Author: Wayne, Lin Wei-Cheng
Advisor: Prof. Wu Ja-Ling

Mathematical Morphology and Its Applications on Image Segmentation

June 7, 2000

Dept. of Computer Science and Information Engineering,
National Taiwan University

Abstract

Mathematical morphology provides a systematic approach to analyze the geometric characteristics of signals or images, and has been applied widely to many applications such as edge detection, object segmentation, noise suppression and so on. Thus, the main purpose of this thesis is to provide an overview of mathematical morphology and review some morphological filters which are widely used in image processing. Furthermore, a morphology-based supervised segmentation system is proposed.

In chapter 2, we first review the basic geometric characteristics of the primitive morphology operators. Some examples of their applications on signal processing are also illustrated. Chapter 3 is devoted to the watershed transformation and the morphological gradient operators that are constructed on the basis of the underlying morphological operations. The morphological supervised segmentation system is described in chapter 4, including the implementation detail and parameter settings. Experimental results can be found in chapter 5 and some discussion of the simulation results are also included. Chapter 6 will represent the conclusion of this thesis and our future works.

Contents

Chapter 1

Introduction

1.1 Motivation

In traditional popular coding standards, like MPEG-1, MPEG-2, and H.263, each frame in an image sequence is represented as rectangular arrays of pixels, and it is partitioned into square blocks as the basic coding units. This coding technique is the so-called *block-based compression*. They achieve high compression ratios and are suitable for a wide range of applications, such as video conference, and efficient multimedia data storage. Second-generation video coding techniques have been proposed, in which the frames are segmented into arbitrarily shaped regions according to a specific criterion of coherence rather than in fixed-sized square blocks. Opposite to block-based compression, it is referred as *region-based compression*. Either block-based or region-based compression is still a low-level technique without a proper mapping to the real world or human visual information. Hence, the latest coding standard MPEG-4 [1] involves a distinct feature called *object-based coding* where audio and video contents are decomposed into objects that may represent a fixed background, a talking man, his voice, and background music. Object-based coding technique can not only improve the coding efficiency but also make it possible to interact among the objects in the images. That is, the user can manipulate a semantic object in a image or observe an object from distinct views.

From the perspective of multimedia data representation, object-based video representation provides a natural framework. Significant efforts have been devoted to object-oriented functionalities, including content-based indexing, content-based retrieval and content browsing systems. For instance, content-based indexing and

retrieval involve a structuring of data in terms of low-level regions, semantic objects and so on. These features are addressed in the ongoing standard MPEG-7 and MPEG-21. In summary, video object segmentation has become a must-do preprocessing in view of multimedia data compression and representation.

1.2 Supervised v.s. Unsupervised Segmentation

Image segmentation has been an important and challenging issue in the field of computer vision over decades, and it plays a critical role for most image analysis tasks, such as object recognition, object-based image compression and content-based indexing. Lots of techniques and algorithms have been proposed, and most of them are dedicated to the development of fully automatic segmentation, i.e. unsupervised segmentation. However, it still lacks a robust and unified algorithm that can be applied to all kinds of images.

Fortunately, the algorithms in the unsupervised domain could accomplish the low-level region segmentation. That is, an image can be partitioned into homogeneous regions according to a given quantitative criteria, such as gray level, color, texture or combination of them. Opposite to the homogeneous regions, an semantic object usually lacks global coherence, but is constituted by one or more regions. However, due to the ambiguity of "semantic", it is very difficult for the computer to automatically group the regions into a semantic meaningful object.

Since only the human knows the really meaning of "semantic", the user interaction could help the computer to group the low-level regions into a object. On the other hand, the computer helps the human to precisely locate the boundary. This is the general concept of supervised segmentation. While lacking high level image understanding to grasp the meaning of "semantic", the adoption of some form of user assistance for image segmentation becomes commonly accepted. In this thesis, a supervised segmentation system is proposed.

1.3 Previous Work

As mentioned earlier, there exist many algorithms for automatic image and video segmentation. Robert M. Haralick and Linda G. Shapiro classified the image segmentation techniques as: measurement space guided spatial clustering, single

linkage region growing schemes, hybrid linkage region growing schemes, centroid linkage region growing schemes, spatial clustering schemes, and split and merge schemes [2]. Nikhil R. Pal and Sankar K. Pal reviewed some traditional image segmentation techniques and included fuzzy set theoretic segmentation and neural network based techniques [3]. Furthermore, most algorithms put emphasize on grayscale image, and some can extend to color image. In this section, we will focus on the algorithms for color image segmentation since it is the majority of image sequences.

Clustering-based algorithms have shown its capability on color image segmentation [4][5][6]. In [4], the k-means algorithm, a classical theme of supervised cluster analysis in statistical pattern recognition, is modified for unsupervised segmentation of color image in the chromaticity plane. It first makes use of only the chromatic information and performs 2D k-means algorithm in the $u'v'$ plane. Then the chromaticity clusters are associated with appropriate intensity value through the 1D version of the k-means algorithm. In [5], the proposed method detects image cluster in some linear decision volumes of the L^*, a^*, b^* uniform color space. The detected clusters are projected onto the line of the Fischer discriminant for 1D thresholding. This algorithm is efficient because clusters are specified in linear decision volumes using only 1D histograms. A procedure similar to vector quantization is borrowed for cluster detection and a local feature analysis technique is also proposed in [6]. Some techniques based on mean-shift algorithm which is a simple nonparametric procedure for estimating density gradients have also been proposed [7][8].

Another trend of image segmentation is the fusion of different features to get a more reliable result [9][10]. In [9], the adaptive bayesian color segmentation based on Gibbs random field is performed first to form a segmentation map. At the same time, the edge map is built up by the color edge detector. Then regions in the segmentation map are split and merged by a region-labeling procedure to enforce their consistency with the edge map. Beside color information, motion is another useful feature for an image sequence. In [10], motion segmentation and color segmentation are integrated to get a promising result for extracting the moving vehicles on the road.

In the domain of supervised segmentation, snakes [11] had attracted lots of discussion. A snake is an energy-minimizing spline guided by external constraint

forces and influenced by image forces that pull it toward features such as lines and edges. In [12], it extracts regions based on the multidimesional analysis of several image features by a spatially constrained fuzzy C-means algorithm. In the semantic step, the user add the points on the top of the semantic object.

Many morphological approaches for image segmentation have also been proposed. A powerful morphological tool for image segmentation is watershed transformation [13][14]. Applying watershed transformation on gradient image could obtain an initial segmentation, but usually yields an over-segmented result. Thus, region merging can be performed as the post-processing, and efficient algorithms have been proposed [15]. A morphological clustering method is illustrated in [16]. The histogram of RGB information in the color image is built up and then using the geometric property of closing to find out the dominant clusters. A morphology-based supervised segmentation is introduced in [17][18]. Its main concept is to create searching area where the real object resides by applying the morphological operators to the initial boundary given by the user. This thesis is based on this idea and proposes an adaptive morphological operators. Moreover, mathematical morphology also shows its power in video segmentation [19][20].

Actually, there already exist some systems combining the video object segmentation and content-based retrieval [21][22][23]. Their main concept is to segment the first frame into homogeneous regions and form the semantic object defined by the user. In following frames, the semantic object is tracked by motion estimation and region aggregation.

1.4 Thesis Outline

The main purposes of this thesis are :

1. to provide an overview of mathematical morphology in view of its geometric characteristics and signal processing.

2. to review some morphological filters which are widely used in image processing.

3. to propose a morphology-based supervised segmentation system.

Hence, in chapter 2, we first review the basic geometric characteristics of the primitive morphology operators. Some examples of its applications on signal pro-

cessing are also illustrated. Chapter 3 is devoted to the watershed transformation and the morphological gradient operators that are constructed on the basis of the underlying morphological operations. The proposed morphological supervised system based on the concept in [18] is described in chapter 4, including the implementation detail and parameter settings. Experimental results can be found in chapter 5 and some discussion of the simulation results are also included. Chapter 6 will represent the conclusion of this thesis and its future work.

Chapter 2

Mathematical Morphology

From a general scientific perspective, the word *morphology* refers to the study of forms and structures. In image processing, morphology is the name of a specific methodology for analyzing the geometric structure inherent within an image. The *morphological filter*, which can be constructed on the basis of the underlying morphological operations, are more suitable for shape analysis than the standard linear filters since the latter sometimes distort the underlying geometric form of the image. Some of the salient points regarding the morphological approach are as follows [24] :

1. Morphological operations provide for the systematic alteration of the geometric content of an image while maintaining the stability of the important geometric characteristics.

2. There exists a well-developed morphological algebra that can be employed for representation and optimization.

3. It is possible to express digital algorithms in terms of a very small class of primitive morphological operations.

4. There exist rigorous representation theorems by means of which one can obtain the expression of morphological filters in terms of the primitive morphological operations.

Generally speaking, the morphological operators transform the original image into another image through the interaction with the other image of certain shape and size, which is known as the *structure element*. Geometric feature of the image

that are similar in shape and size to the structure element are preserved, while other features are suppressed. Therefore, morphological operations can simplify the image data, preserving their shape characteristics and eliminate irrelevancies. In view of applications, morphological operations can be employed for many purposes[25], including edge detection [26][27], segmentation [17][14][16][28], enhancement [29] of images and so on.

This chapter begins with binary morphology that is based on set theory. The following grayscale morphology can be regarded as the extension of binary morphology to the three-dimensional space since a grayscale image can be considered as a set of points in 3D space. While binary morphology and grayscale morphology are well-developed and widely used, it is not straightforward to extend mathematical morphology to color images. The part of color morphology in our study comes mainly from [30] and simply introduces the approaches for extension of mathematical morphology to color images. The basic geometric characteristics of the primitive morphology operators are introduced in this chapter while the detail of morphological algebra goes beyond the scope of the thesis. A systematic introduction of theoretical foundations of mathematical morphology, its main image operations, and their applications can be found in [25], [24],[31].

Mathematical morphology defined on Euclidean setting is called Euclidean morphology and that defined on digital setting is called digital morphology. In general, their relationship is akin to that between continuous signal processing and digital signal processing. Although the actual implementation of morphological operators will be in the digital setting, the Euclidean model is essential to the development of an understanding of and intuitive feel of how the operators function in theory and application.

2.1 Binary Morphology

The theoretical foundation of binary mathematical morphology is *set theory*. In binary images, those points in the set are called the foreground and those in the complement set are called the background.

Besides dealing with the usual set-theoretic operations of union and intersection, morphology depends extensively on the *translation* operation. For convenience, $\bigcup$ denotes the set-union, $\bigcap$ denotes set-intersection and $+$ inside the set

notation refers to vector addition in the following equations.

Definition 2.1 (Translation) *Given an image A, the translation of A by the point x, denoted by A_x, is defined by*

$$A + x = \{a + x : a \in A\} \tag{2.1}$$

.

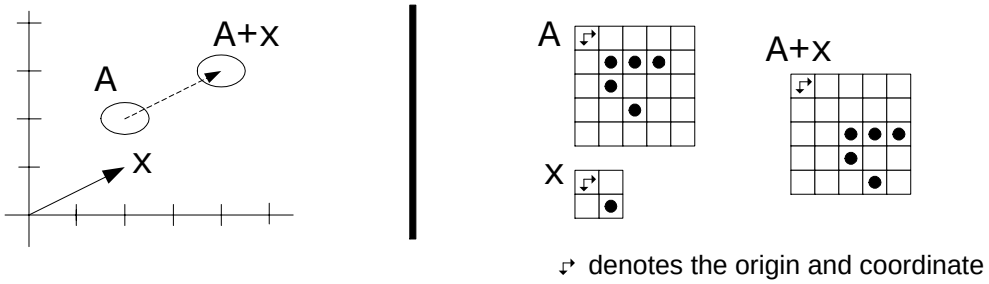


Figure 2.1: Translation operation on Euclidean(left) and digital(right) setting

Figure 2.1 is an example of the translation operation. In the left part, A is an ellipse centered at (2,2) and x is a vector of (2,1). Translation of A by x will shift A with the displacement (2,1) such that A is centered at (4,3). In the right part, A is an image in Z^2, with $\{(1,1),(2,1),(3,1),(1,2),(3,4)\}$ and x is a vector of (1,1). Translation of A by x results in $\{(2,2),(3,2),(4,2),(2,3),(4,5)\}$.

2.1.1 Binary Dilation

Definition 2.2 (Dilation) *Dilation of a binary image A by structure element B, denoted by $A \oplus B$, is defined as*

$$A \oplus B = \{a + b \mid for\ a\ \in A\ and\ b \in B\}. \tag{2.2}$$

From Eqn. (2.2), dilation is equivalent to a union of translates of the original image with respect to the structure element:

$$A \oplus B = \bigcup_{b \in B} A_b \tag{2.3}$$

.

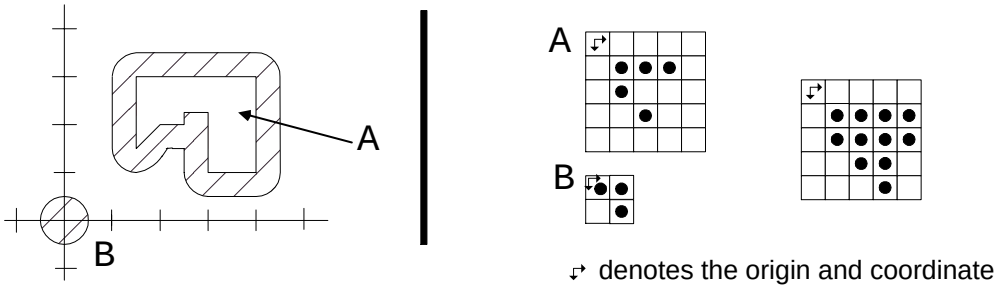


Figure 2.2: Dilation on Euclidean(left) and digital(right) setting

Intuitively, the structure element plays the role of a template. Dilation is found by placing the center of the template over each of the foreground pixels of the original image and then taking the union of all the resulting copies of the structure element, produced by using the translation operation. From Fig. 2.2, you can see how dilation modifies the original image with respect to the shape of the structure element. In general, dilation has the effect of "expanding" an image, and hence, the small hole can be eliminated.

