confusion matrices. In the similarity matrices, the similarities of the detected shapes with the comparison of images are expressed. The similarity matrix for OSD is shown in Fig. 6.8(a) and that for GHT is shown in Fig. 6.8(b). The columns of the similarity matrices denote all images (1)-(25), and the rows of them denote images that include airplanes (1)-(10) and (21)-(25) as Fig. 6.6. Ideally, all elements of columns (1)-(10) and (21)-(25) would be maximum similarity value, and those of columns (11)-(20) would be minimum similarity value except for elements in the diagonal line. Although most of the elements of the similarity matrix for OSD have this tendency, those for GHT have similar values regardless of the columns. These results show that OSD determines the similarity between two shapes, but GHT does not. Without parameter tuning through learning a large number of shapes, GHT cannot define the similarity between two shapes. The results of the statistical analysis for the similarity matrices are shown with the confusion matrices shown in Table 6.1 (for OSD) and Table 6.2 (for GHT). In the confusion matrices, the number of "true positives" (actual airplanes that are correctly detected as airplanes), "false negatives" (airplanes that are not detected), "false positives" (other objects that are incorrectly detected as airplanes), and "true negatives" (non-airplanes that are correctly classified as non-airplanes) are shown. The rows of the confusion matrices denote the actual class ("positive" denotes that the airplanes are included in the images, and "negative" denotes that they are not), and the columns denote the predicted class ("positive" denotes that airplanes are detected with OSD or GHT, and "negative" denotes that they are not). As a result, 71.4 % of all airplanes are detected with OSD while 44.3 % of them are detected with GHT. Then, OSD exactly recognizes 82.7 % of all images which do not include airplanes while GHT recognizes 66.7 % of them. The classification accuracy with OSD is, then, 76.1 % while that with GHT is 53.6 % (the classification accuracy of even random classification method is still 50 %). It is assumed that there are two reasons of the failure of the detection using OSD. One is the influence of the small fragments. For our evaluation, we chose one pixel in each sample image, and we

selected segments that include the pixel. It is supposed that some of these segments are shapes as small fragments. The other reason of the failure is that images of our evaluation include airplanes of different shapes. It is true that our human brains have acquired the concept of "airplanes", it would be too difficult to represent "airplanes" only with one shape.

Fig. 6.4 Mini-scale Images for Evaluation.

Fig. 6.5 Satellite Image presented by MapMart.

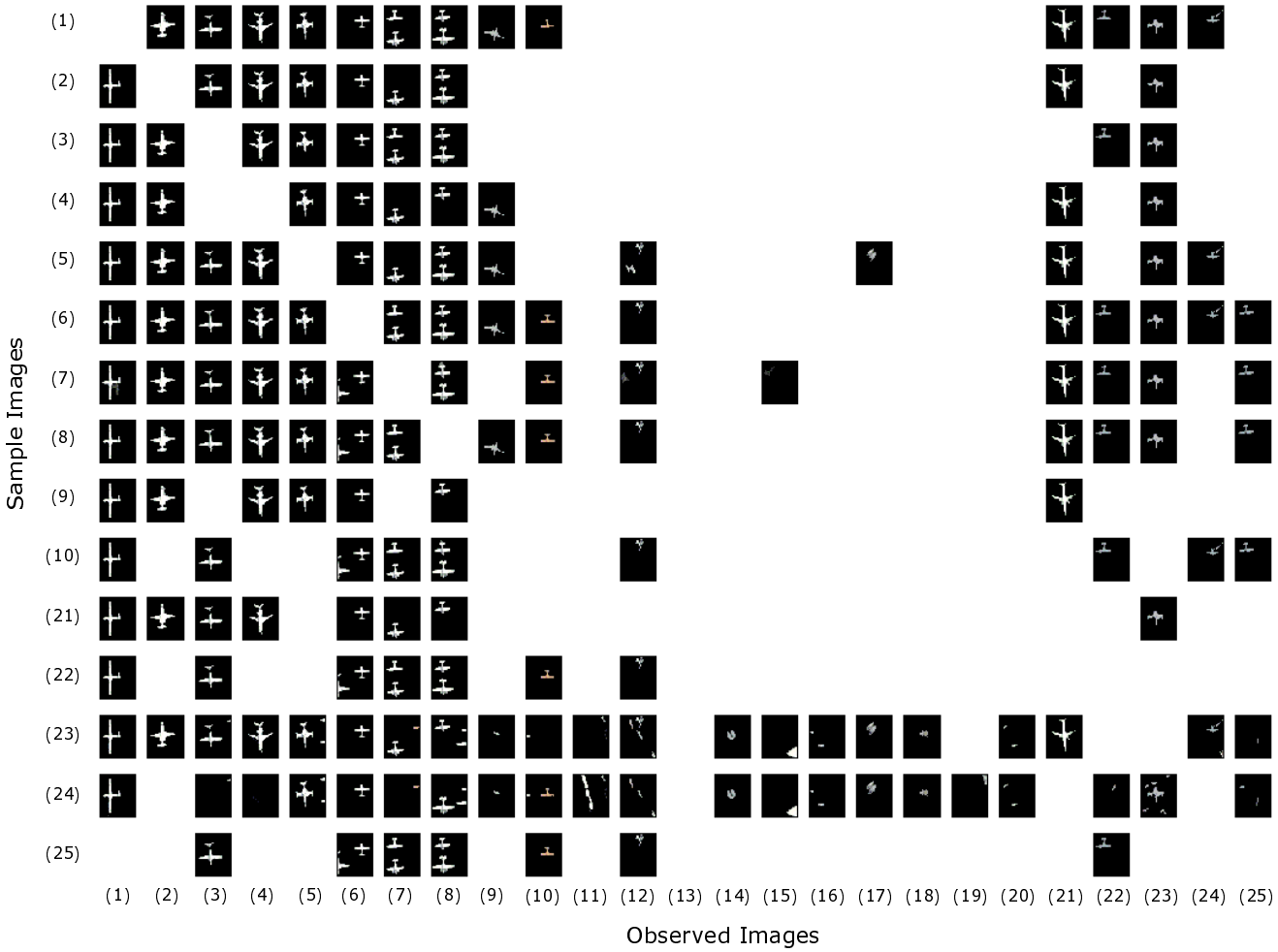


Fig. 6.6 Detected Shapes.

