Multiresolution Colour Image Segmentation

A Thesis

Submitted to the Committee

in Fulfillment of the Requirements

for the Degree of M. Sc.

in the

Department of Computational Science

University of Saskatchewan

Saskatoon, Saskatchewan

by

Jianqing Liu

January 1993

Contents

ABSTRACT

Image segmentation is the process by which the original image is partitioned into some meaningful regions. It is the foundation of higher level processing; e.g., object recognition. Although there are many segmentation methods, many of them are applicable only to monochromatic images. Colour provides useful information to the image segmentation process.

In this thesis, a novel multiresolution colour image segmentation (MCIS) algorithm which uses the Markov Random Fields (MRFs) is proposed. The proposed approach is a stochastic relaxation process that converges to the MAP (maximum a posteriori) estimate of the segmentation. The quadtree structure is used to implement the multiresolution framework, and a relaxation process is employed to control the splitting and merging of nodes, so as to minimize an energy function and therefore maximize the MAP estimate. The multiresolution scheme enables the use of different dissimilarity measures at different resolution level. Consequently, the proposed algorithm is very noise resistant. Since the global clustering information of the image is required in the proposed approach, the scale space filter (SSF) is employed as the first step. In addition, the image is segmented coarsely using SSF. The multiresolution approach is used to refine the segmentation. The effects of using different parameters, dissimilarity measures and colour spaces are examined in the experiments. In order to evaluate the segmentation results of both synthesized and real images quantitatively, a new evaluation function is developed.

Chapter 1

Introduction

One of the goals of computer vision is to construct explicit and meaningful descriptions of physical objects from images [2]. Before the objects in the image can be recognized and manipulated, images must be organized into subimages that are likely to correspond to "interesting objects". Image segmentation is the process by which the original image description is partitioned into some meaningful regions. It is usually assumed that the objects or substantial parts of the objects are represented by regions that are homogeneous in some sense (e.g., brightness, colour, texture, etc.). Many segmentation procedures have been proposed to partition an image into homogeneous regions or segments. Yet, no sufficiently rigourous and general solution of the segmentation problem is available. In fact, the segmentation problem has not been formulated in rigourous terms. Procedures are mostly heuristic and sometimes interactive. Furthermore, most of the segmentation algorithms are applicable only to monochromatic images.

It is noteworthy that the development of an efficient segmentation algorithm — avoiding exhaustive enumeration — for the analysis of large images is also important. It is common practice that segmentation procedures are designed for specific applications, and the quality of the segmentation result is determined qualitatively by an expert. In some cases, however, a specific segmentation procedure (or the underlying principles) can be applied in other areas as well. Segmentation is linked with data aggregation, in particular, it includes clustering. The aggregation of complex data is often endowed with a definite structure, e.g., a graph.

Colour is a useful property which adds more than beauty to a scene — it adds information about the physical properties of objects. Humans seem to be good at using colour information not only to segment a scene but also to spot regions of interest. It appears that colour can also help in the design of the segmentation process [32].

Colour correlates with the class identity of an object because pigments form part of the appearance of an object. Animals make use of colour to send messages of enticement or warning [31]. In an artificial environment, colour is used as a trademark or identification feature, such as in packing goods, advertising signs and road signs.

Colour has not been an important topic among researchers in the field of computer vision, perhaps because of the following reasons [14] [31]:

1. Until recently, colour monitors were expensive and colour images were relatively difficult to obtain.

2. Computation is expensive because more information has to be stored and processed with colour images than with the same image in black and white. Hence, it is required to have high disk storage and fast computers.

3. There were no good algorithms for colour constancy, that is, perceiving stable colours over varying light conditions, as people do in most circumstances.

4. Colour is not regarded as essential to the class identity of the object. Geometrical cues are considered to be the most reliable indication of the object's identity. While this may be true in general, it may not be true for some cases in which colour may be a far more distinctive feature.

With the advancement of computer hardware, some of the technological problems mentioned above have been solved. The price of hardware which includes memory and colour monitors has dropped drastically during the last five years. Also, there is some great progress in the development of algorithms for colour constancy. Besides, it was noticed that in some cases in which geometrical cues are not of interest, colour is a more reliable feature for the segmentation process [31]. Hence, it is believed that colour can be used in identifying object surfaces, even under varying lighting conditions.

In this thesis, a new multiresolution colour image segmentation (MCIS) algorithm which uses the multiresolution Markov Random Fields (MRFs) is proposed. This approach is a stochastic relaxation process that converges to the MAP (maximum a posteriori) estimate of the segmentation. The quadtree structure is used to implement the multiresolution framework. Since some global information is needed in this algorithm, the scale space filter (SSF) [25] is employed as the first step. The organization of this thesis is as follows. Some background knowledges about colour and image segmentation are presented in Chapter 2 and 3, respectively. Chapter 4 describes the proposed MCIS algorithm. Chapter 5 introduces the new evaluation measure and other experimental results. Chapter 6 is the summary and future work.

Chapter 2

Colour Theory

It is well known that almost all perceived colours can be produced by a combination of three differently coloured lights. Accurate measurements of the light of the three colours needed to match any given colour is the basis of colourimetry, or colour matching. From the results of such measurements on typical colours selected from the spectrum, components necessary to match any colour of known spectral composition can be computed. The three components are best chosen from near the middle and near the two ends of the visible spectrum. Within these regions the choice of the components is unlimited. The components need not be very narrow bands in the spectrum. Once the components are selected, a wide variety of colours can be produced by suitably adjusting the amounts of the three components. These amounts are called the *tristimulus values* of the colours produced by the mixture. There are many sets of colour components. Some of them are listed below.

2.1 RGB colour space

Traditionally, colour has been specified with the RGB triplets [9]. This model is based directly on the three types of phosphors and implicitly represents a colour sampled at three wavelengths — one in the short range (blue), one in the medium range (green) and one in the long range (red). If the spectral energy distribution of a light source is E_λ, then this colour can be represented by integrating the appropriate colour matching function , i.e., $\overline{r}_\lambda$, $\overline{g}_\lambda$ or $\overline{b}_\lambda$, with the spectral energy distribution E_λ [13] :

$$R = \int_\lambda E_\lambda \overline{r}_\lambda d\lambda$$

$$G = \int_\lambda E_\lambda \overline{g}_\lambda d\lambda$$

$$B = \int_\lambda E_\lambda \overline{b}_\lambda d\lambda$$

The colour matching functions for RGB space is shown in Figure 2.1(a).

The drawback of this space is that some values of the colour matching functions for the red colour are negative and hence can not be produced in reality.

2.2 XYZ colour space

CIE (International Commission on Illumination) defines a set of primaries with positive colour matching functions across the entire visible spectrum (Figure 2.1(b)). The primaries, X, Y and Z are strictly imaginary and have no correspondence to any monochromatic wavelengths in the visible spectrum. One advantage is that all colour matching and specification can be accomplished with positive coefficients. Another advantage is that there is a great body of published data on the CIE primaries.