2.1.2 Binary Erosion

Definition 2.3 (Erosion) *Erosion of a binary image A by structure element B, denoted by $A \ominus B$, is defined as:*

$$A \ominus B = \{p \mid p + b \in A \ \ \forall b \in B\}. \tag{2.4}$$

Whereas dilation can be represented as a union translates, erosion can be represented as an intersection of the negative translates:

$$A \ominus B = \bigcap_{b \in B} A_{-b} \tag{2.5}$$

where -b is the scalar multiple of the vector b by -1.

Like dilation, the erosion of the original image by the structure element can be described intuitively by template translation. The template is moved across the original image. For a given foreground pixel, put the center of the template onto it, i.e. translate the template to that pixel. If the translation of the template is a subimage of the original image, that pixel is activated in erosion; otherwise, it is

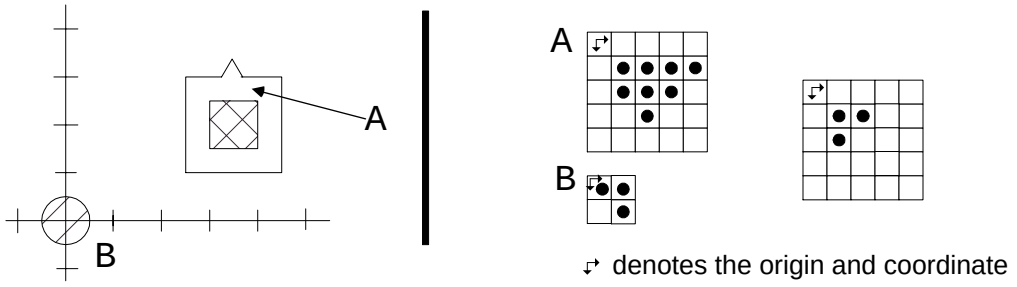


Figure 2.3: Erosion on Euclidean(left) and digital(right) setting

not activated. Figure 2.3 shows that erosion "shrinks" the original image and the small peak is eliminated.

2.1.3 Binary Opening

Definition 2.4 (Opening) *Opening of a binary image A by structure element B, denoted by $A \circ B$, is defined as*

$$A \circ B = (A \ominus B) \oplus B. \tag{2.6}$$

From the definition, the original image is eroded first and then dilated. Therefore, it can intuitively be thought as "rolling the structure element about the inside boundary of the image". The following definition gives a rigorous set-theoretic characterization of this "fitting" property. It states that the opening of A by B is obtained by taking the union of all the translates of B which fit into A.

Definition 2.5 (Opening(2))

$$A \circ B = \bigcup \{B + x \mid B + x \subset A\}. \tag{2.7}$$

Figure 2.4 shows how this original image is smoothed and the spot-like noise is removed because the disk can't fit into them. It is worth noticing that smoothing effect of the object boundary highly depends on the shape of the structure element.

2.1.4 Binary Closing

Definition 2.6 (Closing) *Closing of a binary image A by structure element B, denoted by $A \bullet B$, is defined as*

$$A \bullet B = (A \oplus B) \ominus B. \tag{2.8}$$

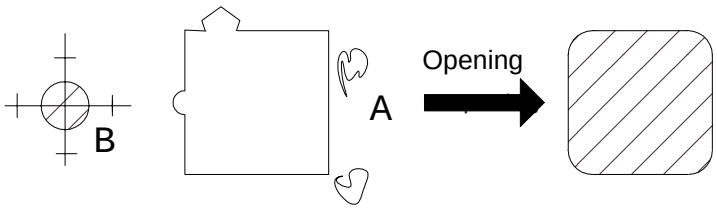


Figure 2.4: Opening

The closing of the original image includes all points satisfying the condition that anytime the point can be covered by a translation of the structure element, there must be some point in common between the translated structure element and the original image.

Definition 2.7 (Closing(2)) *z is an element of* $A \bullet B$ *if and only if* $(B+y) \bigcap A \neq \emptyset$*, for any translate* $(B + y)$ *containing z.*

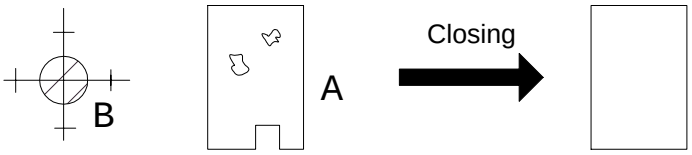


Figure 2.5: Closing

As stated in Eqn. (2.8), closing is done by first dilating the image and then eroding it. Hence, instead of eliminating the small peaks, it will "fill" the holes, as shown in Fig. 2.5. In other words, it has the effect of "clustering" each spatial point to a connected set.

2.2 Gray-scale Morphology

In intuition, a 2D grayscale image can be thought of a set of points in 3D space, $p = (x, y, f(x, y))$, (f(x,y) is the function to represent the grayscale image). By

applying the *umbra transform* **U**, a 2D grayscale image can be transformed as a 3D binary image. Therefore, grayscale morphological operators may be regarded as the extension of binary morphological operators to three-dimensional space.

Definition 2.8 (Umbra Transform) *Given a signal f, the umbra transform of f, denoted as* **U***[f], is defined as:*

$$\mathbf{U}[f] = \{(x, y) : x \in D_f \text{ and } y \leq f(x)\} \tag{2.9}$$

Definition 2.9 (Top Surface) *Given an umbra A, we define the top surface of A, denoted as* **S***[A], to be the set of all points (x,y) such that x is in the domain of A and* $y = sup\{\ y : (x, y) \in A\}$.

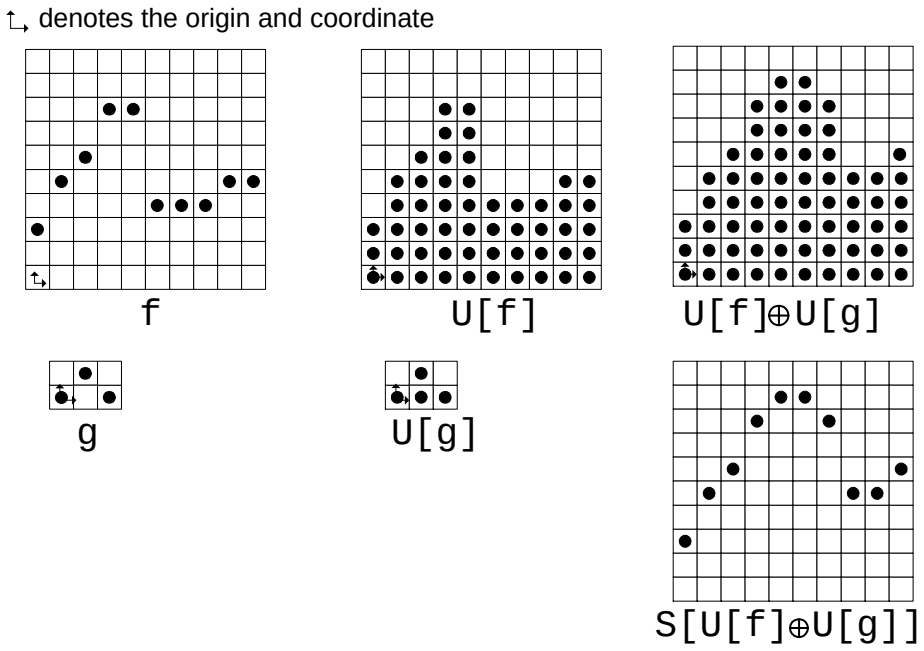


Figure 2.6: Grayscale Dilation based on Umbra transform and Surface in digital setting

Theoretically, given two signals f and g, dilation of f by g can be computed as $\mathbf{S}[\mathbf{U}[f] \oplus \mathbf{U}[g]]$ where $\oplus$ is the binary dilation operator (c.f. Fig. 2.6).

However, although the umbra transform tells us conceptually how to compute grayscale morphology from binary morphology, the mathematical expression is still necessary for implementation. Two operators are employed for defining the grayscale mathematical morphology operators, EXTSUP and INF.

Definition 2.10 (EXTSUP) *Given a collection of signals* $\{f_k\}$*, we define*

$$[EXTSUP(f_k)](t) = \begin{cases} sup[f_k(t)], & \text{if there exists at least one } k \text{ such that } f_k \text{is defined at } t, \\ & \text{and where the supremum is over all such } k \\ undefined, & \text{if } f_k(t) \text{ is undefined for all } k \end{cases}$$

Definition 2.11 (INF) *Given a collection of signals* $\{f_k\}$*, we define*

$$[INF(f_k)](t) = \begin{cases} INF[f_k(t)], & \text{if } f_k(t) \text{ is defined for all } k \\ undefined, & otherwise \end{cases}$$

For any signal f, with domain D_f, and one point x, we define the translation f_x by $f_x(t) = f(t - x)$. That is, f_x is f translated x units to the right. Hence, $D_{f_x} = D_f + x$. In addition, $f(t) + y$ translates f up y units and possesses the same domain. Furthermore, given two signals f and g, $f \ll g$ means that D_f is the subset of D_g, and for any t in D_f, $f(t) \leq g(t)$.

2.2.1 Grayscale Dilation

Definition 2.12 (Grayscale Dilation) *For signals f and g, with respective domain* D_f *and* D_g*, we define the dilation of f by g as*

$$\mathbf{D}(f, g) = EXTSUP_{x \in D_f}[g_x + f(x)] \tag{2.10}$$

Geometrically, the dilation is obtained by taking an extended supremum of all copies of g that have been translated over x units and up f(x) units. As binary dilation, g plays the role of template. In grayscale dilation, for each point of f, shift g so that its center coincides with (x,f(x)) and EXTSUP is applied to the resulting copies of g.

In Fig. 2.7 (f is the original signal and g is the structure element), f has two points of discontinuity, and the result shows two features of dilation. First, the discontinuity in the dilation due to the peak of the original signal demonstrates the sensitivity of dilation to functional changes at a point of "jump". Second, the "hole" of the original signal can be eliminated in the dilation by nature of "supremum" operation. Furthermore, the domain of Dilate(f,g) is also dilated according to the domain of g.

In implementation, the "supremum" operation in Eqn. (2.10) is replaced by "maximum", and furthermore, Eqn. (2.10) can be re-written as follows.

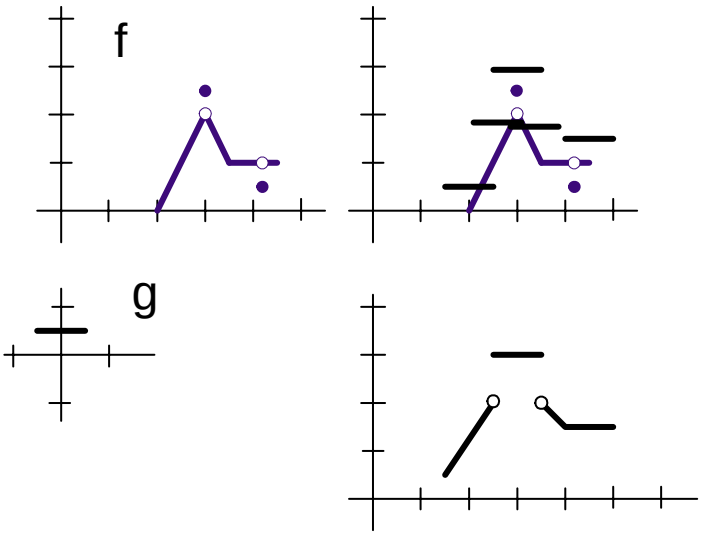


Figure 2.7: Grayscale Dilation

Definition 2.13 (Grayscale Dilation(2)) *Dilation of a grayscale image f(r.c) by a grayscale structure element g(r,c) is defined as*

$$\mathbf{D}(f,g)(r,c) = max_{(i,j)}(f(r-i,c-j)+g(i,j)) \tag{2.11}$$

2.2.2 Grayscale Erosion

Definition 2.14 (Grayscale Erosion) *For signals f and g, with respective domain D_f and D_g, we define the erosion of f by g as*

$$[\mathbf{E}(f,g)](x) = sup\{y : g_x + y \ll f\} \tag{2.12}$$

To find the value of the erosion of f by g at the point x, we shift g so that it is centered at x and then find out the largest vertical translation y that will leave $g_x + y$ beneath f. It is analogous to the "fitting" property in binary erosion. Eqn. (2.12) can be expressed in terms of INF operation as:

$$\mathbf{E}(f,g) = INF_{x \in D_g}[f_{-x} - g(x)] \tag{2.13}$$

Figure 2.8 illustrates the effect of the erosion. Notice the eroding of the graphs of the signals and also the domain. It is due to the $\ll$ operator defined in Eqn. (2.12) or the INF operator in Eqn. (2.13).