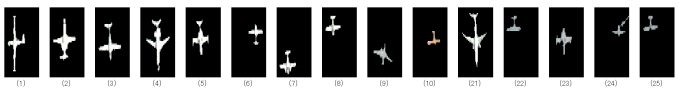


Fig. 6.7 Corresponding Shapes.

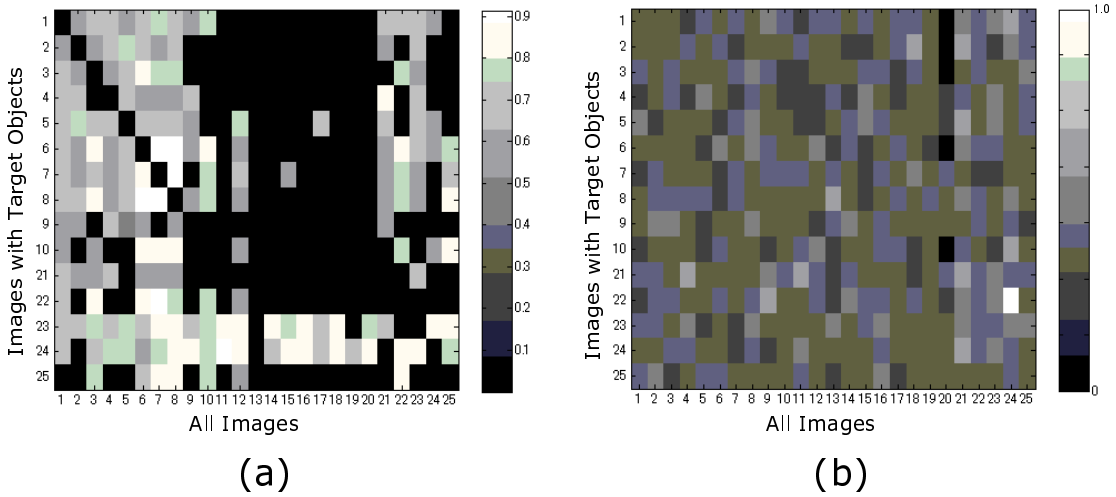


Fig. 6.8 Evaluation Results (Similarity Matrices).

Table 6.1 Confusion Matrix for OSD

		Predicted Class	
		Positive	Negative
Actual Class	Positive	150	60
	Negative	26	124

Table 6.2 Confusion Matrix for GHT

		Predicted Class	
		Positive	Negative
Actual Class	Positive	93	117
	Negative	50	100

6.4 Discussion

In this section, we discuss the effectiveness and the applicability of our proposed method. Our method "One Shot Detector" (OSD) is composed of image segmentation step and shape representation step. The effectiveness of these two steps is confirmed in condition without prior knowledge. Traditionally, the image segmentation cannot be accomplished without using the prior knowledge on the images such as the number of segments included in these images. Similarly, previous shape representation cannot be invariant in case when noise, preliminary unknown occlusion, and geometric distortion simultaneously occur. Our OSD, however, determines these parameters using the information included in the observed images even without this prior knowledge. As a

result, our method accomplished much greater performances than a previous method, Geometric Hough Transformation (GHT). We, finally, discuss the applicability of our proposed method. Although our method does not need prior knowledge for object recognition, there is plenty of knowledge on objects with huge number of samples, such as human faces or human finger prints. Now that satellite images are widespread, knowledge on general airplanes can easily be stored. However, there are still many objects without enough knowledge. For example, it is quite challenging to store knowledge when "new" airplane is developed. Our method is applicable especially in these cases.

6.5 Conclusion

In this chapter, we proposed an approach to detect objects on satellite images using OSD. Our approach, which is suitable for the case without prior knowledge, is shown to accomplish a great performance (the classification accuracy of 76.1 % compared to 53.3 % of a previous method) in satellite images. The detection tests for other objects are the future work.

Chapter 7

Discussion and Conclusion

In this thesis, we proposed models to solve issues on the processes of "perceptual grouping", "shape representation", and "shape mapping" in order to make it clear how the human visual system recognizes object in real-world scenes, which are in unpredictably and dynamically changing uncertain environments. As a result, we made it clear that how figure-ground separation occurs through the models of "perceptual grouping", "shape representation", and "shape mapping" in real-world scenes.

Firstly, our proposed model of the perceptual grouping in color images models the inter-element interaction, the inter-layer interaction (the interaction between the globally formed groups and locally formed connections), and the inter-variable interaction (the interaction among RGB values) to solve the ill-posed problem. Our proposed model consists of two hierarchical layers of neural oscillator networks and their interactions [31, 46, 47]. As a result, we showed that our model can perceive not only the color surface but also the gradation on the surface.