2.3 HSV colour space

Hue (H), the colour aspect of a visual impression, normally corresponds to the stimulation of the retina by light within a narrow range of wavelengths. Saturation (S)

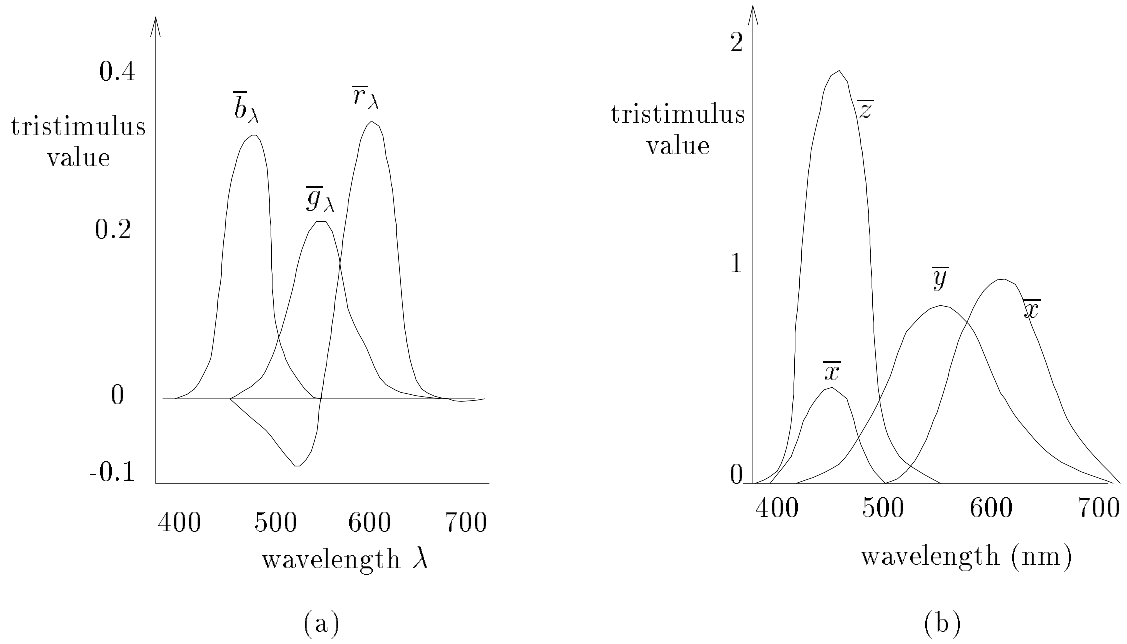


Figure 2.1: (a) Colour match functions in the RGB primary space, where R=700 nm, G=546.1 nm, and B=435.8 nm (b) Colour matching functions in the primary space of the CIE (International Commission on Illumination).

signifies the relative content of whiteness of a stimulus perceived as having a particular hue. A fully saturated colour is of "pure" hue. V is the intensity value. It is an attribute of light or of surface colour which may be ordered on a scale from dim to bright. It is the brightness factor that turns orange into brown and gray into black or white. The scale for brightness correlates most highly with the physical intensity of light. There are some other colour metrics which can be obtained through some transformations of the above colour spaces.

2.4 Colour transformations

Typically, a colour image is composed of three spectral components obtained using filters with different sensitivities in the red, green and blue wavelengths [6]. In many cases, it is desirable to transform from the resulting red, green and blue (RGB) images to other colour spaces which separate intensity from colour (hue). It is argued that linear transformations from RGB data are preferable. Nonlinear transformations such as hue, saturation and intensity value (HSV) and normalized colour suffer from nonremovable singularities and spurious modes and gaps, because when $R + G + B$ is small, the transformed values become unstable and meaningless [21] [28]. However, with these constraints in mind, non-linear transforms often provide more useful information by separating intensity and hue information, as in the HSV space [6].

2.5 Colour space selections in segmentation

Various colour spaces have been used in colour image segmentation. Celenk [5] claims that it is desirable to use colour space possessing uniform characteristics, such as Lab, for segmentation. In a uniform colour space, the same distances in the colour coordinates cause the same perception differences. But he does not do any experiment to compare the segmentation results of using this uniform colour space with the results of using other colour spaces. Ohta, Kanade, and Sakai [28] suggest that histograms should not be computed individually on the red, green, and blue (R, G, and B) variables because each of the three variables has a strong factor of intensity and therefore they are strongly correlated. They derive a new colour space by the Kauhunen Loeve transformation, an orthonormal transformation using the eigenvectors of the covariance matrix of the R, G and B distributions in the image. The colour space is called $I1, I2'$ and $I3'$, where $I1 = (R+G+B)/3, I2' = (R-B)/2$, and $I3' = (2G-R-B)/2$. Gonzalez-Rodriguez *et al.* [15] argue that UVY, the colour TV transmission primaries, has a greater discriminatory capacity than the others. Sarabi and Aggarwal [30] map the pixels of an image into XYI, the normalized colour space, and the histogram analysis is performed in this space. But the reason for using this colour space is not mentioned, and no experiment to compare the results using other colour spaces is presented.

Chapter 3

Previous work on colour image segmentation

Image segmentation is the process by which the original image is partitioned into certain meaningful regions. Identifying such regions is difficult in general.

3.1 Criteria of image segmentation

There are several criteria to judge if an image segmentation is good or not [16]:

1. Regions of a segmented image should be uniform and homogeneous with respect to some characteristics such as colour or texture.

2. Regions should be simple and should not contain many small holes.

3. Adjacent regions should have significantly different characteristics.

4. Boundaries of each region should be simple, not ragged, and must be spatially accurate.

Achieving all these desired properties is difficult because strictly uniform and homogeneous regions are typically full of small holes and have ragged boundaries.

Insisting that adjacent regions have large differences in values can cause regions to merge and boundaries to be lost. So there is always some compromise in order to get a good segmentation result.

Since colour images contain more information than the monochromatic images, using the colour information beneficially is one of the major concerns in colour image segmentation.

Colour image segmentation techniques can be roughly classified into three types, histogram-based segmentation, neighbourhood-based segmentation and physically-based segmentation methods.

3.2 Histogram-based segmentation

Histogram-based segmentation uses the clustering process to define a partition in the measurement space. Each pixel is assigned a label in the measurement space partition. A region is defined as a connected set of pixels having the same label [16].

Histogram-based segmentation assumes that homogeneous objects in the image manifest themselves as clusters in the measurement space. Image segmentation is accomplished by mapping the cluster back to the image domain where maximal connected components of the mapped-back clusters constitute the image regions. For black and white images, calculation of the histogram in an array is straightforward. The measurement space clustering can be accomplished by determining the valleys in the histogram and declaring the clusters to be the intervals of values between valleys.

For colour images, one way to use colour information is to process each of the R, G, and B histograms separately. But this approach can not reflect the correlation among different colour components.

Ohlander, Price and Reddy [27] use histogram analysis in a recursive way. Their method begins by selecting a mask containing all the pixels in the image. Given any mask, the histogram of the image covered by the mask is computed. In order to select a colour space with greater discriminatory capacity, nine redundant colour features, RGB, YIQ (or UVY, the colour space for TV signals) and HSV (hue, saturation and intensity value) will be used for colour image segmentation. Then the best peak in the set of histograms formed by the nine colour features is selected. This enables the separation of one cluster of the histogram from another cluster. Pixels in the image are then identified with the cluster to which they belong. If there is more than one cluster, then each connected component of all the pixels within the same cluster is used to generate a mask which is placed on a stack. During successive iterations the next mask on top of the stack is popped and is used to select pixels in the histogram computation process. Clustering is repeated for each new mask until the stack is empty. The advantage of this method is that it is capable of generating partial (or coarse) segmentation as well as detailed segmentation of an image. The disadvantage is that it needs to compute the histograms of all the features, some of which may be redundant. This is more time-consuming.