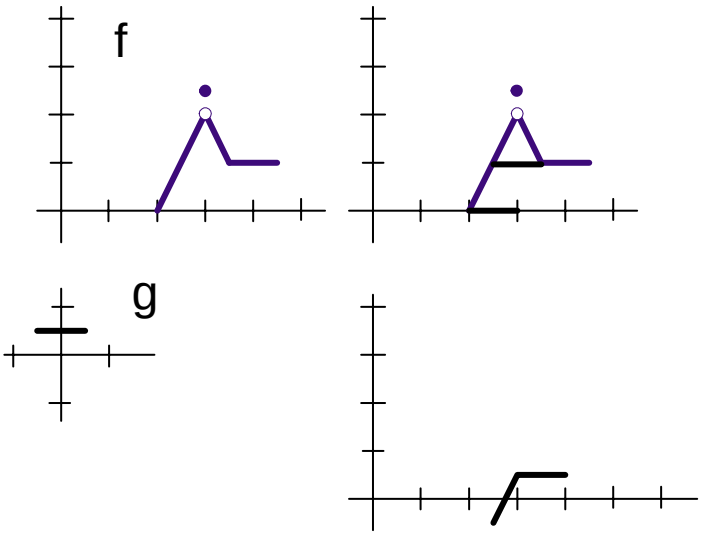


Figure 2.8: Grayscale Erosion

As grayscale dilation, by replacing "infimum" with "minimum", Eqn. (2.13) can be re-written as follows:

Definition 2.15 (Grayscale Erosion(2)) *Erosion of a grayscale image f(r.c) by a grayscale structure element g(r,c) is defined as*

$$\mathbf{E}(f,g)(r,c) = min_{(i,j)}(f(r+i,c+j) - g(i,j)) \tag{2.14}$$

2.2.3 Grayscale Opening

Definition 2.16 (Grayscale Opening) *For signals f and g, we define the opening of f by g as*

$$\mathbf{O}(f,g) = \mathbf{D}[\mathbf{E}(f,g),g] \tag{2.15}$$

Figure 2.9 shows how grayscale opening eliminates the peak of the signal.

2.2.4 Grayscale Closing

Definition 2.17 (Grayscale Closing) *For signals f and g, we define the closing of f by g as*

$$\mathbf{C}(f,g) = \mathbf{E}[\mathbf{D}(f,g),g] \tag{2.16}$$

Figure 2.9 shows how grayscale closng fills the hole of the signal.

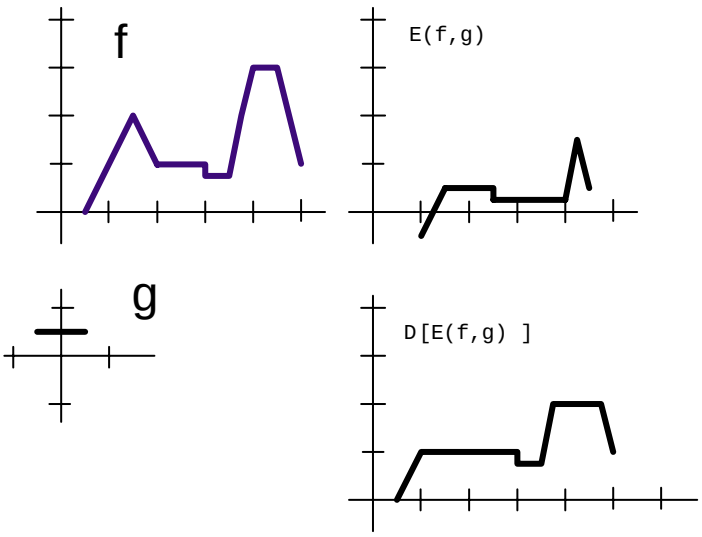


Figure 2.9: Grayscale Opening

2.3 Color Morphology

After introducing binary and grayscale morphology, the task turns on dealing with the color image morphology. However, the extension of mathematical morphology to color images is not straightforward. In [30], it investigates two approaches for color morphology: a vector approach and a component-wise approach.

In component-wise approach, the grayscale morphological operator is applied to each channel of the color image. For example, component-wise color dilation of $f(x,y) = [f_R(x,y), f_G(x,y), f_B(x,y)]^T$ by the structure element $h(x,y) = [h_R(x,y), h_G(x,y), h_B(x,y)]^T$ in RGB color space is defined as:

$$(f \oplus_c h)(x,y) = [(f_R \oplus h_R)(x,y), (f_G \oplus h_G)(x,y), (f_B \oplus h_B)(x,y)]^T \qquad (2.17)$$

where the symbol $\oplus_c$ represents component-wise dilation and $\oplus$ on the right-hand side is grayscale dilation. Component-wise color erosion, opening, and closing can be defined in the same way. Because the component images are filtered separately with the component-wise filter, there is a possibility of altering the spectral composition of the image, e.g., the color balance and object boundary.

A different way to examing the color morphology is to treat the color at each pixel as a vector. Furthermore, vector-based color morphology makes use of the multivariate ranking concept. First, reduced ordering is performed. Each multi-

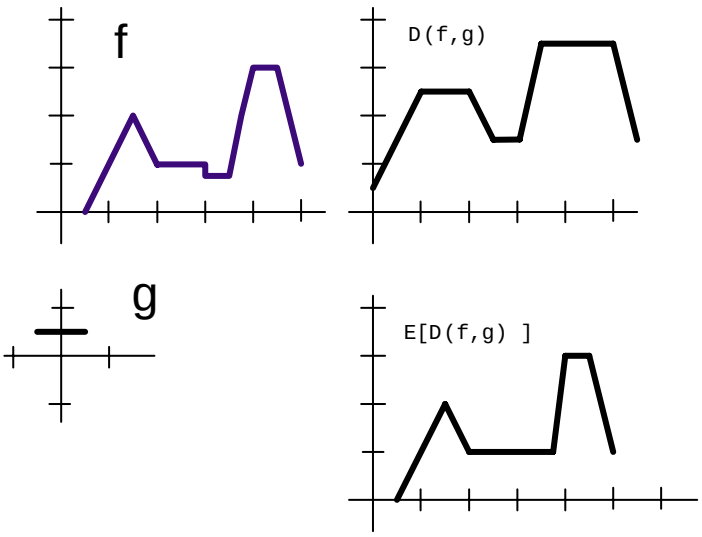


Figure 2.10: Grayscale Closing

variate sample is mapped to a scalar value based on reduced ordering function and then the samples are ordered according to the mapped scalar value. Given the reduced ordering function d and the set H, the value of vector color dilation of f by H at the point (x,y) is defined as :

$$(f \oplus_v h)(x,y) = a \ (\ \oplus_v \ \text{denotes vector dilation}\ ) \tag{2.18}$$

where

$$a \in \{f(r,s) : (r,s) \in H\}$$

and

$$d(a) \geq d(f(r,s)) \quad \forall (r,s) \in H_{(x,y)}$$

Similarly, the vector erosion of f by H at the point (x,y) is defined as :

$$(f \ominus_v h)(x,y) = a \ (\ \ominus_v \ \text{denotes vector dilation}\ ) \tag{2.19}$$

where

$$a \in \{f(r,s) : (r,s) \in H\}$$

and

$$d(a) \leq d(f(r,s)) \quad \forall (r,s) \in H_{(x,y)}$$

In [29] and [30], some experiments of these two approaches are examined. Each approach has its pros and cons. In this thesis, the idea of component-wise approach is used for multiscale edge detection.

Chapter 3

Watershed Transformation and Morphological Gradient Operators

In chapter 2, the mathematical definitions and the geometric characteristics of the primitive morphological operators have been reviewed. Now is the time to introduce the morphological filters which can be constructed based on the underlying morphological operations. Chapter 3 is divided into two parts: the first part deals with the "watershed transformation" and the second one is devoted to the "multiscale gradient operator".

3.1 Watershed Transformation

Unlike the typical morphological filters, watershed transformation is not composed of the primitive morphological operations. In fact, the first algorithm for computing watersheds are found in the field of topography. The introduction of the watershed transformation as a morphological tool is due to H. Digabel and C. Lantuéjoul [32]. Later, a joint work of C. Lantuéjoul and S. Beucher led to the "inversion" of this original algorithm in order to extend it to the more general framework of grayscale images [33][34]. Watersheds were then approached theoretically by F. Maisonneuve and used in numerous grayscale segmentation problems [35]. When combined with other morphological tools, the watershed transformation is at the basis of extremely powerful segmentation procedures [13].

3.1.1 Definition of Watersheds

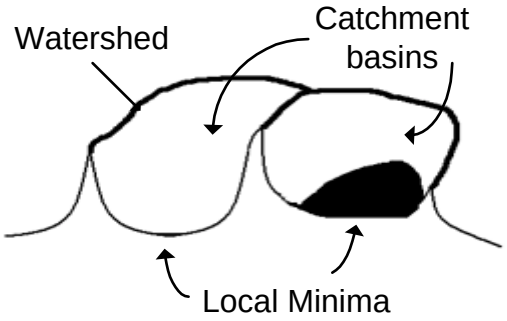


Figure 3.1: Watersheds, catchment basins and minima

In terms of topography, the watershed can be imagined as the high mountain that separates two regions. Each region has its own minimum and, if a drop of water falls on one side of the watershed, it will reach the minimum of the region. The regions that the watershed separates are called catchment basins. The statements are shown in Fig. 3.1. Although the definition stated above is quite intuitive, it is not well suited to practical implementation.

The algorithmic definition of [13] is based on the concept of *"immersion"*. We can figure that we have pierced holes in each regional minima of a grayscale image I which can be thought as a surface. Then we slowly immerse out surface into a lake. Starting from the minima of lowest altitude, the water will progressively fill up the different catchment basins of I. At each pixel where the water coming from two different minima would merge, in this way, we build a "dam". At the end of this immersion procedure, each minimum is totally surrounded by dams corresponding to watersheds of I.

Before expressing the catchment basins and watersheds by immersion more formally, we now need to recall the definitions of the *geodesic distance* and the *geodesic influence zones*, which can deal with the "plateau" for the watershed transformation.

Definition 3.1 (Geodesic Distance) *The geodesic distance $d_A(x, y)$ between two pixels x and y in A is the infimum length of the paths which join x and y and are*

totally included in A.

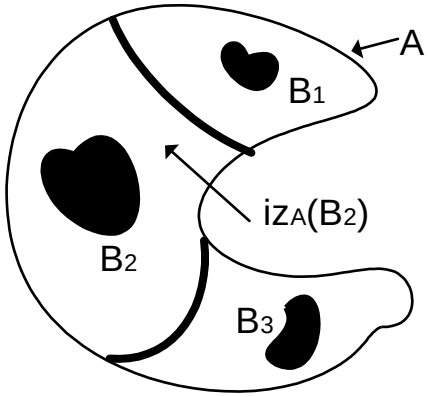


Figure 3.2: Geodesic influence zones of connected component Bi inside set A

Definition 3.2 (Geodesic Influence Zone) *Suppose A contains a set B consists of several connected components B_1, B_2, ... B_k. The geodesic influence $iz_A(B_i)$ of a connected component B_i of B in A is the locus of the points of A whose geodesic distance to B_i is smaller than their geodesic distance to any other component of B (as shown in Fig. 3.2). That is,*

$$iz_A(B_i) = \{p \in A, \forall j \in [1,k]/\{i\}, d_A(p,B_i) < d_A(p,B_j)\}$$

Now we can define immersion procedure and its associated catchment basins and watersheds in terms of the geodesic influence zone.

Definition 3.3 (Catchment Basins and Watersheds by Immersion) *The set of the catchment basins of the grayscale image I is equal to the set $X_{h_{max}}$ obtained after the following recursion:*

1. $X_{h_{min}} = T_{h_{min}}(I)$, $T_h(I) = \{\ p \in D_I\ ,\ I(p) \leq h\ \}$
2. $\forall h \in [h_{min}, h_{max} - 1]$, $X_{h+1} = min_{h+1} \bigcup IZ_{T_{h+1}(I)}(X_h)$

The immersion procedure is done in the recursion and watersheds of I correspond to the set of the points of D_I which does not belong to any catchment basin.

In [13], an immersion-based algorithm for watersheds extraction is proposed. It consists of two steps, sorting and flooding, and the queue structure is used to facilitate the speed. Actually, it claims that this fast algorithm runs in linear time and yields a good result. More details can be found in [13].

To use watersheds in image segmentation, we need to determined the gradient image first. This can be done by the Laplacian operator, the Sobel operator, or morphological gradient operators. Intuitively, the contours of the objects correspond to the ridge of the gradient image; therefore, the contours can be extracted as watersheds of the gradient.

Figure 3.3: Watersheds on Akiyo

3.1.2 Region Merging

As what you can see in Fig. 3.3, there are lots of small regions, even in the "visually" homogeneous regions. The correct contours are lost in a mass of irrelevant ones. It is the well-known weakness of the watershed transformation called *over-segmentation*. Over-segmentation mainly comes from the noise or quantization error of the original image which causes the local minima in the gradient image. Some researches pay their attention on modifying the gradient image to eliminate the local minima [36]. Some put emphases on the post-processing of watershed transformation. The most natural methodology is to merge the small regions in a homogeneous region since they may posses certain homogeneous characteristic in

Figure 3.4: Region Merging of Watersheds on Akiyo

intensity, color or texture. For example, merge the adjacent regions in Fig. 3.3 if their color distance is smaller than 15 will yield the result in Fig. 3.4.

As you can see, some homogeneous regions have been recovered in the region merging process. However, region merging may consume lots of time without carefully programming. Hence, some researches try to develop fast algorithms for region merging [15]. Moreover, some "visually heterogeneous regions" may be "statistically homogeneous regions". To overcome this problem, careful intervention from the user or explicit prior knowledge on the image structure is necessary.