Secondly, our proposed shape representation "curvature partition" was shown to be biologically plausible for matching general shapes with their contour curvature information. Although our representation is not based on hierarchical neural network models, it reflects the constraints that are required for shape matching. We firstly defined a "curvature partition" as a descriptor of a group of partitioned contours. We, then, implemented a Gaussian blur in multiple scales in order to obtain a coarse and a fine spatial coding. Finally, we applied the geometric transformation in order to achieve the invariant shape recognition. Therefore, our shape representation "curvature partition" is applicable in condition of occlusion, geometric transformation, and difference in the resolution of images.

Thirdly, we proposed a shape retrieval method using a "curvature partition" and a shape mapping including a normalization using an "angle-length profile". Although the curvature partition excels at representing shapes, the previous methods do not have a mapping process including a normalization that works when there are occlusions, geometric distortions, and differences in image resolution. Here, we described such a normalization process using the angle-length profile. As a result, unlike the previously proposed methods, our method is shown to be applied in situations with occlusion, geometric distortion, and differences in image resolution.

Fourthly, we presented the evaluation results that our proposed model of perceptual grouping, shape representation, and shape mapping method are applied for recognizing object in real-world scenes (side scan sonar images and satellite images). The evaluation results were compared to those of generalized Hough transformation (a promising conventional method). Through the comparison, it was shown that our models are applicable for object recognition method even without using machine learning scheme with a large number of samples.

Through these models and evaluations, the understanding on how the human visual system (visual cortex) recognizes object is progressed especially from the view of shape recognition. Although the mechanism of "shape" recognition is understood through this paper, it is still unclear how the visual cortex recognizes "objects" by referring to their shapes. In other words, it is still unclear what is it to "recognize object". It is assumed that the visual cortex assumes what the objects are through their appearances (shapes). In other words, when the visual cortex assumes what an object is, the visual cortex infers its whole and semantic by referring to the appearance (shape) in the context in which the visual cortex is. Now few are known on semantic (whole) understanding, it would be too much challenging to computationally model how to "recognize object". However, we assume that there is still a good chance that computational modeling approaches help us to develop an understanding of semantic (whole) understanding. The computational modeling of the time-series variations or the shape patterns of shapes as inference processes are good examples to help us to understand the processes of the object recognition. Therefore, our future works are the computational modeling approaches for the understanding of object recognition.

Finally, we explain the significance of applying the knowledge of brain science for computational visual system. Although our computational approaches are based on the knowledge of brain science, computational visual systems do not always need to be based on the brain's visual system. Also, "object recognition" can be partially accomplished if the problems on "object recognition" are clearly set (defined) without using brain-like computational approaches. However, the brain's functions are not only to recognize objects but also to "limit" the context in which the brain recognize the objects. Because our brain is in the real world, which is an unpredictably and dynamically changing environment, the semantics of objects cannot be determined unless the contexts are clearly determined. Therefore, we assume it is quite meaningful to investigate how to "limit" the contexts in which objects are and how to recognize the objects in these contexts. Notably, this study approaches problems on the real world (an unpredictably and dynamically changing environment). What we achieved in this study is to detect shapes which commonly included in arbitrary two pictures without any prior knowledge. In other words, in arbitrary two real-world scenes, without any prior knowledge on the scenes or objects, our methods detects commonly included shapes by limiting the context through the processes of "perceptual grouping", "shape representation", and "shape mapping". We would like to emphasize that it is meaningful to investigate how to "limit" the context in which the brain recognizes objects for computational visual system.

Achievements

Academic Journals

1. Matsuda Y, Ogawa M, Yano M (2012) Visual shape representation with geometrically characterized contour partitions. Biological cybernetics 106(4-5): 295-305.
2. Matsuda Y, Ogawa M, Yano M (2015) Shape retrieval with geometrically characterized contour partitions. IEEE Access (accepted in 2015.05.19).

International Conferences

1. Matsuda Y, Ogawa M, Yano M (2015) Brain-like Object Recognition on Satellite Images without Machine Learning Scheme. ISRS2015, APR 22-24, Tainan, Taiwan.
2. Matsuda Y, Ogawa M, Yano M (2015) Object recognition system without machine learning schemes. MAST Asia 2015, MAY 13-15, Yokohama, Japan.
3. Matsuda Y, Ogawa M, Yano M (2015) System of detecting underwater threats in side scan sonar images. Oceans MTS/IEEE 2015, MAY 18-21, Genova, Italy.

Japanese Conferences

1. Matsuda Y, Ogawa M, Yano M (2012) General shape recognition using contour representation with a coarse to fine structure. Technical Report of IEICE (NC2012–55).
2. Matsuda Y, Ogawa M, Yano M (2013) General shape retrieval method using curvature of occluded contours. Technical Report of IEICE (NC2013–7).
3. Matsuda Y, Ogawa M, Yano M (2014) Brain-like Object Recognition on Satellite Images without Prior Knowledge. 57th (2014 Autumn) Meeting of the Remote Sensing Society of Japan (B05).
4. Matsuda Y, Ogawa M, Yano M (2015) Brain-like Occluded Object Recognition on Satellite Images without Prior Knowledge. DICOMO2015 (Multimedia, Distributed, Cooperative, and Mobile Symposium) of IPSJ (Information Processing Society of Japan) (2H–4).

Award

1. Best Presentation Award for research on “Brain-like Occluded Object Recognition on Satellite Images without Prior Knowledge.” from DICOMO2015 (Multimedia, Distributed, Cooperative, and Mobile Symposium) of IPSJ (Information Processing Society of Japan).