Witkin [34] proposes a method, called the *Scale Space Filter* (SSF), for histogram analysis which enables the most prominent and stable peaks in the histogram to be found. Huang et al. [20] use SSF in the colour image segmentation, and each of the R, G and B components is processed individually. Lim and Lee [25] propose a method of combining the clustering information into the three dimensional colour space.

Celenk [5] proposes a method that uses the uniform colour coordinate space Lab. Image clusters in some circular-cylindrical decision elements of the colour space are detected. Surfaces with constant lightness and constant chromaticity loci are used as decision elements. Each surface is obtained by using the histogram of only one of the L^*, H° and C^* cylindrical coordinates of the image data or the extracted feature vector. Celenk's algorithm starts from the 1D histogram of the space, detects the most prominent cluster ($M1$) and its neighbour ($M2$) in some ranges of a particular colour component. Then the two prominent clusters, $M1$ and $M2$, are projected onto the other two colour coordinates and their respective 1D histograms are used to find the spatial distributions of the two detected clusters. And then the estimated colour distributions of $M1$ and $M2$ are projected onto the line of Fisher discriminant for 1D thresholding. The above processes are repeated until no decisive 1D peak exists.

The advantage of the histogram-based segmentation is that it does not need *a priori* information of the image. Unfortunately, the approach suffers from the lack of local spatial knowledge. As a result, the regions produced are not smoothly bounded, and they often have holes, giving the effect of "salt-and-pepper" noise [16].

3.3 Neighborhood-based segmentation

Statistical methods, such as the classical Bayes decision theory, have been quite popular in image segmentation. However, these methods depend on global *a priori* knowledge about the image contents and organization.

Stochastic relaxation and Markov Random Fields (MRFs) have been used in image segmentation and image restoration [12] [11] [3] [6] [26] . MRFs provide an elegant means of specifying the interaction of neighbours and embody the expected dependencies of neighbouring pixels including both the prior and posterior probabilistic distributions. This local neighbourhood-based specification of dependencies avoid *ad hoc* methods using global image knowledge.

Markov Random Fields possess several characteristics which are useful in colour image segmentation. Properties such as smoothness and continuity of colour regions over an entire image can be enforced using only dependencies among local neighbours. Discontinuities which separate regions of constant colour may be computed while smooth regions are being found. In addition, the inclusion of both the prior and posterior distributions (through Bayes' rule) establishes a relationship between the noisy observed image and the result from the segmentation.

Briefly, a Markov Random Field is a lattice of sites, e.g., a two-dimensional array of pixels [6]. Since MRFs are stochastic processes, the pixels in an image may take on any of their allowed values, which means that all images can be generated. In addition, the conditional probability of a particular pixel having a certain value is a function only of the neighbouring pixels, not of the entire image. The Hammersley-Clifford theorem establishes the equivalence between the conditional probabilities of the local characteristics in the MRFs and local energy potentials in a Gibbs distribution [11]. Therefore, the *a priori* probability that the MRF is in a particular state can be calculated by summing the local energies over the entire image. In the proposed research, it is required to determine the MRF state that maximizes the *a posteriori* probability of the final segmentation given the observed data. By Bayes' theorem, the *a priori* energies can be added to an *a posteriori* energy term involving the difference between the actual observed data and the current MRF state (or predicted image). Therefore, the Gibbs distribution energy function consists of two parts, one describing the interaction potential between neighbours, and the other describing the difference between the predicted image and the observed data. Several methods of minimizing the energy function over the image (i.e. maximizing the probability) can be used, among them simulated annealing, deterministic procedures, and network solutions [6].

Besag [3] uses the local MRFs for the reconstruction of the true scene from "dirty" pictures. The basic assumption is that pixels close together tend to have the same or similar colours. It is assumed that the local characteristics of the true scene can be represented by a non-degenerate MRF. A method called the *Iterated Conditional Modes* (ICM) is proposed to estimate the true scene. ICM is equivalent to instantaneous freezing in simulated annealing. The most distinguishing feature of ICM is its rapid convergence. It is sufficient for almost all but some very difficult images [12].

Daily [6] has used MRFs in colour image segmentation of natural outdoor scenes. He has developed and compared the use of three different lattice structures for coupled MRFs with line processes and colour processes based on squares, hexagons and triangles. The simplest lattice structure is a rectangular grid, and the line processes are defined over some neighbourhoods of a pixel. By adjusting the energy function, larger cliques (neighbours of a site) encompassing additional directional and topographical structures can be heuristically added. For example, line process energies which penalize the formation of adjacent parallel discontinuities may be added. For the rectangular case, the line process can be broken into two components, horizontal and vertical, corresponding to the nearest neighbours in the 4-connected system. Each of the four neighbours is of equal distance from the central pixel and is weighted identically. Two kinds of colour differencing method are used and compared in this paper. One is the Euclidean RGB metric and the other is the Maximum HSV method. The Maximum HSV method is claimed to be more sensitive to colour changes and hence causes over-segmentation.

Geman *et al.* [12] propose a statistical framework for finding boundaries and for partitioning scenes into homogeneous regions. Their model is applied mainly to the problem of texture segmentation. The model is a joint probability distribution for the array of pixel gray levels and an array of "labels". In boundary finding, the labels are binary; in partitioning, the label values are *generic*. The distribution incorporates a measure of disparity between certain spatial features of pairs of blocks of pixel gray levels, using the Kolmogorov-Smirnov nonparametric measure of difference between the distributions of these features. Large disparities encourage intervening boundaries and distinct partition labels. The number of model parameters is minimized by *forbidding* label configurations that are inconsistent with prior beliefs, such as those defining very small regions, or redundant or blindly ending boundary placements. Probabilities of forbidden configurations are assigned with the value zero. The MAP (*maximum a posteriori*) estimator of boundary placements and partitionings is examined. The forbidden states introduce constraints into the calculation of these configurations. Stochastic relaxation methods are extended to accommodate *constrained* optimization.

Huang et. al [20] use the MRFs in colour image segmentation, and the relaxation processes of using the simulated annealing and ICM are compared. Although the algorithm is used for colour image segmentation, each of the R, G and B components is processed individually, and then combined as a segmented image. Dubes et al. [7] also compare some image segmentation algorithms based on MRFs models. The images for segmentation are monochromatic, and it is assumed that there are only two classes of objects in the images.

Muzzolini, Yang and Pierson [26] present a new Multiresolution Texture Segmentation (MTS) approach which generalizes the conventional simulated annealing method to a multiresolution framework and minimizes an energy function which is dependent on the resolution of the size of the texture blocks in an image. Based on *a priori* information, the MTS algorithm selects the best feature to determine if a texture block is homogeneous or not. If it is homogeneous, then no splitting is required; otherwise splitting of the texture block into four children is performed. Based on a similar argument, merging is performed by examining the similarity of a texture block with its neighbours.