3.2 Multiscale Gradient Operators

Image edge detection is a basic tool in image segmentation since edges carry valuable features of the objects in the image. Edges in an image are formed due to variations of illumination in the scene. Hence, the conventional approach to edge detection is composed of "gradient calculation" and "thresholding". First, the original image is transformed to a gradient image which represents the edge strength of each pixel. A threshold is then applied to classify each pixel to the edge point or non-edge point. Traditionally, the gradient image can be obtained by means of first-order differential operators or Laplacian operators which can enhance the spatial intensity changes in the image. Morphological edge detectors have also

been proposed for their robustness under noisy condition and some of them are discussed in the following section.

3.2.1 Multiscale Edge Detectors

Morphological gradients is based on the concept of *residues* which make difference between the transformations or filters. Some simple morphological gradient operators are illustrated as follows ($\oplus$ denotes dilation, $\ominus$ denotes erosion and B represents the structure element) [28]:

Gradient by Erosion $G_e(f) = f - (f \ominus B)$.

Gradient by Dilation $G_d(f) = (f \oplus B) - f$.

Morphological Gradient $G(f) = (f \oplus B) - (f \ominus B)$.

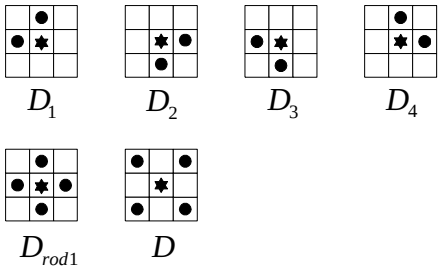


Figure 3.5: Domains of the structure elements

If B is chosen as the *rod* structure element with *flat top* (i.e its value is zero) whose domain is the 4-neighbor of the origin including it (see Fig. 3.5), then gradient by erosion and dilation are also called *erosion residue edge detector* and *dilation residue edge detector* respectively. Furthermore, they can be reduced to the following mathematical expressions [26]:

$$
\begin{aligned}
& G_e(r,c) \\
&\quad = \quad f(r,c) - (f \ominus b)(r,c) \\
&\quad = \quad f(r,c) - [min_{(i,j)\in D_{rod}}(f(r-i,c-j))]
\end{aligned}
\tag{3.1}
$$

$G_d(r,c)$

$$
\begin{aligned}
&= (f \oplus b)(r,c) - f(r,c) \\
&= [max_{(i,j)\in D_{rod}}(f(r-i,c-j))] - f(r,c) \qquad (3.2)
\end{aligned}
$$

However, the above three morphological residue gradients are not good enough due to sensitivity to noise. The first robust morphological edge detector is proposed by Lee, Haralick and Shapiro called *blur-minimization edge detector* [37].

$$f_o(z) = min\{f_{av}(z) - (f_{av} \ominus Bv)(z), (f_{av} \oplus Bv)(z) - f_{av}(z)\} \qquad (3.3)$$

where $f_{av}(z)$ is the input image blurred with a 2D running mean filter, $f_o(z)$ is the output image, and B is the structure element which is a square with (2n+1)*(2n+1) pixels. The *ATM edge detector* replaces the mean filter of Eqn. (3.3) with the α-trimmed mean filter. In [37], an *ASF edge detector* is introduced and its performance is better than the above two edge detectors.

To become more robust to noise, a *multiscale* gradient algorithm can be taken into consideration. The term *multiscale* means to analyze the image with structure elements with different sizes. Intuitively, using the structure element of smaller size may detect fine edges but become more sensitive to noise. On the other hand, the larger the size of the structure element, more noise can be removed but the thicker the edge becomes. Hence, combining the morphological gradients in different scales is able to not only become insensitive to noise but also extract the various fineness of the edges.

In [26], a multiscale morphological edge detector is introduced. First, the *improved erosion residue operator* and *improved dilation residue operator* are defined as

$$
\begin{aligned}
G'_e(r,c) = min\{&f(r,c) - erosion_{D_{rod1}}(r,c), \\
&f(r,c) - erosion_D(r,c), G''_e(r,c)\}, \qquad (3.4)
\end{aligned}
$$

where $G''_e(r,c)$ is defined as

$$
\begin{aligned}
G''_e(r,c) = max\{&|erosion_{D_1}(r,c) - erosion_{D_2}(r,c)|, \\
&|erosion_{D_3}(r,c) - erosion_{D_4}(r,c)|\} \qquad (3.5)
\end{aligned}
$$

and

$$
\begin{aligned}
G'_d(r,c) = min\{&dilation_{D_{rod1}}(r,c) - f(r,c), \\
&dilation_D(r,c) - f(r,c), G''_d(r,c)\}, \qquad (3.6)
\end{aligned}
$$

where $G''_d(r,c)$ is defined as

$$G''_d(r,c) = max\{|dilation_{D_1}(r,c) - dilation_{D_2}(r,c)|, \\ |dilation_{D_3}(r,c) - dilation_{D_4}(r,c)|\} \tag{3.7}$$

respectively. Recall that D_i is the structure element with flat top and its domain is defined in Fig. 3.5.

In Fig. 3.5, each structure element is defined with radius 1. To extend the above operators to scale n, the structure elements are modified in proportion to n. That is, nD_1, nD_2, nD_3, nD_4 have the same domain as D_1, D_2, D_3, D_4 but have size n. Similarly, D_{rodn} is defined as nD_{rodn} and $nD = nN_8/nN_4 \cup (0,0)$ (where N_8 means 8-neighbor and N_4 means 4-neighbor). Therefore, the improved dilation residue operator at scale n is defined as

$$G^n_d(r,c) = min\{dilation_{D_{rodn}}(r,c) - f(r,c), \\ dilation_{nD}(r,c) - f(r,c), G^{n'}_d(r,c)\}, \tag{3.8}$$

where $G''_d(r,c)$ is defined as

$$G^{n'}_d(r,c) = max\{|dilation_{nD_1}(r,c) - dilation_{nD_2}(r,c)|, \\ |dilation_{nD_3}(r,c) - dilation_{nD_4}(r,c)|\} \tag{3.9}$$

The improved erosion residue operator at scale n can be defined in the similar way. In [26], it focuses on the improved dilation residue operator and mentions that we can also define strength by the improved erosion residue operator or by sum of them.

After defining the edge image of each scale, the task turns to combining the edge strength. It is very intuitive to use the weighted pixel summation, i.e.

$$f'(r,c) = \sum_k^l w_n f'_n(r,c) \tag{3.10}$$

where $[k,l]$ represents the range of scale and w_i's are respective weights of each scale. Other kinds of combinations are also possible.

Since the edge strength of pixel has been figured out, we can directly threshold each pixel to classify it into edge or non-edge. In [26], it makes use of *"Non-maximal suppression"*. Note that the combined edge image appears to contain long range of mountains and the true edge lie along its ridge (see the right part of Fig. 3.6); hence, the watershed transformation introduced in the previous section is suitable for extracting the ridge points in an edge image.

3.2.2 The Multiscale Gradient Operator for Watershed Transformation

In the previous section, the general concept of the multiscale gradient operators has been reviewed, and a useful multiscale gradient algorithm in [26] has also been introduced. In this section, a multiscale gradient operator in [36] is described and we will focus on how to build up the edge image which is suitable for watershed transformation.

In watershed transformation, each local minimum of the gradient image will be treated as a catchment basin. Hence, conventional gradient algorithms, such as Gaussion filter, Laplacian operators and some morphological gradient operators, produce too many local minima in the homogeneous regions due to the noise or quantization error. This leads to partition the homogeneous region into a large number of *"spots"*. It is what we called *"over-segmentation"*. The most straightforward methodology is thresholding the gradient directly. However, thresholding cannot preserve the blurred edges whose gradient are low even though the intensity change between two sides may be high.

3.2.2.1 A Multiscale Gradient Algorithm

To overcome the problem stated above, a multiscale gradient algorithm based on morphological operators followed by an algorithm of the local minima elimination has been proposed in [36].

First, the gradient image is produced by a multiscale gradient algorithm.

$$MG(f) = \frac{1}{n}\sum_{i=1}^{n}[((f \oplus B_i) - (f \ominus B_i)) \ominus B_{i-1}] \tag{3.11}$$

where n is the scale, and B_i denotes the group of square structure elements whose size is $(2i+1) \times (2i+1)$ pixels. The choice of square structure element has the advantage of low computational cost. Moreover, other structure elements, like line segments of different directions, can be used for specific application.

3.2.2.2 Local Minima Elimination

Local minima consists of a small number of pixels or has a low contrast with respect to their neighbors. It is usually caused by noise or quantization error. The

procedure to eliminate local minima used in [36] takes advantage of the technique "morphological grayscale reconstruction" proposed in [38].

Morphological reconstruction transformation is well-known in the binary case, where it simply extracts the connected components of an image which are "marked" by another image. While extending to grayscale reconstruction, it could accomplish several tasks such as image filtering, extrema, domes, and basins extraction. In following paragraphs, we will review morphological reconstruction first and then describe how to apply it to modify the gradient.

Definition 3.4 (Binary Reconstruction) *Let $X, Y \subset Z^2$ and $Y \subseteq X$. The reconstruction of X from Y is obtained by iterating elementary geodesic dilations of Y inside X until stability is reached; that is,*

$$\rho_X(Y) = \bigcup_{n \geq 1} \delta_X^{(n)}(Y) \tag{3.12}$$

where $\delta_X^{(n)}(Y)$ can be obtained by iterating n elementary geodesic dilation and the geodesic dilation is defined as :

$$\delta_X^{(1)}(Y) = (Y \oplus B) \bigcap X \text{ (} B \text{ is the structure element of size 1)} \tag{3.13}$$

.

Definition 3.5 (Dilation-based Grayscale Reconstruction) *Let I, J be two grayscale images defined on the same domain and $J \leq I$. The reconstruction of I from J, denoted as $\gamma_I^{(rec)}(J)$, is obtained by iterating elementary geodesic dilations of J under I until stability is reached:*

$$\gamma_I^{(rec)}(J) = \bigvee_{n \geq 1} \delta_I^{(n)}(J) \tag{3.14}$$

where $\delta_I^{(n)}(J)$ can be obtained by iterating n elementary geodesic dilation and the geodesic dilation is defined as :

$$\delta_I^{(1)}(J) = (J \oplus B) \bigwedge I \tag{3.15}$$

(B is the flat structure element of size 1 and $\bigwedge$ stands for pointwise minimum)

Definition 3.6 (Erosion-based Grayscale Reconstruction) *Let I, J be two grayscale images defined on the same domain and $I \leq J$. The reconstruction of I*

from J, denoted as $\varphi_I^{(rec)}(J)$*, is obtained by iterating elementary geodesic erosions of J above I until stability is reached:*

$$\varphi_I^{(rec)}(J) = \bigwedge_{n \geq 1} \varepsilon_I^{(n)}(J) \tag{3.16}$$

where $\varepsilon_I^{(n)}(J)$ *can be obtained by iterating n elementary geodesic dilation and the geodesic dilation is defined as :*

$$\varepsilon_I^{(1)}(J) = (J \ominus B) \bigvee I \tag{3.17}$$

(B is the flat structure element of size 1 and $\bigvee$ *stands for pointwise maximum).*

Now is the time to describe how the morphology reconstruction helps to eliminate the local minima. Recall the discussion in section (2.2.1). Generally, grayscale dilation has the effect of eliminating small peaks in a signal, and it can be employed to smooth the gradient image. Given the gradient image by applying the multi-scale gradient algorithm stated in the last section, the procedure of local minima elimination in [36] can be stated as follows:

1. The gradient image $MG(f)$ is dilated with a square structure element B_s of 2×2 pixels, i.e. $MG(f) \oplus B_s$.

2. A constant h is added to the dilated gradient image and perform erosion-based reconstruction. That is, the final gradient image can be expressed as $\varphi_{(MG(f))}^{(rec)}((MG(f) \oplus B_s) + h)$.

In step 1, if the local minimum is "narrower" than the size of B_s, it will be filled by the nature of dilation. In step 2, the local minima with a contrast lower than h can be filled irrespective to their absolute value. If the local minimum is wide and deep, it still cannot be removed. However, comparing to thresholding the gradient image directly which removes only the minima with low absolute value, it is more reasonable to use the morphological approach shown above to eliminate the local minima. Fig. 3.8 shows the effect of local minima elimination using grayscale reconstruction.