Acknowledgement

The author would like to express his heartfelt gratitude to a professor Dr. Shigeo Sato for accepting him as a Ph.D student, supervising this work, giving helpful information and comments, and doing critical reading of the manuscripts. He also would like to express his gratitude to a professor Dr. Shinichiro Omachi and a professor Dr. Takuo Suganuma for reviewing his work, and giving helpful advices. He also deeply thanks a professor emeritus Dr. Masafumi Yano for lecturing "brainware", giving intuitive information and comments, and supporting his research in NEC Corporation. Besides, he is also deeply grateful to a principal researcher Mr. Masatsugu Ogawa in NEC Corporation for giving interesting information and comments, and supporting his research as a supervisor. Also, he is deeply grateful to all members of The Society for Brainware including the representative Mr. Toshiyuki Kano, memebers of the laboratory of Nano-Integration Devices and Processing, and Brain Computing Research Team in NEC Corporation for valuable advices and encouragement.

References

[1] Leventhal A.G., Thompson K.G., Liu D, Zhou Y, and Ault S.J. Concomitant sensitivity to orientation, direction, and color of cells in layers 2, 3, and 4 of monkey striate cortex. *J Neurosci*, Vol. 15, No. 3 Pt 1, pp. 1808–18, March 1995.

[2] Sincich L.C. and Horton J.C. The circuitry of V1 and V2: integration of color, form, and motion. *Annu Rev Neurosci*, Vol. 28, pp. 303–326, January 2005.

[3] Hubel D.H. and Wiesel T.N. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *J Physiol*, Vol. 160, No. 1, p. 106, 1962.

[4] Connor C.E., Gallant J.L., Preddie D.C., and Van Essen D.C. Responses in area V4 depend on the spatial relationship between stimulus and attention. *J Neurophysiol*, Vol. 75, No. 3, pp. 1306–8, March 1996.

[5] Pasupathy A and Connor C.E. Shape representation in area V4: position-specific tuning for boundary conformation. *J Neurophysiol*, Vol. 86, No. 5, pp. 2505–19, November 2001.

[6] Pasupathy A and Connor C.E. Population coding of shape in area V4. *Nat Neurosci*, Vol. 5, No. 12, pp. 1332–8, December 2002.

[7] Sakamoto K, Kumada T, and Yano M. A computational model that enables global amodal completion based on V4 neurons. *ICONIP 2010*, pp. 9–16, 2010.

[8] Brincat S.L. and Connor C.E. Underlying principles of visual shape selectivity in posterior inferotemporal cortex. *Nat Neurosci*, Vol. 7, No. 8, pp. 880–6, August 2004.

[9] Roelfsema P.R. Cortical algorithms for perceptual grouping. *Annu Rev Neurosci*, Vol. 29, pp. 203–27, January 2006.

[10] Tanaka K. Inferotemporal cortex and object vision. *Annu Rev Neurosci*, Vol. 19, pp. 109–39, January 1996.

[11] Poggio T, Torre V, and Koch C. Computational vision and regularization theory. *Nature*, Vol. 317, pp. 314–319, 1985.

[12] van Lier R.J. and Leeuwenberg E.L.J. Multiple completions primed by occulusion patterns. *Perception*, Vol. 24, pp. 727–740, 1995.

[13] Sajda P and Finkel L. Intermediate-Level Visual Representations and the Construction of Surface Perception. *J Cogn Neurosci*, Vol. 7, pp. 655–660, 1995.

[14] Sekuler A.B., Stephen P.E., and Flynn C. Local and Global Processes in Visual Completion. *Psychol Sci*, Vol. 5, No. 5, pp. 260–267, September 1994.

[15] Williamson J.R. Neural network for dynamic binding with graph representation: form, linking, and depth-from-occlusion. *Neural Comput*, Vol. 8, No. 6, pp. 1203–25, August 1996.

[16] Inui T. Properties of human visual memory for block patterns. *Biol Cybern*, Vol. 187, pp. 179–187, 1988.

[17] Klymenko V, Weisstein N, Topolski R, and Hsieh C.H. Spatial and temporal frequency in figure-ground organization. *Percept Psychophys*, Vol. 45, No. 5, pp. 395–403, May 1989.

[18] Sakai K and Inui T. A feature-segmentation model of short-term visual memory. *Perception*, Vol. 31, No. 5, pp. 579–589, 2002.

[19] Victor J. D. and Conte M.M. Encoding and stability of image statistics in working memory. *Vision Res*, Vol. 46, No. 24, pp. 4152–62, November 2006.

[20] Ito M, Tamura H, Fujita I, and Tanaka K. Size and position invariance of neuronal responses in monkey inferotemporal cortex. *J Neurophysiol*, Vol. 73, No. 1, pp. 218–26, January 1995.

[21] Booth M.C. and Rolls E.T. View-invariant representations of familiar objects by neurons in the inferior temporal visual cortex. *Cerebral cortex (New York, N.Y. : 1991)*, Vol. 8, No. 6, pp. 510–23, September 1998.

[22] Matsuda Y, Ogawa M, and Yano M. Visual shape representation with geometrically characterized contour partitions. *Biol Cybern*, Vol. 106, No. 4-5, pp. 295–305, July 2012.

[23] Matsuda Y, Ogawa M, and Yano M. Shape retrieval with geometrically characterized contour partitions. *IEEE Access*.

[24] Marr D and Poggio T. Cooperative computation of stereo disparity. *Science*, Vol. 194, pp. 283–287, 1976.

[25] Marr D and Poggio T. A computational theory of human stereo vision. *Proc R Soc Lond B204*, pp. 301–328, 1979.