Taxt and Bolviken [33] use single particle models from quantum physics to restore and segment images. The basic idea is to associate with each pixel i a physical system consisting of a single virtual particle with a certain amount of energy E_i and a potential function V_i between the pixel and its neighbours. One of the main ingredients of quantum mechanics is that the behaviour of such a particle is probabilistic. If its motion is confined to N_i (N_i includes pixel i and its neighbours), the position of this particle can be expressed as a probability distribution over N_i. The following restoration process is repeated iteratively. Suppose that the particle (pixel i) can move to a certain position j within N_i with probability p_{ij}, then a local restoration at pixel i

has the form

$$\hat{g} = \sum_{j \in N_i} p_{ij} g_j$$

where $\hat{g}$ is the estimated true value of pixel i. The probabilities p_{ij} sum to 1 over all j ($j \in N_i$).

Neither the histogram-based segmentation nor the neighbourhood-based segmentation has used underlying physical models of the colour image formation process in developing colour difference metrics. Physically-based algorithms have produced good segmentation for colour images obtained under controlled conditions [22],[23],[24], [19].

3.4 Physically-based segmentation

Most colour image segmentation methods that are based on a very simple interpretation of colour changes in an image generally segment images not only along material boundaries but also along other lines exhibiting colour or intensity variations, such as highlight and shadow boundaries, or object edges with significant shading changes. To solve this problem, Klinker, Shafer and Kanade [22] [23] [24] present an approach to segment surfaces with colour variations due to highlights and shading. They establish a model called the Dichromatic Reflection Model.

Materials can be divided into two classes on the basis of their optical properties, that is, optically homogeneous and optically inhomogeneous materials [18]. Optically homogeneous materials have a constant index of refraction throughout the material. Metals, glass, and crystals are the most common examples of homogeneous materials. Optically inhomogeneous materials are composed of a vehicle with many embedded colourant particles that differ optically from the vehicle, e.g., plastics, paper, textiles and paints.

When light is incident upon the surface of an object, some fraction of it is reflected.

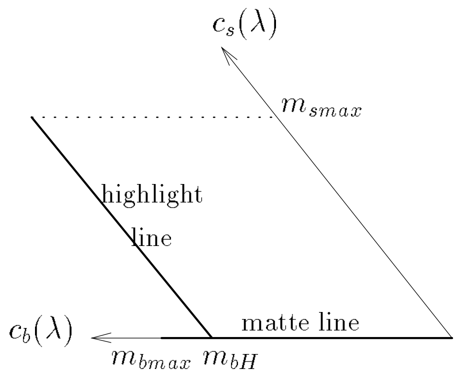


Figure 3.1: Spectral cluster on the dichromatic plane spaned by C_s and C_b

The part of reflection which is reflected at the interface in the direction such that the angle of incidence equals the angle of reflection is called *surface reflection*. The fraction of the incident light which is reflected after traversing through the body of the object is called *body reflection*.

The Dichromatic Reflection Model (DRM) describes the total reflection R, which is the amount of light reflected from the surface of a dielectric and nonuniform material, as a linear combination of the surface reflection and body reflection. The reflected light $L(\lambda, g)$ is expressed as:

$$L(\lambda, g) = m_s(g)C_s(\lambda) + m_b(g)C_b(\lambda)$$

where λ is the wavelength, g is the geometric factor, and m_s and m_b are the spectral power distribution of the surface reflection and body reflection, respectively. C_s and C_b are geometric scale factors. The points that are in the surface reflection areas are called *highlight points*. The remaining object points are *matte points*, the intensity of which is determined mainly by the body reflection process. Figure 3.1 shows the shape of the corresponding spectral histogram in the dichromatic plane. The combined spectral cluster of matte and highlight points resembles a skewed-T.

The segmentation method using DRM alternates between generating optical hypotheses from local image data and verifying whether the hypotheses fit the image. First, a linear hypotheses which corresponds mainly to matte areas is generated and exploited to segment the image coarsely. The highlight areas have not been analyzed at this stage. In order to refine the segmentation, a planar hypotheses is used. The feature of the skewed-T's is used to distinguish the matte and highlight regions.

Healey [17] [18] [19] uses the DRM and the normalized colour space in colour image segmentation. First, he uses conventional discriminant functions to segment the image. Then the DRM is applied to examine whether or not a region corresponds to a highlight region. If it is a highlight region, it is merged into its neighbouring regions. Otherwise, it corresponds to a material different from that of its neighbouring regions. In [18], Healey claims that the advantage of using normalized colour is that the geometric factors are removed in the normalized colour. But in [19] he admits that the normalized colour corresponding to a point in a shadow will be different from the normalized colour of a directly illuminated point of the same material. That is to say, the normalized colour still depends on the geometry. Therefore, the advantage of using normalized colour is doubtful.

There are many limitations of the DRM. DRM is applicable only to some inhomogeneous dielectric materials. Also, it makes several simplifying assumptions which restrict the illumination conditions [24]. It is assumed that the imaged scene is illuminated by a single point source with a single spectral power distribution. For many realistic scenes the assumptions do not hold. Therefore, the DRM model can be used only to understand scenes taken under very restricted conditions.

Chapter 4

The proposed MCIS algorithm

The advantage of the histogram-based segmentation is that it does not need any *a priori* information on the image. But the approach suffers from the lack of spatial knowledge. The neighbourhood-based approach uses local information instead of the global one. In order to use this approach, however, some cluster information about the image must be available in advance.

A new multiresolution colour image segmentation (MCIS) method is proposed that is a combination of the above two approaches. First, the histogram-based technique is used to get a coarse segmentation of the image, and to provide some global information about the image, e.g., the number of clusters in the image and the position of each cluster in the colour space. The scale space filter (SSF) is applied for this coarse segmentation [25]. And then, the proposed multiresolution scheme is used to refine the segmentation. The multiresolution scheme of the MCIS uses part of the ideas of MTS, the Multiresolution Texture Segmentation algorithm [26]. The major difference is that MTS is applicable to the monochrome texture image, while the MCIS is a general-purpose colour image segmentation algorithm. The essence of the MCIS approach is a relaxation algorithm which generates a sequence of configurations that converges to the MAP (Maximum a posteriori) estimate.

The proposed work consists of the following parts:

1. Develop the MCIS for colour image segmentation, the SSF is used as the first step;
2. Develop a new criterion to evaluate segmentation results quantitatively;
3. Use the new evaluation criterion to examine the performance of MCIS under different parameters. The effect of different dissimilarity measures, simulated annealing and some other options in MCIS are studied;
4. Use different colour spaces in MCIS, and compare their effects on segmentation.

The first part of the proposed work, the MCIS algorithm, is introduced in this chapter. The other three parts will be discussed in the later chapter. The organization of the proposed algorithm is shown in Figure 4.1

4.1 Using Scale Space Filter for the coarse segmentation

To use a histogram to segment an image, the significant peaks in the histogram are to be determined. It has been known that the derivatives, the extrema and the interval bounded by the extrema of a signal are useful entities for a qualitative analysis of a physical signal [25]. For this purpose, Witkin [34] proposes a method called the *scale space filter (SSF)* for histogram analysis. In this method, the Gaussian convolution of a histogram $f(x)$ is used to smooth it. The convolution is given by

$$F(x,\sigma) = f(x) * g(x,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{\infty} f(u) e^{-\frac{(x-u)^2}{2\sigma^2}} du \tag{4.1}$$

where "$*$" denotes convolution with respect to x. x is the grey scale, σ the standard deviation of the Gaussian (called *scale* constant).