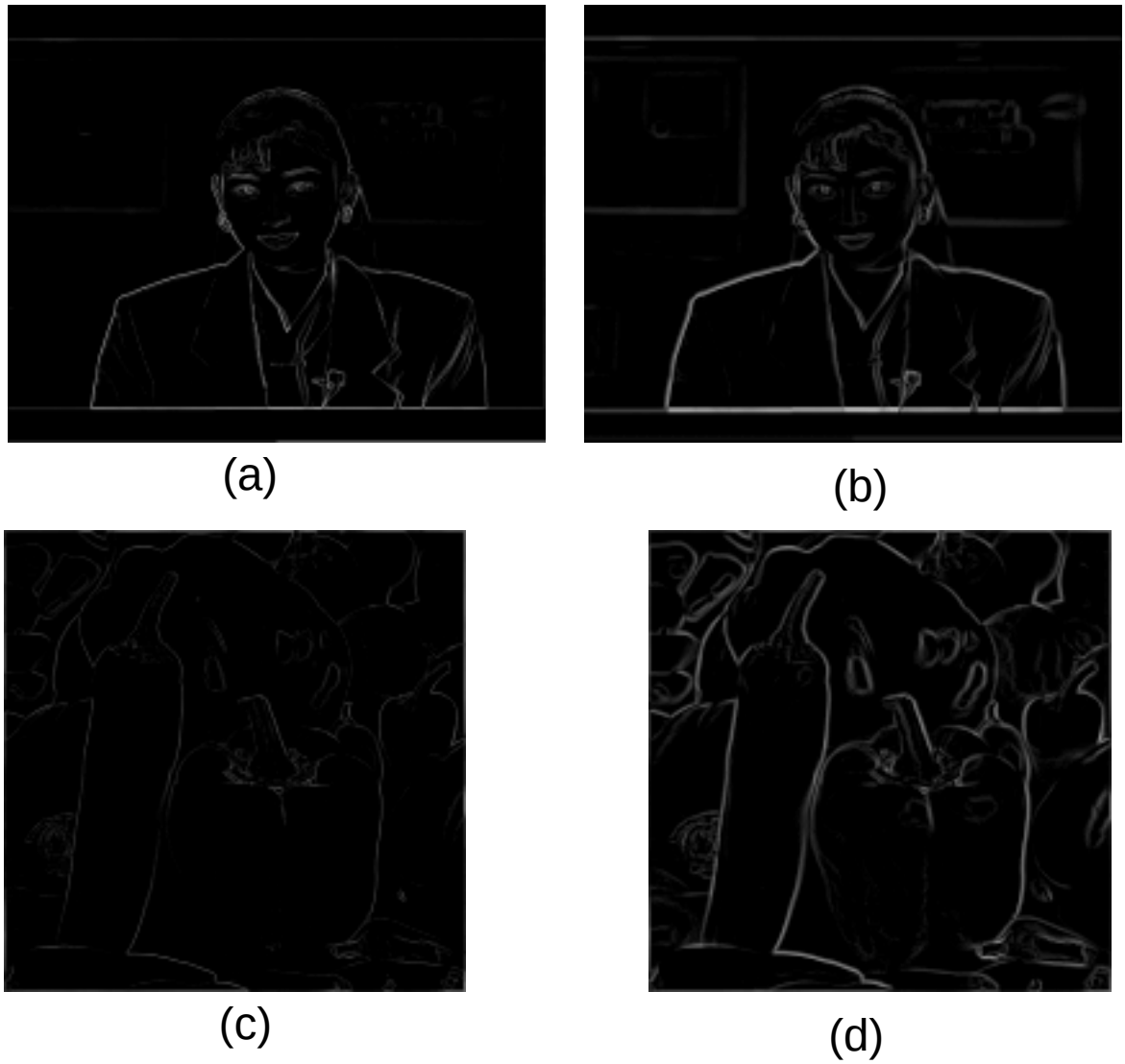


Figure 3.6: Akiyo and Peppers : Multiscale Edge Detector at scale 1(left part) and scale 5(right part)

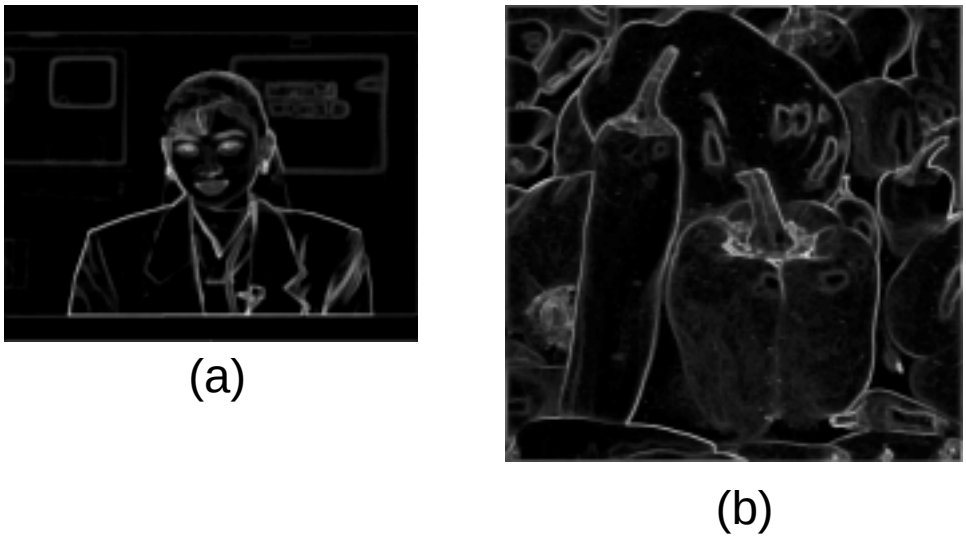


Figure 3.7: Akiyo and Peppers : Multiscale Edge Detector (n = 2)

Without Reconstruction

Reconstruction with 16 iterations

Figure 3.8: Akiyo images with / without reconstruction

Chapter 4

The Supervised Segmentation System Based on Mathematical Morphology

4.1 The Proposed System

Figure 4.1 shows the flowchart of the proposed system. The aim of the "object definition" stage is to obtain the rough definition of the semantic object through user interaction. At the "region analysis" stage, the information of the features like edge strength in the image is extracted. Based on the information from "object definition" and "region analysis" modules, the "adaptive morphological operators" stage is devoted to properly create the *band* where the actual object boundary resides. Finally, the object boundary will be located at the "classification" stage. In following sections, the functionalities of each stage are explained respectively.

4.2 Object Definition

During the past decades, many research works of image or video segmentation have concentrated themselves on unsupervised segmentation. However, the definition of "semantic" is ambiguous and only a human knows the real meaning of "semantic". That is why it is very difficult for the computer to accomplish automatic segmentation of semantic meaningful objects if lacking high level image understanding. On the other hand, a computer can perform a low level segmentation of an image

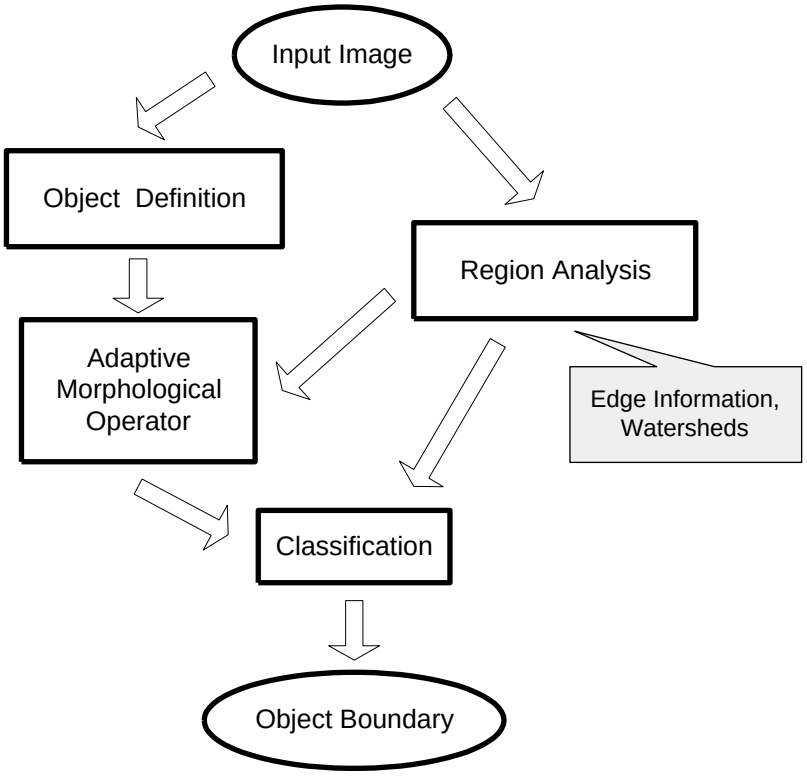


Figure 4.1: Flowchart of the proposed system

into homogeneous regions based on some image features. Hence, the adoption of some form of user interaction in the segmentation process becomes commonly accepted and some research works of semi-automatic object segmentation have been proposed [11] [12] [17].

In this "object definition" stage, user assistance is involved to provide an approximation of the object boundary as the definition of the semantic object. In view of how a user interacts with the computer, two types of methods, pixel-based and contour-based approaches, are commonly used. In a pixel-based method, a user needs to input the positions of the boundary. On the other hand, only the key nodes to draw the outline of the boundary are required for contours-based method. Generally speaking, the contour-based approach is more intuitive and requires less human effort than the pixel-based one.

In this system, a contour-based polygonal method is designed. An user clicks the mouse to provide the key nodes along the object. Then the key nodes are connected to form the polygonal approximation of the object boundary. Although

the polygonal approach cannot approximate the complicated boundary well, it provides some information to define the semantic object. Figure 4.2 shows an example of the resultant object definition obtained in this stage.

Figure 4.2: Object Definition

4.3 Region Analysis

Image segmentation is a key issue in the area of computer vision for a long time and a number of techniques and algorithms have been proposed. Although the semantic object cannot be extracted due to lack of image understanding, the image could be partitioned into some non-intersecting regions such that each region is homogeneous on the given quantitative criteria, such as grayscale, color, texture. In the proposed system, the well-known *watershed transformation* is used to perform the region analysis. The reasons of choosing the watershed transformation include:

1. The watershed transformation can run in linear time using the algorithm proposed in [13].

2. The edge strength image is designed as an parameter of the adaptive morphological operators and it is also the only information required for the watershed transformation. Hence, no more calculation is needed for preparing the watershed transformation.

3. It is known that the watershed transformation usually has the problem of

over-segmentation. From another point of view, the spot-like region can be used as the unit for the user to refine the segmentation result.

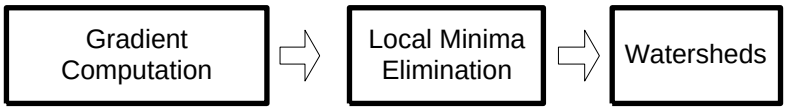


Figure 4.3: Flow diagram of typical watershed-based image segmentation

Figure 4.3 shows the procedures for applying watersheds to region ananlysis. The general concept of each module has been reviewed in chapter 3. In the following paragraph, the implementation detail of each component will be introduced.

4.3.1 Gradient Computation

Recall the multiscale gradient algorithms introduced in chapter 3. Both of them are more robust to noise and suitable for watershed transformation than traditional gradient operators like Sobel or Laplacian operators. In addition, they are defined in the grayscale domain. Traditionally, to make use of such operators, a color image must transform to a grayscale image first. In this thesis, the idea of component-wise color morphology introduced in chapter 1 is involved.

1. Suppose the color image is defined in the RGB domain. Let I_R, I_G, I_B denote the Red, Green, Blue component of the original image, respectively.

2. Supposed the multiscale gradient algorithm in [36] is used. Apply the multiscale edge detector to each component image.

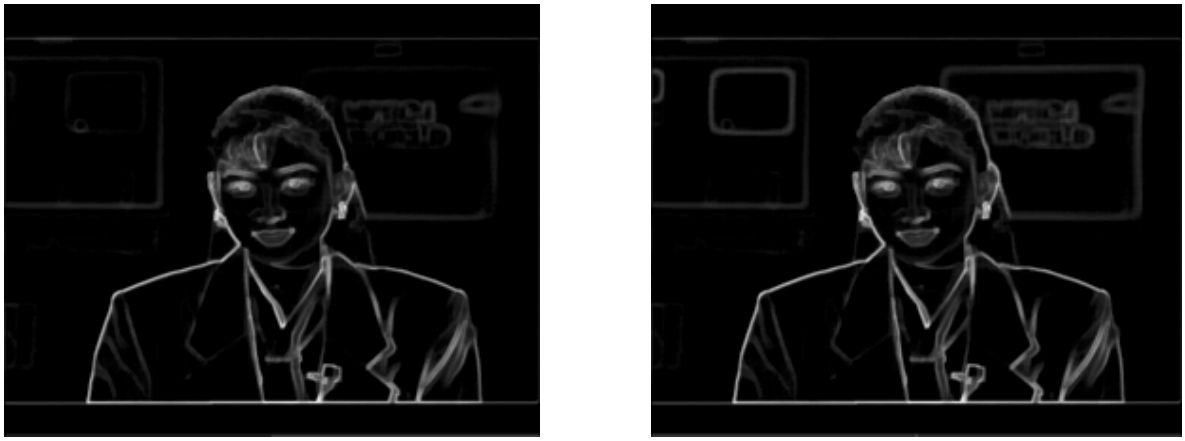
$$
\begin{aligned}
MG_R &= \frac{1}{n}\sum_{i=1}^{n}[((I_R \oplus B_i) - (I_R \ominus B_i)) \ominus B_{i-1}] \\
MG_G &= \frac{1}{n}\sum_{i=1}^{n}[((I_G \oplus B_i) - (I_G \ominus B_i)) \ominus B_{i-1}] \\
MG_B &= \frac{1}{n}\sum_{i=1}^{n}[((I_B \oplus B_i) - (I_B \ominus B_i)) \ominus B_{i-1}] \qquad (4.1)
\end{aligned}
$$

where n is the scale and B_i denotes the group of square structure elements whose size is $(2i+1) \times (2i+1)$ pixels.

3. Given the edge strength image of each component, use weighted pixel-wise summation to combine them.

$$MG(r,c) = \sum_i w_i \times MG_i(r,c) \tag{4.2}$$

where i = { R, G, B }.



(a) (b)

Figure 4.4: Grayscale Edge Detector and Component-wise Gradient Operator

Figure 4.4 illustrates the results of the component-wise gradient operator and grayscale edge detector. The two parameters n and w_i are chosen as 3 and equal weight, i.e. $\frac{1}{3}$. The most significant difference can be found in the boundaries of two blue screens in the right and left of Akiyo. In the grayscale case, the edge strength is not as high as that in component-wise case because the gray values of the blue screen and their adjacent background are almost the same. On the other hand, the edge strength of blue component will enhance the boundary of the blue screen.