[26] Zhou H, Friedman S.H., and von der Heydt R. Coding of Border Ownership in Monkey Visual Cortex. *J Neurosci*, Vol. 20, pp. 6594–6611, 2000.

[27] Zhaoping L. Border ownership from intracortical interactions in visual area v2. *Neuron*, Vol. 47, pp. 143–153, 2005.

[28] Craft E, Schutze H, and von der Heydt E, Nieburand R. A Neural Model of Figure-Ground Organization. *J Neurophysiol*, Vol. 97, pp. 4310–4326, 2007.

[29] Grossberg S and Mingolla E. Neural dynamics of form perception: boundary completion, illusory figures, and neon color spreading. *Psychol Rev*, Vol. 92, pp. 173–211, 1985.

[30] Yamaguchi Y and Shimizu H. Pattern recognition with figure-ground separation by generation of coherent oscillations. *Neural Netw*, Vol. 7, pp. 49–63, 1994.

[31] Sato N and Yano M. A model of binocular stereopsis including a global consistency constraint. *Biol Cybern*, Vol. 371, pp. 357–371, 2000.

[32] Cao Y and Grossberg S. Stereopsis and 3D surface perception by spiking neurons in laminar cortical circuits: a method for converting neural rate models into spiking models. *Neural Netw*, Vol. 26, pp. 75–98, 2011.

[33] Cesmeli E and Wang D.L. Motion segmentation based on motion/brightness integration and oscillatory correlation. *IEEE Trans Neural Netw*, Vol. 11, pp. 935–947, 2000.

[34] Grossberg S, Leveille J, and Versace M. How do object reference frames and motion vector decomposition emerge in laminar cortical circuits? *Atten Percept Psychophys*, Vol. 73, pp. 1147–1170, 2011.

[35] Grossberg S and Raizada R.D. Contrast-sensitive perceptual grouping and object-based attention in the laminar circuits of primary visual cortex. *Vision Res*, Vol. 40, pp. 1413–1432, 2000.

[36] Nomura N, Ichikawa M, Okada K, Miike H, and Sakurai T. Raction-diffusion systems in pattern formation and pattern recognition processes. *ISCGAV '12*, pp. 161–166, 2010.

[37] Wang D.L. and Terman D. Locally excitatory globally inhibitory oscillator networks. *IEEE Trans Neural Netw*, Vol. 6, pp. 283–286, 1991.

[38] Yu G and Slotine J.J. Visual grouping by neural oscillator networks. *IEEE Trans Neural Netw*, Vol. 20, pp. 1871–1884, 2009.

[39] Li C and Li Y. Fast and robust image segmentation by small-world neural oscillator networks. *Cogn Neurodyn*, Vol. 5, pp. 209–220, 2011.
[40] Meier M, Haschke R, and Ritter H.J. Perceptual grouping through competition in coupled oscillator networks. *Neurocomputing*, Vol. 141, pp. 76–83, 2014.
[41] Quiles M.G., Wang D.L., Zhao L, Romero R.M.F., and Huang D.S. Selecting salient objects in real scenes: An oscillatory correlation model. *Neural Netw*, Vol. 24, pp. 54–64, 2011.
[42] Chen K and Wang D.L. A dynamically coupled neural oscillator network for image segmentation. *Neural Netw*, Vol. 15, pp. 423–439, 2002.
[43] Wang D.L. The time dimension for scene analysis. *IEEE Trans Neural Netw*, Vol. 16, pp. 1401–1426, 2005.
[44] Itti L, Koch C, and Niebur E. Computational vision and regularization theory. *IEEE Trans Pattern Anal Mach Intell*, Vol. 20, pp. 1254–1259, 1998.
[45] Koch C and Ullman S. Shifts in selective visual attention: towards the underlying neural circuitry. *Hum Neurobiol*, Vol. 4, pp. 219–227, 1985.
[46] Shimizu H, Yamaguchi Y, Tsuda I, and Yano M. Pattern recognition based on horonic information of vision. In: Haken H(ed) Complex systems – operational approaches in neuro- biology, physics, and computers. *Springer, Berlin Heidelberg New York*, Vol. 69, pp. 225–239, 1985.
[47] von der Malsburg C. Nervous structures with dynamical links. *Ber Bunsenges Phys Chem*, Vol. 89, pp. 703–710, 1985.
[48] Kimura S, Yano M, and Shimizu H. A self-organizing model of walking patterns of insects. *Biol Cybern*, Vol. 69, pp. 183–193, 1993.
[49] Land E.H. Color vision and the Natural image part I. *Proc Natl Acad Sci*, Vol. 45, pp. 115–129, 1959.
[50] Land E.H. Color vision and the Natural image part II. *Proc Natl Acad Sci*, Vol. 45, pp. 636–644, 1959.
[51] Land E.H. Experiments in Color Vision. *Sci Am*, Vol. 200, pp. 84–99, 1959.
[52] Land E.H. Experiments in Color Vision. *Am Sci*, Vol. 52, pp. 247–264, 1964.
[53] Land E.H. The Retinex Theory of Color Vision. *Sci Am*, Vol. 218, pp. 108–128, 1977.
[54] O'Brien V. Contour Perception, Illusion and Reality. *J Opt Soc Am*, Vol. 48, pp. 112–119, 1958.
[55] Semmelroth C.C. Prediction of Lightness and Brightness on Different Backgrounds. *J Opt Soc Am*, Vol. 60, pp. 1685–1689, 1970.
[56] Fukunaga K and Hostetler D. The extimation of the gradient of a density function. *IEEE Trans Inf Theory*, Vol. 21, pp. 32–40, 1975.
[57] Cheng Y. Mean shift, mode seeking and clustering. *IEEE Trans Pattern Anal Mach Intell*, Vol. 17, pp. 790–799, 1995.
[58] Comaniciu D and Meer P. Mean shift: A robust approach toward feature space analysis. *IEEE Trans Pattern Anal Mach Intell*, Vol. 24, pp. 603–619, 2002.
[59] Dunn J.C. A fussy relative of the ISODATA process and its use in detecting compact well-separated clusters. *J Cybern*, Vol. 3, pp. 32–57, 1973.
[60] Bezdek J.C. A convergence theorem for the fuzzy ISODATA clustering algorithm. *IEEE Trans Pattern Anal Mach Intell*, Vol. 2, pp. 1–8, 1980.
[61] Lim Y.W. and Lee S.U. On the color image segmentation algorithm based on the thresholding and the fuzzy c-means techniques. *Pattern Recognit*, Vol. 23, pp. 935–952, 1990.
[62] Terman D and Wang D.L. Global competition and local cooperation in a network of neural oscillators. *Physica D*, Vol. 81, pp. 148–176, 1995.
[63] Wang D.L. and Terman D. Image segmentation based on oscillatory correlation.