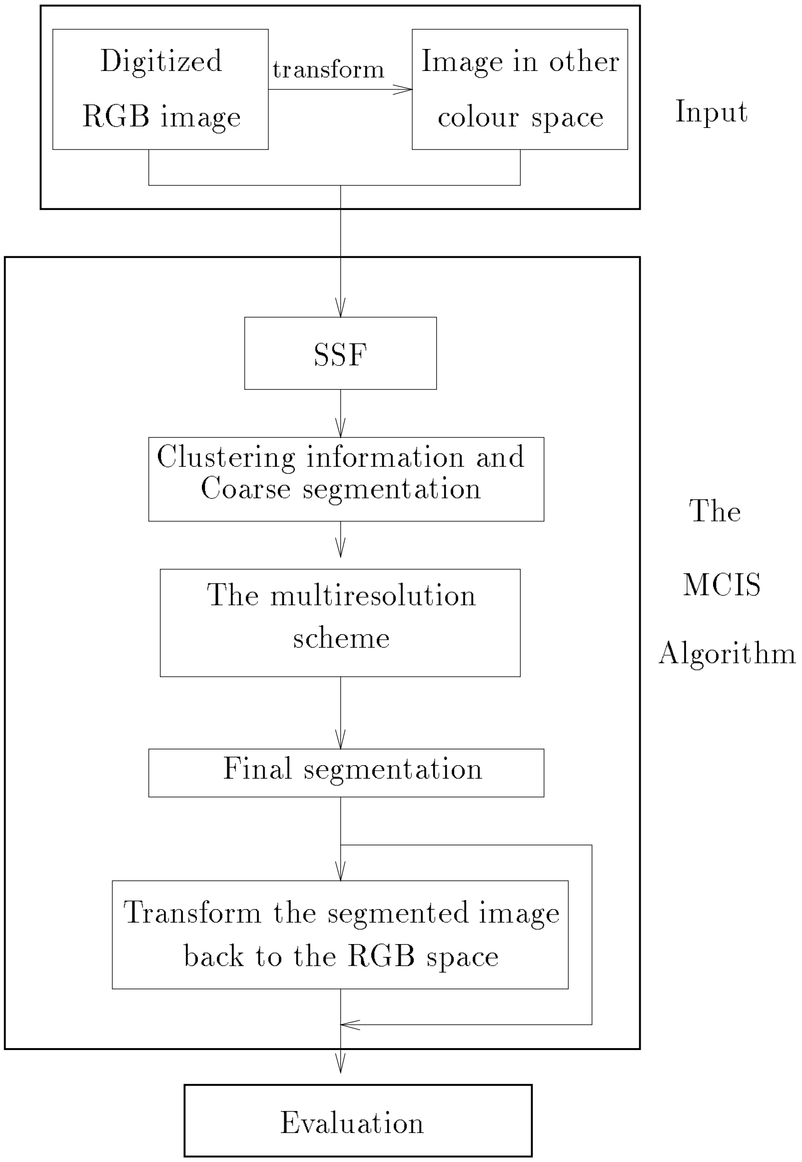


Figure 4.1: Organization of the algorithm.

$F(x, \sigma)$ defines a surface which swept out as σ is smoothly varied. The space defined by (x, σ) is called the *scale space*, and the surface, F, defined in equation (4.1), the *scale space image* of f. It is well-known that the zero-crossing information of the derivatives of a signal is important to finding the peaks and valleys of f. Also, pairs of zero-crossings represent meaningful events such as peaks and valleys. The zero-crossing locations of $F_{xx}(x, \sigma)$, where

$$F_{xx}(x, \sigma) = \frac{\partial^2}{\partial x^2} F(x, \sigma) \tag{4.2}$$

are commonly called the *fingerprint* of F.

Decreasing the scale constant, σ, has two effects on the locations of zero-crossings: the continuous motion of existing zero-crossings, and the appearance of new ones in pairs. It is very important to select a proper scale in the analysis of the histogram. The reason is that pairs of zero-crossings at lower scale could be meaningless local extrema and those at higher scale might not represent all the desired peaks and valleys [25]. Based on these facts, Witkin [34] developed a multiscale representation and a method for extracting the perceptually salient features such as valleys and peaks of a signal. Since the goal is colour image segmentation, consideration should be given to combining from the three 1D clusters into the three-dimensional (3D) colour space. It is assumed that the colour space is X1-X2-X3, where X1-X2-X3 can be RGB, HSV, XYZ, etc..

The scale space filter method for colour image segmentation consists of the following steps: First, the fingerprint of a histogram is obtained by recording the positions of zero-crossings of the second derivative. And then, the interval tree is established from which the active nodes are selected to partition the histogram. Thirdly, the information from each of the histogram in the X1, X2 and X3 colour component is combined to form clusters in the 3D colour space. Finally, each pixel in the image is labeled according to the clustering information. Details are as follows:

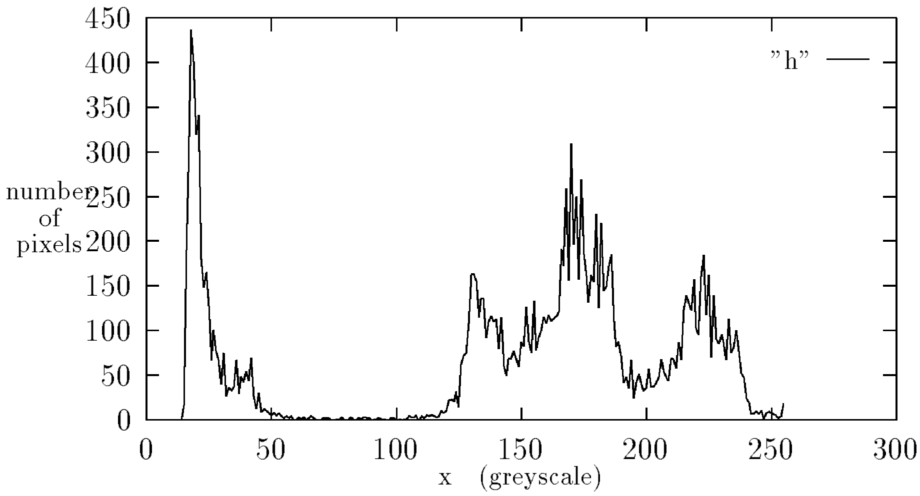
- Step 1 — to get the fingerprint of a histogram

  The original histogram (Figure 4.2(a)) is smoothed by taking the Gaussian convolution (Figure 4.2(b)) at a certain scale. Then the first and second derivatives of the smoothed histogram are computed (Figure 4.3). The positions of the zero-crossings are recorded. Each zero-crossing is given a flag to indicate whether it corresponds to the rising part of a peak or to the falling part of a peak. The above process is repeated at different scales, e.g., σ from 0 to σ_{max}, where σ_{max} is a predefined threshold. With the decrease of the scale σ, the positions of zero-crossings will shift, and some new pairs of zero-crossings will appear. These zero-crossings at different scales constitute the fingerprint of the histogram (Figure 4.4(a)).