Furthermore, there are three parameters that can be set by the user. First, the color space can be transformed to another domain, like YCbCr or LUV. Second, the scale of multiscale algorithm controls the ability of edge detection and insensitivity to noise. Briefly speaking, the thickness of edge will increase in proportion to the number of scale, as illustrated in Fig.3.6. The parameter w_i can be designed for

differentiating the weight of each color component. The selection of color space and w_i needs the background knowledge of color science which is beyond the scope of this thesis.

4.3.2 Local Minima Elimination

The most obvious weakness of watershed transformation is "over-segmentation" due to a lot of local minima. At the same time, as the number of regions after watersheds increases, the complexity and memory consumption will also increase in the post-processing, like region merging. The goal of local minima elimination is to improve the segmentation result of watersheds and facilitate the post-processing.

The algorithm of eliminating local minima proposed in [36] has been reviewed in chapter 3 and is adopted in this stage.

Let $MG(f)$ denote the gradient image computed from the multiscale gradient algorithm. The final gradient is modified as:

$$\varphi^{(rec)}_{(MG(f))}((MG(f) \oplus B_s) + h) \tag{4.3}$$

where

$$\varphi^{(rec)}_I(J) = \bigwedge_{n \geq 1} \varepsilon^{(n)}_I(J) \tag{4.4}$$

Theoretically, Eqn 4.4 is iterated until stability is reached. Nevertheless, in practical implementation, the user can modify n as long as the corresponding result is good enough.

4.4 Adaptive Morphological Operators

Once obtaining the object definition by user assistance, the approximation of the real object boundary is not too far away. Hence, the tasks turns to creating the band of interest where the real object boundary resides. It is exactly the goal of "adaptive morphological operators". The concept is shown in Fig. 4.5.

In Fig. (4.5), B indicates the real object boundary, B_{init} is the initial boundary from the "object definition" stage. If we could find out the interior boundary B_{in} lying inside the object and the exterior boundary B_{out} lying outside the object, then we can limit the searching area where the real object boundary lies. In [17],

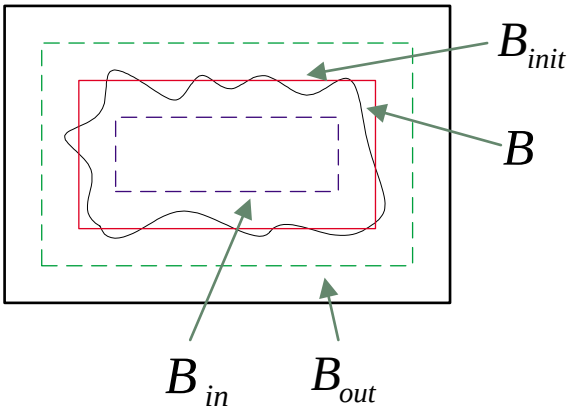


Figure 4.5: Interior and Exterior Boundary

the binary morphological operators introduced in chapter two are chosen to obtain B_{in} and B_{out}. While thinking each pixel inside the B_{init} as foreground and others as background, a binary image $I_{B_{init}}$ is created. Then binary dilation and binary erosion are performed to obtained two images $I_{B_{in}}$ and $I_{B_{out}}$:

$$I_{B_{in}} = (I_{B_{init}} \ominus SE) \tag{4.5}$$

$$I_{B_{out}} = (I_{B_{init}} \oplus SE) \tag{4.6}$$

where SE is the structure element.

By properly choosing the structure element SE, we can guarantee that

$$I_{B_{in}} \subseteq I_{B_{init}} \subseteq I_{B_{out}}$$

and, furthermore,

$$B_{in} \subseteq B \subseteq B_{out}. \tag{4.7}$$

The searching area then equals to $I_{B_{out}} - I_{B_{in}}$.

Note that the criterion in Eqn. (4.7) is quite strict and almost decides the accuracy of the segmentation result since wrong searching area will not contain right boundary. Moreover, the interior and exterior boundaries are obtained from the binary morphological operators; hence, the issue turns to how to choose the structure element properly. In this methodology, the size and shape of the structure element controls the tolerance of the displacement between the initial boundary and real object boundary. In [17], the structure element is chosen as a square

structure element of size 2. The ability of fault tolerance becomes relatively weak and careful user assistance is necessary. To tolerate the error between the initial boundary and real object boundary, an *adaptive morphological operator* is proposed to accurately create the band where the object boundary resides.

4.4.1 General Concept of Adaptive Morphological Operators

	vertically invariant	vertically variant
horizontally invariant	shift-invariant operator	H-operator
horizontally variant	V-operator	general operator

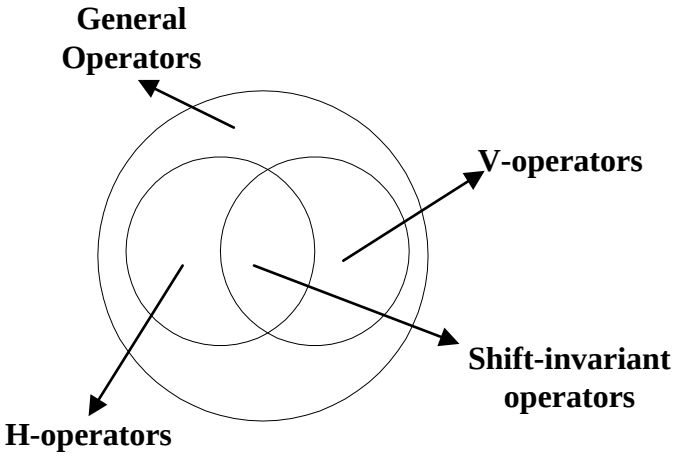


Figure 4.6: Four types of morphological operators

Most morphological operators for image or signal processing in the past belong to shift-invariant operators. In [39], V-operators are investigated because the geometric interpretations of V-operators can be viewed as time-varying or space-varying versions generalized from those of shift-invariant ones. In fact, if an effective structure element assignment procedure is used, then a V-operator can adaptively change its processing scale according to the local characteristics of a signal [39].

4.4.2 Adaptive Dilation and Erosion

The V-operator using progressive umbra-filling (PDF) procedure for adaptive assignment of structure element has been proposed in [39] for signal filtering. The reason why the adaptive morphological operators are employed in this thesis is to accomplish the aim stated in Eqn. (4.7); that is,

$$B_{in} \subseteq B \subseteq B_{out}. \tag{4.8}$$

Therefore, the procedure of adaptive assignment of structure element should be designed such that the domain of structure element whose center coincides with each pixel belonging to the initial boundary includes the real object boundary. That is,

$$\forall (x, y) \in B_{init} \quad SE_{(x,y)} \cap B \neq \emptyset \tag{4.9}$$

where $SE_{(x,y)}$ denotes to translate the structure element SE with vector (x,y).

The following paragraphs describe the procedure for adaptive assignment of structure elements.

4.4.2.1 Maximizing Edge Strength

Intuitively, if the domain of the structure element includes the object boundary, the average edge strength will be relatively higher than that in case the domain of the structure element does not include the object boundary. The *Maximizing Edge Strength* procedure is designed according to this concept.

1. Compute the gradient image MG(x,y) representing the edge strength of each pixel by means of the multiscale gradient algorithm introduced in the last section.

2. Suppose the structure elements SE^k are circles with radius k ranging from 1 to r. Given a point (x,y) lying on B_{init}, calculate the average edge strength under each structure element.

$$\text{For k = 1 to r, } ES_{SE^k}^{(x,y)} = \frac{1}{\#SE^k} \sum_{(i,j) \in SE_{(x,y)}} MG(i,j) \tag{4.10}$$

where $\#SE^k$ denotes the number of points in the domain of SE^k.

3. Let $B_{init} = \{B^1_{init}, B^2_{init}, ..., B^l_{init}\}$. The structure element of B^p_{init}, $p \in [1, l]$ is chosen as the one that maximizes the average edge strength.

$$SE^k \text{ of } B^p_{init} = max_{k=[1,r]} ES^{B^p_{init}}_{SE^k} \tag{4.11}$$

Figure 4.7: Adaptive Dilation and Erosion

After the "maximizing edge strength" procedure, the interior and exterior boundaries are found by applying binary erosion and dilation, respectively. The simulation result with r=15 is illustrated in Fig.4.7.

4.5 Classification

After determining the searching area in which the real object boundary resides, we need to classify each pixel in the area as "object"(in) and "non-object"(out). Typically, the cluster centers need to be decided first, and then assign each undetermined unit to its cluster center that maximizes the similarity measurement.

In this stage, supposing that the cluster centers of "in"(object) and "out"(non-object) have already been located, two classification methods to group the unclassified units are illustrated in Fig. 4.8, pixel-wise classification and morphological watershed.

In pixel-wise classification, each pixel is considered independently so that the spatial relationship among neighboring pixels will not be take into account in the classification process. Hence, the result could be sensitive to noise, and the pixel geometrical relationship could be destroyed.

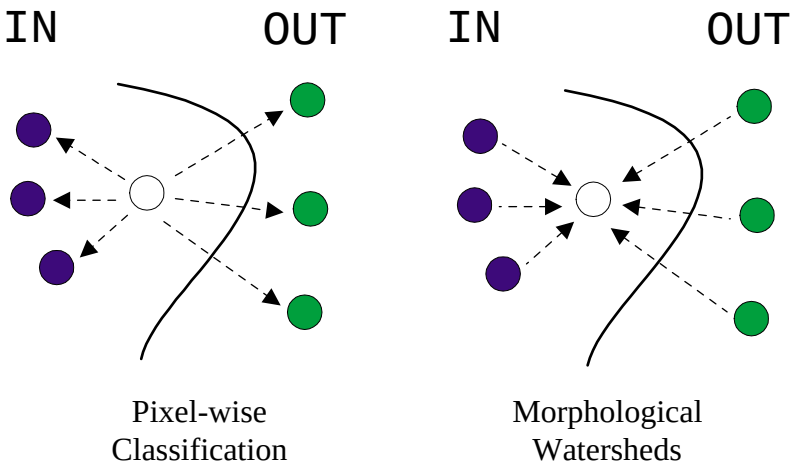


Figure 4.8: Two types of classification

In [17], a *multi-valued watershed* algorithm is proposed. Its main concept is that, if a point is chosen for classification, it is the neighborhood of a marker, and the similarity between them is the highest at that time than any other pair of points and neighborhood markers. In addition, a *hierarchical queue* data structure is used to accomplish this task.

A hierarchical queue is a set of queues with different priorities and each queue is a first-in-first-out data structure. At any time, the element popped from the hierarchical queue is the one from the queue with the highest priority, and, if the queue with higher priority is empty, the element in the first non-empty of lower priority is pulled out.

While using the hierarchical queue for multivalued watershed, two steps are involved: *initialization* and *flooding*. In initialization, the neighborhood units of all "in" and "out" markers are pushed into the hierarchical queue according their similarity with the corresponding markers. In the flooding procedure, while the hierarchical queue is not empty, extract one unit and check if the unit has been classified to any marker. If not yet, the similarity between this unit and its neighboring markers is calculated and this unit is classified to the most similar marker. Then all of the neighborhood of this unit are pushed into the hierarchical queue based on their similarity to the marker. The flooding process continues until the hierarchical queue is empty; that is, all the units in the searching area are already classified.

In the "classification" stage of the proposed system, two kinds of classification methods based on hierarchical queue data structure are introduced. Pixel-based classification mainly comes from the morphological watershed algorithm proposed in [17], except the modification of similarity measurement function and the flooding process. In region-based classification, the basic unit is no longer the image pixel but the regions created from the watershed transformation.

4.5.1 Pixel-based Classification

Marker Extraction Since the interior and exterior boundaries have been found in the "adaptive morphological operators" stage, the "in" and "out" cluster centers can be created by taking all the pixels along B_{in} and B_{out}. Let $B_{in} = \{I_1, I_2, ..., I_m\}$ and $B_{out} = \{O_1, O_2, ..., O_n\}$.

Similarity Measurement Intuitively, the similarity function can be derived from the opposite of the color distance. Generally, the color distance in RGB color space between two points with (R1, G1, B1) and (R2, G2, B2) can be defined as follows:

$$\text{color distance } d = w_R|R1 - R2| + w_G|G1 - G2| + w_B|B1 - B2| \quad (4.12)$$

where w_i denotes the weight for each color component. And then, the similarity function can be defined as opposite of the color distance, i.e. $similarity = MAX\_DISTANCE - colordistance$.

Initialization Given the searching area SA and the hierarchical queue HQ, the initialization procedure can be described as following mathematical expression:
for k = 1 to m,

$$\forall NG(I_k) \in SA,\ \text{HQ.PUSH}\ (NG(I_k), similarity)$$

for j = 1 to n,

$$\forall NG(O_j) \in SA,\ \text{HQ.PUSH}\ (NG(O_j), similarity)$$

The *similarity* is computed between the pixel to be pushed and its neighboring markers by means of the above formula.

The Flooding Procedure The flooding procedure can be expressed in the following C-like pseudo codes:

```
while ( HQ.EMPTY() == false ) {
  pixel = HQ.POP();
  if (  UNCLASSIFY( pixel ) == true ) {
    for each neighboring marker {
      calculate the similarity
    }
    CLASSIFY ( pixel , the marker maximizing the similarity)
    for each NG(pixel) {
      if ( UNCLASSIFY( NG(pixel) ) == true )
        similarity is computed between pixel and NG(pixel)
        HQ.PUSH( NG(pixel), similarity );
    }
  }
}
```

Boundary Decision Since the pixels in the searching area have been classified to either "in" that means it belongs to the object or "out" that means it is in the background, the object boundary is exactly where the "in" pixels meet the "out" pixels.