Neural Comput, Vol. 9, pp. 805–836, 1997.

[64] Wang D.L. Object selection based on oscillatory correlation. *Neural Netw*, Vol. 12, pp. 579–592, 1999.

[65] Ando H, Miyake M, Morie T, Nagata M, and Iwata A. A nonlinear oscillator network circuit for image segmentation with double-threshold phase detection. *ICANN1999*, Vol. 2, pp. 655–660, 1999.

[66] Kazanovich Y and Borisyuk R. Object selection by an oscillatory neural network. *Biosystems*, Vol. 67, pp. 103–111, 2002.

[67] Hubel D.H. and Wiesel T.N. Receptive fields of single neurones in the cat's striate cortex. *J Physiol*, Vol. 148, pp. 574–591, 1959.

[68] Hubel D.H. and Livingstone M.S. Color and contrast sensitivity in the lateral geniculate body and primary visual cortex of the macaque monkey. *J Neurosci*, Vol. 10, pp. 2223–2237, 1990.

[69] Conway B.R., Hubel D.H., and Livingstone M.S. Color contrast in macaque V1. *Cereb Cortex*, Vol. 12, pp. 915–925, 2002.

[70] Neisser U. Cognitive psychology. *New York : Appleton-Century-Crofts.*, 1967.

[71] Treisman A.M. and Gelade G. A feature-integration theory of attention. *Cogn Psychol*, Vol. 12, pp. 97–136, 1980.

[72] Treisman A.M. and Nancy G.K. Perceiving visually presented objects: recognition, awareness, and modularity. *Curr Opin Neurobiol*, Vol. 8, pp. 218–226, 1998.

[73] Fukushima K. Neocognitron: a self organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biol Cybern*, Vol. 36, No. 4, pp. 193–202, January 1980.

[74] Riesenhuber M and Poggio T. Hierarchical models of object recognition in cortex. *Nat Neurosci*, Vol. 2, No. 11, pp. 1019–25, December 1999.

[75] Grossberg S and Huang T. ARTSCENE: A neural system for natural scene classification. *J Vision*, Vol. 9, pp. 1–19, 2009.

[76] Ben-Shahar O and Zucker S. Geometrical computations explain projection patterns of long-range horizontal connections in visual cortex. *Neural Comput*, Vol. 16, No. 3, pp. 445–76, March 2004.

[77] Tanaka J.W. and Farah M.J. Parts and wholes in face recognition. *Q J Exp Psychol-A*, Vol. 46, No. 2, pp. 225–45, May 1993.

[78] Näsänen R. Spatial frequency bandwidth used in the recognition of facial images. *Vision Res*, Vol. 39, No. 23, pp. 3824–33, November 1999.

[79] Drucker D.M. and Aguirre G.K. Different spatial scales of shape similarity representation in lateral and ventral LOC. *Cerebral cortex (New York, N.Y. : 1991)*, Vol. 19, No. 10, pp. 2269–80, October 2009.

[80] Lamdan Y and Wolfson H. Geometric hashing: A general and efficient model-based recognition scheme. *IEEE I Conf Comp Vis*, pp. 238–249, 1988.

[81] Sprinzak J and Werman M. Affine point matching. *Pattern Recogn Lett*, Vol. 15, No. 4, pp. 337–339, April 1994.

[82] Tarr M.J. and Bülthoff H.H. Image-based object recognition in man, monkey and machine. *Cognition*, Vol. 67, No. 1-2, pp. 1–20, July 1998.

[83] Richards W, Dawson B, and Whittington D. Encoding contour shape by curvature extrema. *J Opt Soc Am A*, Vol. 3, No. 9, pp. 1483–91, October 1986.

[84] Mokhtarian F and Mackworth A. Scale-Based Description and Recognition of Planar Curves and Two-Dimensional Shapes. *IEEE Trans Pattern Anal Mach Intell*, Vol. PAMI-8, No. 1, pp. 34–43, January 1986.

[85] Mokhtarian F and Mackworth AK. A theory of multiscale, curvature-based shape representation for planar curves, 1992.

[86] Mokhtarian F. Silhouette-based isolated object recognition through curvature scale space. *IEEE Trans Pattern Anal Mach Intell*, Vol. 17, No. 5, pp. 539–544, May 1995.

[87] Mokhtarian F. Shape similarity retrieval under affine transforms. *Pattern Recogn*, Vol. 35, No. 1, pp. 31–41, January 2002.