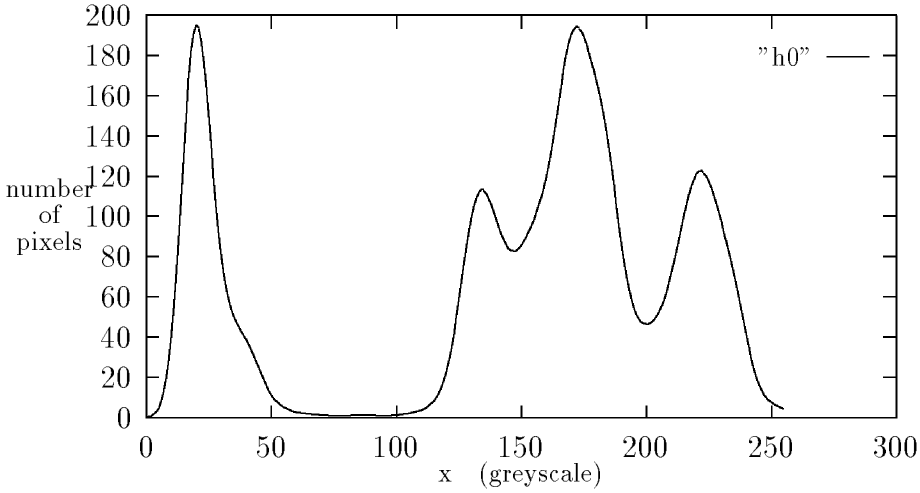
- Step 2 — to establish the interval tree and to select active nodes

  The scale space extrema whose scales exceed a given threshold, σ_T, partition the x axis into intervals. As σ is decreased, each new pair of zero-crossings divides the interval into three subintervals [34]. Each of these intervals defines a rectangle in the scale space, bounded above by the scale at which the interval emerges, bounded below by the scale at which new intervals emerge, and bounded on the left and the right sides by the x-coordinates of the events. Such intervals correspond to nodes in a tertiary tree, which is called the *interval tree* (Figure 4.4(b)).

  A node is defined to be an *active node* if its height, i.e., the distance between the top and the bottom bounds, is greater than the mean height of its offsprings (Figure 4.4 (c)). The justification of selecting active nodes is that the height of a node in the (x, σ) space is a measure of its persistence or stability. *Maximum stability descriptions* can be obtained by starting from the root of the tree, and moving down the tree recursively if the stability of a node is less than the average stability of its children. Active nodes represent local maxima, each of

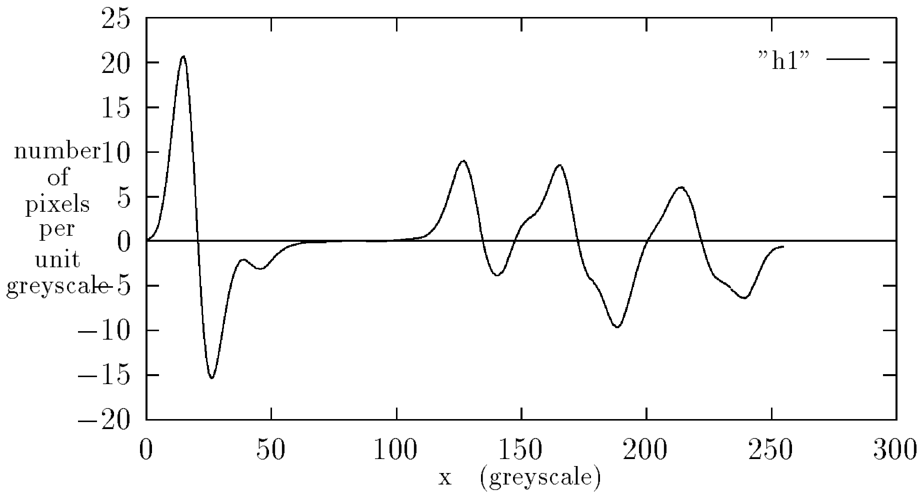


. (a)

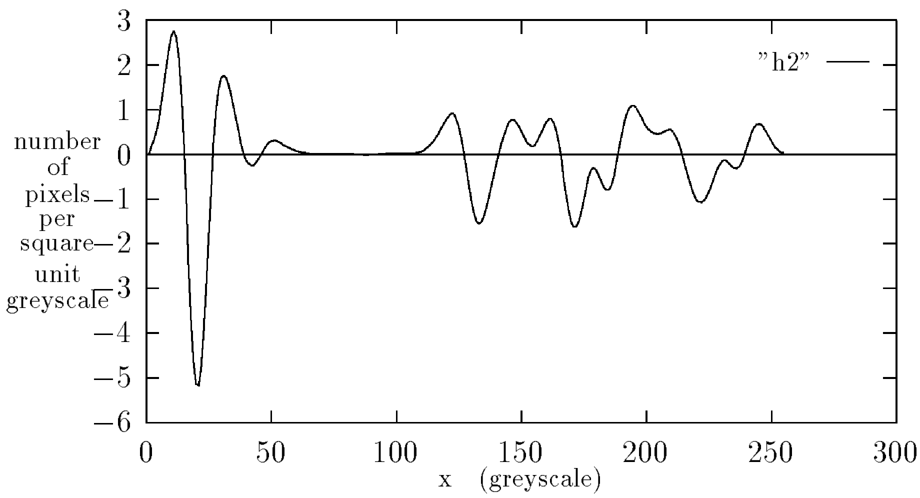


. (b)

Figure 4.2: (a)A typical histogram taken from an image; (b) the histogram smoothed by the Gaussian convolution; $\sigma = 5$.



.
(a)



.
(b)

Figure 4.3: (a) The first derivative of the smoothed histogram; (b) the second derivative of the smoothed histogram.