This flooding algorithm is simplified from the morphological watershed in [17]. In formal classification, the representation of cluster centers will be updated whenever a new unit is classified to it. Furthermore, the similarity is usually computed between the pixel to push and the representation of the neighboring marker, not the neighboring markers directly. The reason why the neighboring marker is chosen instead of the representation of the cluster centers is as follows. The representation of the cluster center is updated once a new pixel join it; however, the pixel is added into the hierarchical queue at one time that it is the neighborhood of a new classified pixel. Suppose the pixel is pushed based on the similarity between it and the representation of the marker. Nevertheless, when it is popped from queue, the representation of the marker that forced the pixel to be pushed may have been changed, and the similarity is no longer the same. In our opinion, it is unreasonable.

On the other hand, the design of similarity function could be modified. One methodology is to explorer another color space that may be more suitable for human visual model. Otherwise, the weight of each component can be modified; however, both need the specific knowledge of color science. Here, the color space is chosen as RGB and the weighting coefficients equal to 1.

4.5.2 Region-based Classification

The idea of region-based classification is to take advantages of the segmentation result of watershed transformation with gradient image. Hence, the regions become the basic unit during the classification process.

Marker Extraction The markers of "in" and "out" are derived from regions that the B_{in} and B_{out} go through respectively. Suppose the "in" markers include m regions, $\{I_1, I_2, ..., I_m\}$ and "out" markers include n ones, $\{O_1, O_2, ..., O_n\}$.

Similarity Measurement Naturally, the representation of each marker can be compute as the multivalued mean of each region. That is,

$$RP_{Region_i}(r, g, b) = \frac{1}{\#Region_i} \sum_{\forall p \in Region_i} p(r, g, b) \tag{4.13}$$

Then both the color distance function and similarity measurement in pixel-based classification can be employed.

Initialization It is almost the same as that in pixel-based methodology, except replacing the neighborhood pixels with adjacent regions.
for k = 1 to m,

$$\forall Adjacent(I_k) \in SA,\ \text{HQ.PUSH}\ (Adjacent(I_k), similarity)$$

for j = 1 to n,

$$\forall Adjacent(O_j) \in SA,\ \text{HQ.PUSH}\ (Adjacent(O_j), similarity)$$

Note that the Adjacent operation is under the constraint of the searching area provided in the "adaptive morphological operators" stage.

The Flooding Procedure The only modification is also the replacement of the neighborhood pixel with adjacent regions. The pseudo codes are skipped.

Boundary Decision After classification, the regions inside the searching area are determined to be "object" or "non-object", and naturally, the object can be obtained by grouping the "object" regions and those already inside the B_{in}.

Till now, there is one problem that is ignored: Does there exist a region that both B_{in} and B_{out} pass through? The answer is "yes" if the "adaptive morphological operators" stage does not take into consideration the region arrangement in the domain of structure element. Fortunately, it should seldom happen because of the "weakness" of watersheds, over-segmentation. Once happened, if the region is large enough to a certain threshold, the region can be further split by the pixel-based classification. If the region is small, it can be classified according to the similarity measurement among its neighboring markers.

4.6 Tracking

There are several approaches to extend image segmentation to video segmentation. One of the most popular methodologies is the combination of spatial segmentation and temporal tracking [17][18][19][40]. Given an image sequence, the I-frame(Intra-frame) is segmented to obtain the object of interest, and the technique of I-frame segmentation can be either supervised or automatic, either spatial information based or temporal information based and, of course, spatio-temporal information based. In the following P-frames(Inter-frames), instead of re-segmenting the image, "tracking" the object based on the temporal information is introduced. Once if the temporal information needs to be compute, the *motion* information must be extracted by a *motion estimator*. Typically, a video object segmentation system consists of the stages, as shown in Fig. 4.9.

In [17], the P-frame tracking consists of four steps:

1. Motion Prediction
2. Motion Estimation
3. Boundary Warping
4. Boundary Adjustment

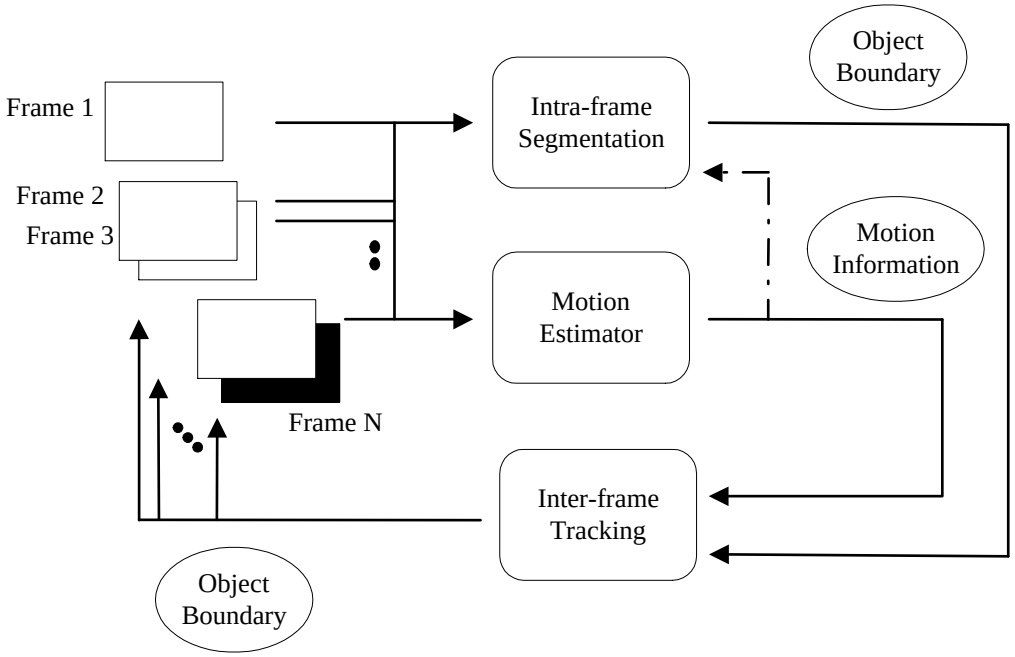


Figure 4.9: Flow diagram of a typical video object segmentation system

In "Boundary Adjustment", its main concept is that, given an approximation of real object boundary from "Boundary Warping", the method of I-frame segmentation can be employed to accomplish the boundary adjustment for non-rigid body boundary refinement.

A simplified tracking scheme (Fig. 4.10) is proposed based on the idea of boundary adjustment and the ability of adaptive morphological operators.

The proposed algorithm is almost the same as the intra-frame segmentation, except the replacement of user interaction with previous segmentation result. Furthermore, the motion estimator is absent in this tracking scheme. The main idea is that, as long as the initial boundary is not too far away from the real object boundary, the adaptive morphological operators are able to properly find out the searching area where the object boundary resides. Hence, under the constraint that "the motion of the object is not too large for adaptive morphological operators to provide the correct searching area", the object boundary will eventually be found in the "classification" stage.

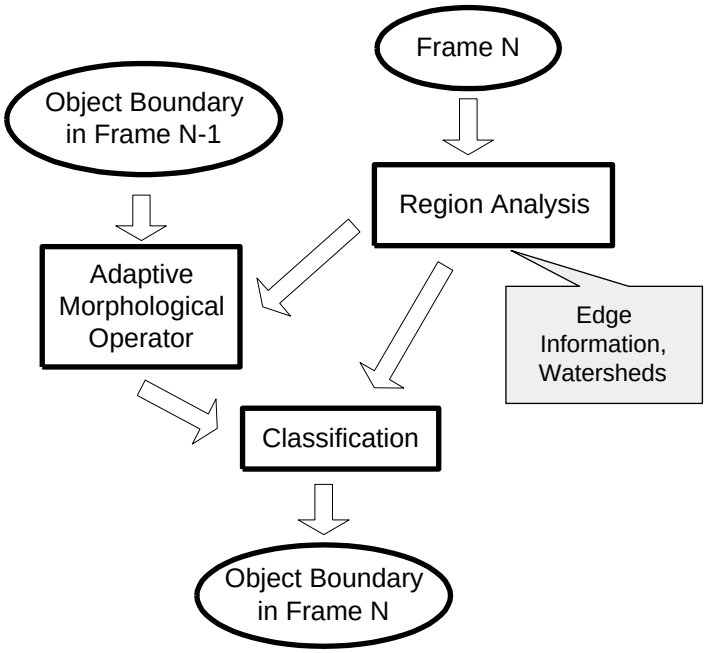


Figure 4.10: Flow diagram of proposed object tracking

Chapter 5

Experimental Results and Discussion

5.1 Experimental Results

Figure 5.1 shows the experimental results of the intermediate steps and the pixel-based segmentation.

(A) The original image of "Akiyo" with 704*576 pixels.

(B) The user clicks eight points around "Akiyo".

(C) The polygonal definition of "Akiyo" are obtained by connecting the successive two points with one line.

(D) The gradient image used for the "maximizing-edge-strength" adaptive morphological operators. The gradient image are computed by means of the multiscale gradient operator introduced in 3.2.2 where the scale number is 3. In the "local minima elimination" step, h equals 3 and the iteration number for "erosion-based reconstruction" is set to 20.

(E) After obtaining the gradient image, the "maximizing-edge-strength" adaptive morphological operators is performed.

(F) The segmentation result of pixel-based classification.

In figure 5.2, the result of region-based classification with different gradient images are shown.

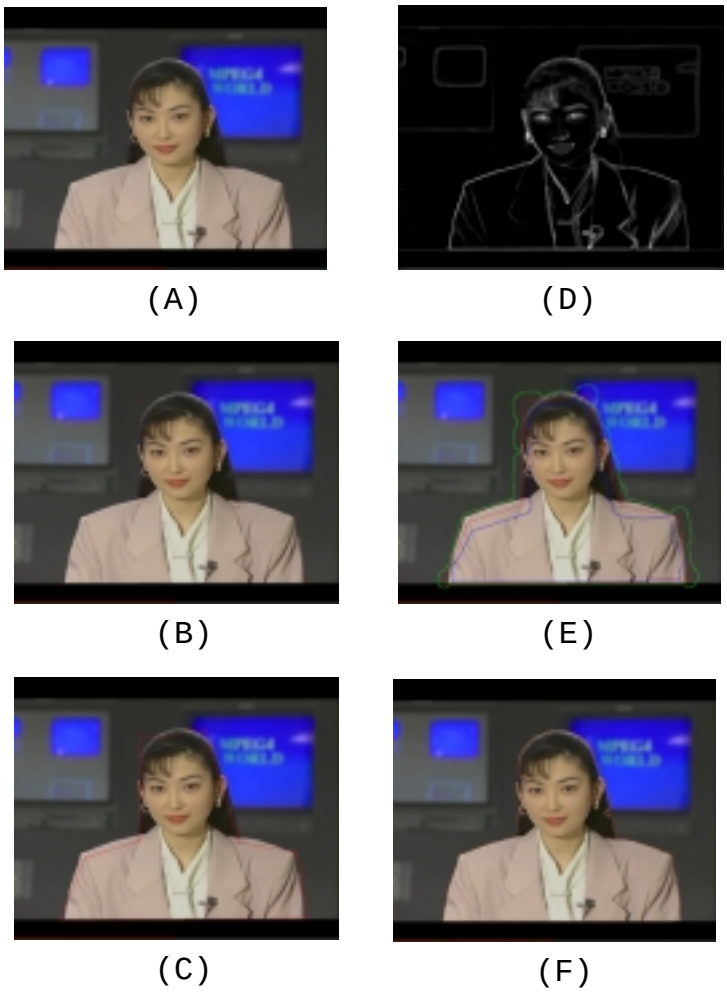


Figure 5.1: Akiyo Demo

(A),(E) The results of watershed transformations with different gradient images. (A) is the same as (D) in figure 5.1 and (E) is computed by the multiscale gradient algorithm without grayscale reconstruction.

(B),(F) The regions which are passed through by the interior boundary.

(C),(G) The regions which are passed through by the exterior boundary.

(D),(H) The result of region-based classification.

5.2 Discussion

1. The erosion-based grayscale reconstruction affects the number of regions generated by watershed transformation. The parameter h used to eliminate the local minima with contrast lower than h dominates the effect of local minima elimination. Moreover, the process of grayscale reconstruction will reach stability soon with respect to the result of watersheds. Typically, the number of iteration can be set to 10.

2. There are two multiscale gradient algorithms introduced in chapter 3. Typically, the multiscale gradient operator usually has the effect of producing thick edge. Comparing to the multiscale gradient algorithm introducing in section 3.2.2, the one described in section 3.2.1 could locate the edge more precisely without increasing the thickness of the edge. However, due to the shape of the flat structure element, it is hardly to be speeded up. On the other hand, the morphological gradient operator introduced in section 3.2.2 could be speeded up by performing the operation with each row followed by each column [41]. Actually, the Intel Image Processing Library, IPL, is used to facilitate the implementation of this operator. IPL provides a set of highly optimized C functions by taking advantage of the parallelism of the new SIMD (single-instruction, multiple-data) instruction of the latest generation of Intel processors. It provides simple dilation and erosion operators which are the maximum and minimum operators with the 8-neighbors of a given point. Recall the definition of the morphological gradient operator in section 3.2.2 and, moreover, $f \oplus B_2 = (f \oplus B_1) \oplus B_1$. Hence, it can be speeded up by means of simple morphological operators implemented in IPL.