[88] Mai F and Chang C.Q. Affine-invariant shape matching and recognition under partial occlusion. *IEEE ICIP, 2010 17th*, pp. 4605–4608, 2010.

[89] Mokhtarian F. Silhouette-based occluded object recognition through curvature scale space. *Machine Vision and Applications*, Vol. 10, No. 3, pp. 87–97, August 1997.

[90] Cui M, Femiani J, Hu J, Wonka P, and Razdan A. Curve matching for open 2D curves. *Pattern Recogn Lett*, Vol. 30, No. 1, pp. 1–10, January 2009.

[91] Belongie S, Malik J, and Puzicha J. Shape matching and object recognition using shape contexts. *IEEE Trans Pattern Anal Mach Intell*, Vol. 24, No. 4, pp. 509–522, April 2002.

[92] Latecki LJ and Lakamper R. Shape descriptors for non-rigid shapes with a single closed contour. *Proc CVPR IEEE*, pp. 424–429.

[93] Mokhtarian F, Abbasi S, and Kittler J. Efficient and robust retrieval by shape content through curvature scale space. *Ser Softw Engn Knowl*, Vol. 8, pp. 51–58, 1997.

[94] Attalla E and Siy P. Robust shape similarity retrieval based on contour segmentation polygonal multiresolution and elastic matching. *Pattern Recogn*, Vol. 38, No. 12, pp. 2229–2241, December 2005.

[95] Yang X and Koknar-Tezel S. Locally constrained diffusion process on locally densified distance spaces with applications to shape retrieval. *Proc CVPR IEEE*, pp. 357 – 364, 2009.

[96] Asada H and Brady M. The curvature primal sketch. *IEEE T Pattern Anal*, No. 1, pp. 2–14, 1986.

[97] Daliri M.R. and Torre V. Classification of silhouettes using contour fragments. *Comput Vis Image Und*, Vol. 113, No. 9, pp. 1017–1025, September 2009.

[98] Diplaros A, Petrakis E, and Milios E. Shape Matching with Occlusion in Image Databases. *Infotech Oulu International Workshop on Information Retrieval, IR'2001, September 19-21*, pp. 142–150, 2001.

[99] Kumar N, Belhumeur PN, and Biswas A. Leafsnap: A computer vision system for automatic plant species identification. *Computer Vision – ECCV: Lecture Notes in Computer Science 2012, pp 502-516*, 2012.

[100] Chang C.C., Hwang S.M., and Buehrer D.J. A shape recognition scheme based on relative distances of feature points from the centroid. *Pattern Recogn*, Vol. 24, pp. 1053–1063, 1991.

[101] Tan K.L., Ooi B.C., and Thiang L.F. Retrieving Similar Shapes Effectively and Efficiently. *Multimedia Tools and Applications*, Vol. 19, No. 2, pp. 111–134, 2003.

[102] Jung B.K., Shin S.Y., Wang W, Choi H.D., and Pack J.K. Similar MRI object retrieval based on modified contour to centroid triangulation with arc difference rate. *SAC2014*, pp. 31–32, 2014.

[103] Shih J.L. and Lin S.Y. A new shape retrieval approach based on the multi-resolution Contour-based descriptor. *Journal of Information Technology and Applications*, Vol. 6, No. 2, pp. 40–51, 2012.

[104] Buffart H and Leeuwenberg E. Coding theory of visual pattern completion. *J Exp Psychol Hum Percept Perform*, Vol. 7, No. 2, pp. 241–274, April 1981.

[105] Fukushima K. Neural network model for completing occluded contours. *Neural*

Netw, Vol. 23, No. 4, pp. 528–40, May 2010.

[106] Fukushima K. Artificial vision by multi-layered neural networks: Neocognitron and its advances. *Neu Net*, Vol. 37, pp. 103–119, January 2013.

[107] Wei H, Ren Y, and Wang Z.Y. A computational neural model of orientation detection based on multiple guesses: comparison of geometrical and algebraic models. *Cognitive Neurodynamics*, Vol. 7, No. 5, pp. 361–379, 2012.

[108] Raizada R and Grossberg S. Towards a theory of the laminar architecture of cerebral cortex: computational clues from the visual system. *Cereb Cortex*, Vol. 13, No. 1, pp. 100–13, January 2003.

[109] Grossberg S. How does the cerebral cortex work? Development, learning, attention, and 3-D vision by laminar circuits of visual cortex. *Behav Cogn neurosci rev*, Vol. 2, No. 1, pp. 47–76, March 2003.

[110] Pasupathy A. Responses to contour features in macaque area V4. *J Neurophysiol*, Vol. 82, No. 5, pp. 2490–2502, November 1999.

[111] Cadieu C, Kouh M, Pasupathy A, Connor C.E., Riesenhuber M, and Poggio T. A model of V4 shape selectivity and invariance. *J Neurophysiol*, Vol. 98, No. 3, pp. 1733–50, September 2007.

[112] Connor C.E., Brincat S.L., and Pasupathy A. Transformation of shape information in the ventral pathway. *Current opinion in neurobiology*, Vol. 17, No. 2, pp. 140–7, April 2007.

[113] Kourtzi Z and Connor C.E. Neural Representations for Object Perception: Structure, Category, and Adaptive Coding. *Annual Review of Neuroscience*, Vol. 34, pp. 45–67, March 2011.

[114] Yau J.M., Pasupathy A, Brincat S.L., and Connor C.E. Curvature Processing Dynamics in Macaque Area V4. *Cereb Cortex*, January 2012.