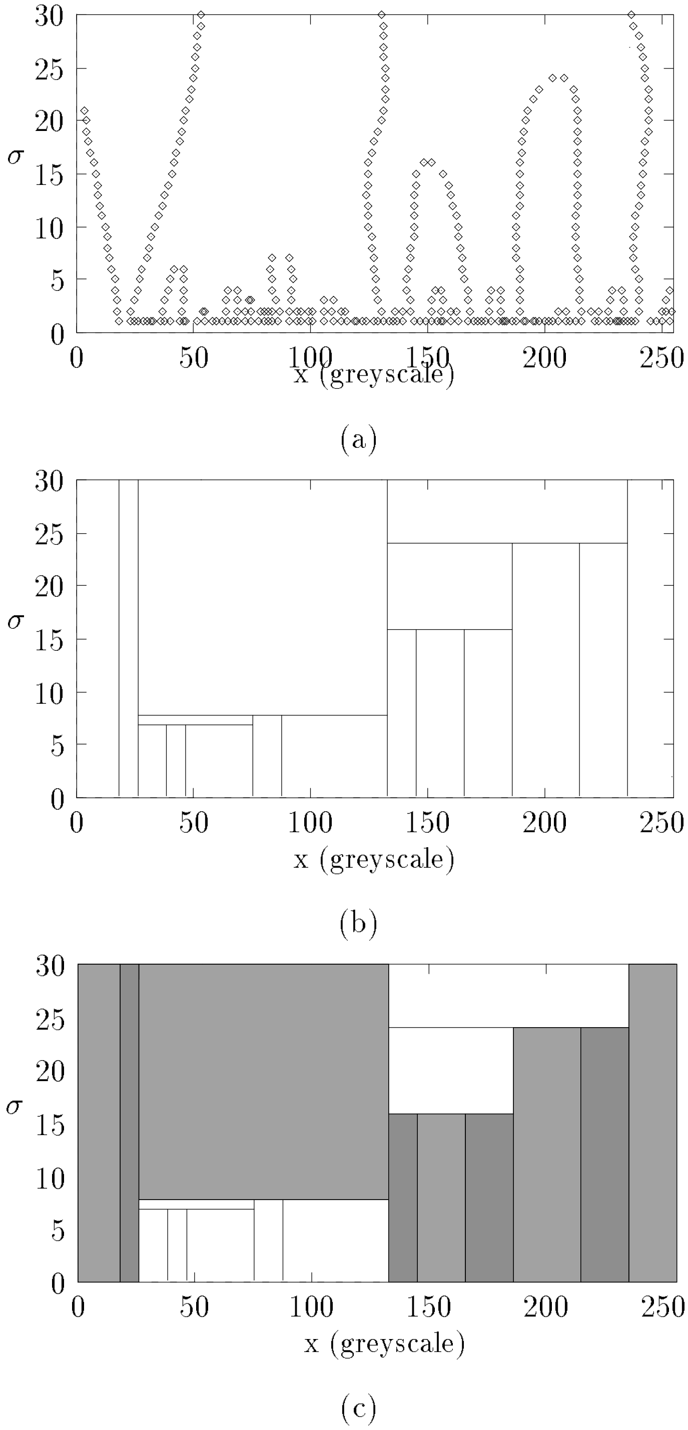


Figure 4.4: (a) the fingerprint of the histogram with maximum $\sigma = 30$; (b) the interval tree established from the fingerprint; (c) the active nodes in the interval tree

which may be refined successively at finer scales. Active nodes are selected to partition the histogram. Those active nodes whose left bounds are the rising parts of the histogram correspond to the peaks in the histogram. The lower and the upper thresholds are defined to be the left and right bounds of such nodes, respectively:

$$low_{X1}(i) = left\_bound(i)$$
$$up_{X1}(i) = right\_bound(i)$$
$$i = 0, 1, ..., M-1$$

where $low_{x1}(i)$ and $up_{x1}(i)$ are the lower and upper thresholds for the ith cluster in the X1 colour component, e.g., the "red" component in the RGB space. M is the number of clusters , or significant peaks, in the histogram. The ith cluster in one of the colour spaces, e.g. X1, is defined as the range between the lower and upper thresholds:

$$U1_{X1}(i) = [low_{X1}(i), up_{X1}(i)] \tag{4.3}$$

- Step 3 — to find clusters in the colour space

  Up to now, clusters in a monochrome image can be defined. Since the goal is to segment colour images, a consistent way to combine the 1D X1, X2 and X3 information and get the 3D clusters in the colour space is required. The proposed approach is to apply the above procedure to the histogram of each colour component. The X1, X2 and X3 colour space is partitioned by:

$$U(i, j, k) = U1_{X1}(i) \otimes U2_{X2}(j) \otimes U3_{X3}(k) \tag{4.4}$$

where $U1_{X1}(i), U2_{X2}(j)$ and $U3_{X3}(k)$ denote a cluster in the X1, X2 and X3

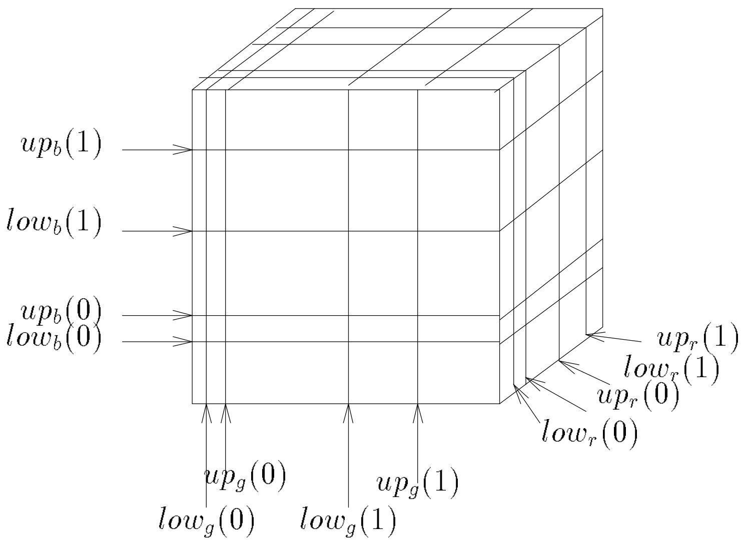


Figure 4.5: The colour space which is partitioned by the lower and upper thresholds.

space respectively, as defined in equation (4.3). $\otimes$ denotes the cross-product operator. Figure 4.5 shows the RGB colour space which is partitioned into several boxes. But not every combination of the three colour components corresponds to a cluster in the colour space. The significance or the size of a cluster enclosed by a box is measured by the number of pixels in the image that is within the range of this cluster. Only the cluster with size larger than a prespecified threshold, N_{min}, is declared as a valid class. Thus, the number of valid classes and their mean vectors can be easily determined. This cluster information is used in the fine segmentation stage.

- Step 4 — to label the image and to get the coarse segmentation
  If the X1, X2 and X3 values of a pixel at site s, $X1_s$ $X2_s$ and $X3_s$, are inside the range of a cluster in the colour space, that is

$$U1_{X1}.low \leq X1_s \leq U1_{X1}.up$$

$$U2_{X2}.low \leq X2_s \leq U2_{X2}.up$$

$$U3_{X3}.low \leq X3_s \leq U3_{X3}.up$$

then this pixel is labeled with the class number. If a pixel is not inside the range of any cluster, then it is assigned the label of the cluster which is closest in the Euclidean distance between the pixel and the centre of the cluster. The segmented image is obtained by assigning the mean colour vector of the cluster according to the label of the pixel.

It is noteworthy that in the proposed approach, the user has the flexibility to control the coarseness of the segmentation by selecting the maximum scale (σ_{max}) and the minimum number of pixels (N_{min}) in a cluster. The default scale for σ_{max} is set at 30 and the default value for N_{min} is set at 100. Actually, this flexibility is very important, because in some applications, it is required that only a small number of clusters is needed, and these clusters are the most prominent, major clusters in the image. But in other applications, finer details may be required. Therefore, more clusters are needed. In SSF, if σ_{max} is increased, the number of active nodes is reduced, and the range of these active nodes are increased. When σ_{max} is decreased, the offsprings of the former active nodes may become active, and a new set of clusters can be obtained. The ranges of the clusters are typically reduced, and the number of clusters is increased.

4.2 Using the Multiresolution Scheme to Refine the Segmentation

After using the SSF, the cluster information about the image is known, and the image is segmented coarsely. The goal of MCIS is to make use of the information provided by SSF to refine the segmentation by using the neighbourhood-based approach. In the MCIS algorithm, the image is stored in a quadtree structure with the root node representing the entire image and the leaf nodes representing the individual pixels in the image [26]. Level x has a finer resolution than level $x-1$. MCIS uses simulated annealing to control the splitting and merging of blocks. The energy of a configuration in the system is a function of the quadtree structure. Note that the structure of the quadtree is different for each new configuration due to the splitting and merging of nodes.