3. Typically speaking, region-based classification can be faster than pixel-based classification if the over-segmentation of watershed transformation could be reduced by the local minima reconstruction. It is true when the input image is large and the searching area is wide. Nevertheless, the fine feature may be lost with larger regions. Hence, the result of watershed transformation controls the accuracy of region-based classification and user assistance may be taken into consideration through the setting of parameter h in the grayscale reconstruction since it dominates the result of watersheds.

4. The weak edge usually causes difficulty to locate the boundary precisely. As what you can see in the right of the head of "Akiyo", the real object boundary is hardly distinguished from the background even for the human eye. Thus, neither pixel-based nor region-based could accomplish this task. To solve this problem, motion information can be taken into consideration; this is what we called moving object segmentation.

5. As expected, the simple tracking works well in "Akiyo" sequence with little motion. That is because the object boundary in the previous frame is a good initial prediction and the adaptive morphological operators could find the searching area where the real object boundary resides. Simulation results are shown in Figure 5.3.

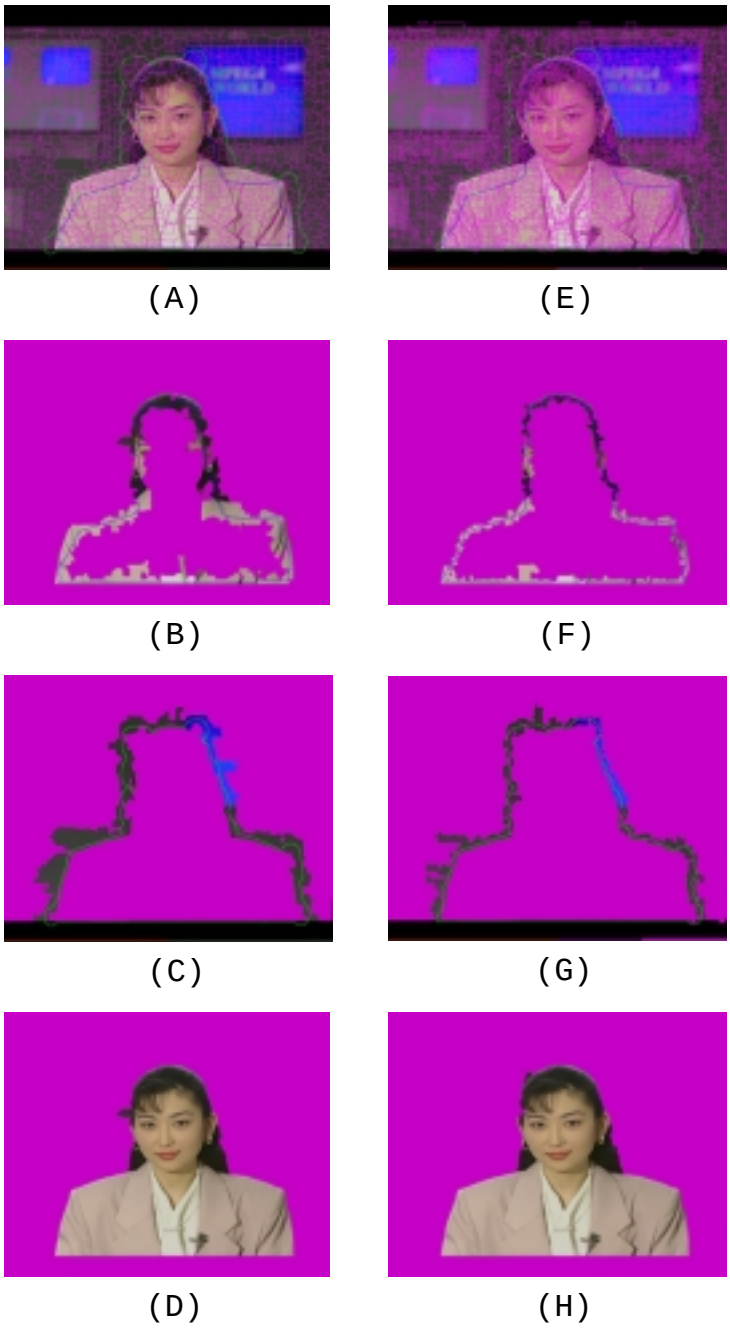


Figure 5.2: Akiyo Demo

Figure 5.3: Tracking: The 1st, 5th, 10th, 15th, 20th, 25th, 30th frames (up-down left-right)

Chapter 6

Conclusion and Future Work

6.1 Conclusion

While lacking high level image understanding, user assistance becomes acceptable in recent research works of image or video segmentation. In this thesis, the user gives the definition of a semantic object and the proposed system will try to locate the real boundary. Once extending to video object segmentation, the object with little motion can be tracked well due to the capability of adaptive morphological operators.

A supervised segmentation system based on mathematical morphology is presented in this thesis. It consists of three morphology-based modules: multiscale gradient algorithms, watershed transformation, and adaptive morphological operators. Multiscale gradient filters on the basis of mathematical morphology are robust to noise comparing to the traditional first-order differential operators or Laplacian operator. Two morphological gradient operators are introduced and each has its pros and cons. One can avoid the problem of producing thick edges but cannot be speeded up due to the specific shape of the structure element; the other is opposite. The watershed transformation is a useful tool for initial segmentation but usually yields over-segmentation because of too many local minima. To avoid this problem, local minima elimination is involved and dominates the result of segmentation. The adaptive morphological operators are proposed to precisely find out the searching area. In this thesis, the "maximizing edge strength" criterion is examined and the experimental result is acceptable. In the stage of "classification", two kinds of approaches using hierarchical queue are proposed: pixel-based

classification and region-based classification. Both yield acceptable results, and the region-based classification is faster once if the watershed transformation could create large homogeneous regions such that the region number is relatively low, especially in the case of large images.

6.2 Future Works

The following works could make the content of this thesis more complete:

1. The interface for user assistance can be designed such that the user could refine the segmentation result.

2. To overcome the problem from weak edges, motion information may be included. Moreover, using motion information can complete the module of object tracking.

3. To select a proper color space, color science and human visual characteristics should be considered.

Bibliography

[1] ISO/IEC JTCI/SC29/WG11 N3056. *MPEG-4 FDAM1*, December 1999.

[2] Robert M. Haralick and Linda G. Shapiro. "Survey: Image Segmentation Techniques". *Computer Vision, Graphics and Image Processing*, 29:100–132, 1985.

[3] Nikhil R. Pal and Sankar K. Pal. "A Review on Image Segmentation Techniques". *Pattern Recognition*, 26(9):1277–1294, 1993.

[4] L. Lucchese and S. K. Mitra. "Unsupervised Sgementation of Color Image Based on k-means Clustering in Chromaticity Plane". *IEEE. Proceedings*, pages 74–78, 1999.

[5] M. Celenk. "Colour Image Segmentation by Clustering". *IEE. Proceedings-E*, 138(5):368–376, September 1991.

[6] Jens-Rainer Ohm and Phuong Ma. "Feature-Based Cluster Segmentation of Image Sequences". *IEEE. Proceedings*, pages 178–181, 1997.

[7] Dorin Comaniciu and Peter Meer. "Robust Analysis of Feature Spaces: Color image Segmentation". *IEEE*, pages 1–2, 1997.

[8] Jongwon Kim Ju Guo and C.-C. Jay Kuo. "Fast Video Object Segmentation Using Affine Motion and Gradient-Based Color Clustering". *IEEE*, pages 750–755, 1999.

[9] A. Murat Tekalp Eli Saber and Gozde Bozdagi. "Fusion of Color and Edge Information for Improved Segmentation and Edge Linking". *IEEE.. Proceedings*, pages 2176–2179, 1996.

[10] Marie-Pierre Dubuisson and Anil K. Jain. "Object Contour Extraction Using Color and Motion". *IEEE. Proceedings*, pages 471–476, 1993.

[11] Andrew Witkin Michael Kass and Demetri Terzopoilos. "Snakes: Active Contour". *International Journal of Computer Vision*, pages 321–331, 1988.

[12] Touradj Ebrahimi Roberto Castagno and Murat Kunt. "Video Segmentation Based on Multiple Feature for Interactive Multimedia Applications". *IEEE. Trans. on Circuits and System for Video Technology*, 8(5):562–571, September 1998.

[13] Luc Vincent and Pierre Soille. "Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations". *IEEE. Trans. on Pattern Analysis and Machine Intelligence*, 13(6):583–598, June 1991.

[14] John M. Gauch. "Image Segmentation and Analysis via Multiscale Gradient Watershed Hierarchies". *IEEE. Trans. on Image Processing*, 8(1):69–79, January 1999.

[15] Serafim N. Nicos Maglaveras Kostas Haris and Aggelos K. Katsaggelos. "Hybrid Image Segmentation Using Watersheds and Fast Region Merging". *IEEE. Trans. on Image Processing*, 7(12):1684–1699, December 1998.

[16] Il Dong Yun Sang Ho Park and Sang Uk Lee. "Color Image Segmentation Based on 3D Clustering: Morphological Approach". *Pattern Recognition*, 31(8):1061–1076, 1998.

[17] Chuang Gu and Ming-Chieh Lee. "Semantic Video Object Segmentation and Tracking Using Mathematical Morphology and Perspective Motion Model". *IEEE. Proceeding*, pages 514–517, 1997.

[18] Chuang Gu and Ming-Chieh Lee. "Semiautomatic Segmentation and Tracking of Semantic Video Objects". *IEEE. Trans. on Image Processing*, 8(5):572–584, September 1998.

[19] Demin Wang. "Unsupervised Video Segmentation Based on Watersheds and Temporal Tracking". *IEEE. Trans. on Circuits and System for Video Technology*, 8(5):539–546, September 1998.

[20] T. Meier and K. N. Ngan. "Segmentation and Tracking of Moving Objects for Content-Based Video Coding". *IEE. Proc.-Vis. Image Signal Processing*, 146(8):144–150, June 1999.

[21] D. Zhong and S. f. Chang. "Video Object Model and Segmentation for Content-Based Video Indexing". *IEEE. International Symposium on Circuits and System*, pages 1492–1495, 1997.

[22] Di Zhong and Shih-Fu Chang. "AMOS: An Active System for MPEG-4 Video Object Segmentation". *IEEE. Proc.*, pages 647–651, 1998.

[23] Di Zhong and Shih-Fu Chang. "An Integrated Approach for Content-Based Video Object Segmentation and Retrieval". *IEEE. Trans. on Circuits and System for Video Technology*, 9(8):1259–1268, December 1999.

[24] Charles R. Giardina and Edward R. Dougherty. *Morphological Methods in Image and Signal Processing*. Prentice-Hall, Inc., 1988.

[25] Petros Maragos. "Tutorial on Advances in Morphological Image Processing and Analysis". *Optical Engineering*, 26(7):623–632, July 1987.

[26] Malay K. Kundu Bhabatosh Chanda and Y. Vani Padmaja. "A Multiscale Morphologic Edge Detector". *Pattern Recognition*, 31(10):1469–1478, 1998.

[27] Xudong Song and Yrjo Neuvo. "Robust Edge Detector based on morphological filters". *Pattern Recognition Letters*, 14:889–894, 1993.

[28] Philippe Salembier. "Morphological Multiscale Sgementation for Image Coding". *Signal Processing*, 38:359–386, 1994.

[29] Mary L. Comer and Edward J. Delp. "An Empirical Study of Morphological Operators in Color Image Enhancement". *Proceedings of the SPIE Conference on Image Processing Algorithms and Techniques III*, 1657:314–325, 1992.

[30] Mary L. Comer and Edward J. Delp. "Morphological Operators for Color Image Processing". *Journal of Electronic Image*, 8(3):279–289, July 1999.

[31] J. Serra. *Image Analysis and Mathematical Morphological*. Academic Press Inc., 1982.

[32] H. Digabel and C. Lantuejoul. "Iterative algorithms". *Proc. 2nd European Symp. Quantitative Analysis of Microstructures in Material Science, Biology and Medicine*, October 1977.

[33] S. Beucher and C. Lantuejoul. "Use of watersheds in contour detection". *Proc. Int. Workshop Image Processing, Real-Time Edge and Motion Detection/Estimation*, September 1979.

[34] S. Beucher. "Watersheds of functions and picture segmentation". *Proc. IEEE. Int. Conf. Acoustics, Speech, and Signal Processing*, May 1982.

[35] F. Maisonneuve. "Sur le partage des eaux". *School of Mines, Paris, France, Internal Rep CMM*, 1982.

[36] Demin Wang. "A Multiscale Gradient Algorithm for Image Segmentaion Using Watersheds". *Pattern Recognition*, 30(12):2043–2052, 1997.

[37] S.J. R.M. Haralick Lee and L. G. Shapiro. "Morphologic Edge Detection". *IEEE. J. Robot. Automat.*, 3(2):142–155, 1987.

[38] Luc Vincent. "Morphological Grayscale Reconstruction in Image Analysis: Applications and Efficient Algorithms". *IEEE. Trans. on Image Processing*, 2(2):176–201, 1993.

[39] C. S. Chen. *"Range Image Processing Using Mathematical Morphology and Its Application on 3D Object Recognition"*. PhD thesis, National Taiwan University, 1996.

[40] Daehee Kim Munchurl Kim, Jae Gark Choi, Hyung Lee, Myoung Ho Lee, Chieteuk Ahn, and Yo-Sung Ho. "A VOP Generation Tool: Automatic Segmentation of Moving Object in Image Sequence Based on Spatial-Temporal Information". *IEEE. Trans. on Circuits and System for Video Technology*, 9(8):1216–1226, December 1999.

[41] Robert M. Haralick and Linda G. Shapiro. *Computer and Robot Vison, volume 1*. Addison-Wesley, 1992.