[115] Weyland N.B.J. and Prasad R. Characterisation of line drawings using generalised Fourier descriptors. *Electronics Letters*, Vol. 26, No. October, pp. 1794–1795, 1990.

[116] Yang L and Prasad R. Recognition of line-drawings based on generalised Fourier descriptors. *International Conference on Image Processing and its Applications*, No. 6, pp. 286–289, 1992.

[117] Gdalyahu Y and Weinshall D. Flexible syntactic matching of curves and its application to automatic hierarchical classification of silhouettes. *IEEE Trans Pattern Anal Mach Intell*, Vol. 21, No. 12, pp. 1312–1328, 1999.

[118] Bengtsson A and Eklundh J. Shape representation by multiscale contour approximation. *IEEE Trans Pattern Anal Mach Intell*, Vol. 13, No. 1, pp. 85–93, 1991.

[119] Gope C, Kehtarnavaz N, Hillman G, and Würsig B. An affine invariant curve matching method for photo-identification of marine mammals. *Pattern Recogn*, Vol. 38, No. 1, pp. 125–132, January 2005.

[120] Liu H.C. and Srinath M.D. Partial shape classification using contour matching in distance transformation. *IEEE Trans Pattern Anal Mach Intell*, Vol. 12, No. 11, pp. 1072–1079, November 1990.

[121] Borgefors G. Hierarchical chamfer matching: a parametric edge matching algorithm. *IEEE Trans Pattern Anal Mach Intell*, No. 6, pp. 849–865, November.

[122] Raphael C. Coarse-to-fine dynamic programming. *IEEE Trans Pattern Anal Mach Intell*, Vol. 23, No. 12, pp. 1379–1390, December 2001.

[123] Ueda N and Suzuki S. Learning visual models from shape contours using multiscale convex/concave structure matching. *IEEE Trans Pattern Anal Mach Intell*, Vol. 15, No. 4, pp. 337–352, April 1993.

[124] Basri R, Costa L, Geiger D, and Jacobs D. Determining the similarity of de-

formable shapes. *Vision research*, Vol. 38, No. 15-16, pp. 2365–85, August 1998.
[125] Rao Z, Petrakis E, and Milios E. Efficient retrieval of deformed and occluded shapes. *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, Vol. 4, pp. 67–71, 2000.
[126] Petrakis E, Diplaros A, and Milios E. Matching and retrieval of distorted and occluded shapes using dynamic programming. *IEEE Trans Pattern Anal Mach Intell*, Vol. 24, No. 11, pp. 1501–1516, November 2002.
[127] Schmidt F.R., Farin D, and Cremers D. Fast matching of planar shapes in sub-cubic runtime. *Computer Vision, 2007. ICCV*, pp. 1–6, 2007.
[128] Felzenszwalb P.F. and Zabih R. Dynamic Programming and Graph Algorithms in Computer Vision. *IEEE Trans Pattern Anal Mach Intell*, Vol. 33, No. 4, pp. 721–740, April 2011.
[129] Wang J, Bai X, You X, Liu W, and Latecki L.J. Shape matching and classification using height functions. *Pattern Recognition Letters*, Vol. 33, No. 2, pp. 134–143, January 2012.
[130] Pajdla T and Gool L.V. Matching of 3-D Curves using Semi-differential Invariants. *Computer Vision, 1995. Proceedings., Fifth International Conference on*, pp. 390–395, 1996.
[131] Lamdan Y, Schwartz J.T., and Wolfson H.J. Affine invariant model-based object recognition. *IEEE Trans Rob Aut*, Vol. 6, No. 5, pp. 578–589, 1990.
[132] Ballard D.H. Generalizing the Hough Transform to Detect Arbitrary Shapes. *Pattern Recogn*, Vol. 13, No. 2, pp. 111–122, September 1981.
[133] Ma L, Liu Y, Jiang H, Wang Z, and Zhou H. An Improved Method of Geometric Hashing Pattern Recognition. *International Journal of Modern Education and Computer Science*, Vol. 3, No. 3, pp. 1–7, 2011.
[134] Xudong L and Wang Z.J. Perceptual Image Hashing Based on Shape Contexts and Local Feature Points. *Information Forensics and Security, IEEE Transactions on*, Vol. 7, No. 3, pp. 1081–1093, 2012.
[135] Zao Y, Wang S, Zhang X, and Yao H. Robust Hashing for Image Authentication Using Zernike Moments and Local Features. *Information Forensics and Security, IEEE Transactions on*, Vol. 8, No. 1, pp. 55–63, 2013.
[136] Manay S, Cremers D, Hong B.W., Yezzi A.J., and Soatto S. Integral invariants for shape matching. *IEEE Trans Pattern Anal Mach Intell*, Vol. 28, No. 10, pp. 1602–18, October 2006.
[137] Yang G, Yan Z, and Zhao H. Touching String Segmentation Using MRF. *2009 International Conference on Computational Intelligence and Security*, No. 1, pp. 520–524, 2009.
[138] Mehrotra H, Majhi B, and Gupta P. Robust iris indexing scheme using geometric hashing of SIFT keypoints. *J Network and Computer Appli*, Vol. 33, No. 3, pp. 300–313, 2010.
[139] Jayaraman U, Gupta A.K., Prakash S, and Gupta P. An Enhanced Geometric Hashing. *2011 IEEE International Conference on Communications (ICC)*, No. 5-9 June 2011, pp. 1–5, 2011.
[140] http://www.l-3mps.com (2015.07.27).
[141] http://www.mapmart.com (2015.05.08).