4.2.1 Neighborhood system

Definition 1:

Let $S = \{s_1, s_2, ..., s_N\}$ be a set of sites. $g = \{G_s : s \in S\}$ is called a neighbourhood system for S if and only if

- $s \notin G_s$, and
- $\forall s, r \in S, s \in G_r \Leftrightarrow r \in G_s$.

The elements of S are referred to as *pixels* (sites) [1]. The pixels $r \in G_s$ are called the *neighbours of* s, and the pair (S, g) is called a *graph*. A set $C \subset S$ is called a *clique* if every two distinct sites in C are neighbours.

One commonly used family of neighbourhood systems over S is

$$g = \{G_{(i,j)} : (i,j) \in S\}$$

$$G_{(i,j)} = \{(k,l) : 0 < (k-i)^2 + (l-j)^2 \leq c, c > 0\}$$

Figure 4.6 shows some of the neighbourhood types with c =1, 2, and 4. In MCIS, $c = 1$, which corresponds to the 4-connectedness neighbourhood system.

A quadtree is a data structure which can represent a two-dimensional image of size $2^N \times 2^N$ hierarchically. The whole image is represented as the root of the quadtree.

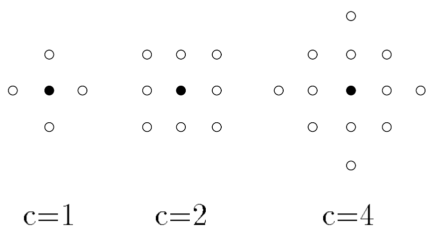


Figure 4.6: Neighborhood types according to size.

If the region that the root node represents is not homogeneous, it is divided into quadrants, each of which is called a *child.* The quadrants are recursively subdivided into subquadrants until each subquadrant represents a homogeneous region. A node that has no child nodes is called a *leaf node.* A node is also referred to as a *block.*

Definition 2:

Let b_1, b_2 be two nodes in a quadtree, η_1 is a neighbourhood system of b_1. Then b_1 and b_2 are neighbours if and only if $\eta_1 \bigcap b_2 \neq \phi$.

Since the 4-connectedness neighbourhood system is used in MCIS, only the nodes that are adjacent in the horizontal or vertical direction are said to be neighbours.

4.2.2 Markov Random Field and Gibbs Distributions

Let $X = X_s, s \in S$ denote a family of random variables indexed by s. In the context of MCIS, $\{X_s\}$ can be considered as a set of nodes in the quadtree. Let Ω be the set of all possible *configurations*:

$$\Omega = \{\omega = (x_{s1}, x_{s2}, ..., x_{sM})\}$$

where x_{si} is a node in the quadtree. M is the total number of nodes in the current quadtree. The event $X_{s1} = x_{s1}, X_{s2} = x_{s2}, ..., X_{sM} = x_{xM}$ is abbreviated as $X = \Omega$.

Definition 3: Markov Random Field in the quadtree structure

X is an Markov Random Field (MRF) with respect to the quadtree neighbourhood system η if

$$P(X = \omega) > 0 \qquad \text{for all } \omega \in \Omega$$

$$P(X_s = x_s | X_r = x_r, r \neq s) = P(X_s = x_s | X_r = x_r, r \in (\eta_s \cup Ch(X_s)) \tag{4.5}$$

for every $s \in S$ and $(x_{s1}, x_{s2}, ..., s_{xM}) \in \Omega$. $Ch(X_s)$ denotes the child nodes of X_s. Equation (4.5) represents the *local characteristics* of the MRF. In most neighbourhood system, the value of a site (normally a colour, or grey scale) depends only on its neighbours. The difference in a quadtree structure is that the value of a node (the size and position) depends not only on its neighbours, but also on its child nodes.

Definition 4: Gibbs Distribution

A Gibbs distribution relative to (S, η) is a probability measure π on configuration Ω with the following representations:

$$\pi(\omega) = \frac{1}{Z} e^{-\frac{U(\omega)}{T}} \tag{4.6}$$

where Z is a normalizing constant, and T the *temperature*. U is called the *energy function*.

Theorem[11]: Let η be a neighbourhood system. Then X is an MRF with respect to η if and only if $\pi(\omega) = P(X = \omega)$ is a Gibbs distribution with respect to η.

$$P(X_s = x_s | X_r = x_r, r \neq s) = \frac{1}{Z_s} e^{-\frac{U(\omega)}{T}} \tag{4.7}$$

This theorem establishes the equivalence between the MRF and Gibbs distribution.

In image segmentation, the objective is to maximize the *a posteriori* probability $P(X_s = x_s)$, that is, to minimize the energy function $U(\omega)$. Simulated annealing can be used to optimize this function.

4.2.3 Simulated annealing

Simulated annealing is a technique used for large scale optimization problems [26]. Typically, these problems involve combinatorial minimization of an objective function with a large search space. The search space is so large that it is combinatorially impossible to exhaust all possible configurations. Simulated annealing uses a much smaller search space to minimize the objective function. Annealing is a process by which crystal is formed when liquid is cooled. But if the liquid cools down too quickly, the molecules do not have time to settle in the lowest energy state before they lose their mobility. Therefore, the system ends up in a polycrystaline state with a high energy. So the essence of simulated annealing is to cool the system slowly so that it can settle into a state of low energy. It is assumed that the system would change its state with probability

$$p = \begin{cases} 1 & \text{if } E_2 \leq E_1 \\ e^{-\frac{E_2 - E_1}{kT}} & \text{otherwise} \end{cases} \tag{4.8}$$

p is the probability of the system changing from the state with energy E_1 to the state with energy E_2, and T is called the *temperature*. If $E_2 \leq E_1$ the new configuration will always occur because p is equal to 1; otherwise, if the state change increases the energy in the system (i.e. $E_2 > E_1$) the new configuration will occur with probability p. As the temperature T decreases, the probability of going into a higher energy configuration decreases. When $T = 0$, the system can not make any transition to a state with higher energy. In this way, the system settles into a state with the lowest energy.

4.2.4 The Multiresolution scheme

The energy of the system with configuration x is represented as

$$U(x) = \sum_{C \in S_x} V_x(C) \tag{4.9}$$

where

x denotes the quadtree structure.

C is a leaf node in the quadtree.

S_x is the set of leaf nodes in x.

$V_x(C)$ is the energy of leaf node C, which is defined as

$$V_x(C) = -\delta(d)\sum_{i=1}^{4}\delta_N(c_i) + (1 - 2\delta(d))\lambda_1 dissim(C) + \lambda_2[1 - \delta(d)]\delta_P(C) \tag{4.10}$$

where $\delta(d) = \delta(d < 0.5)$, d is a uniformly distributed random number with value between 0 and 1. If $\delta(d) = 1$, the possibility of splitting C into quadrants will be considered; otherwise, the possibility of merging C and its siblings (the three nodes which have the same parent as C) will be considered.

If $\delta(d) = 1$, splitting is considered. Then

$$V_x(C) = V_x^{(s)}(C) = -\sum_{i=1}^{4}\delta_N(c_i) - \lambda_1 dissim(C) \tag{4.11}$$

where $V_x^{(s)}(C)$ represents the energy after the splitting process, and λ_1 a weight constant. Otherwise,

$$V_x(C) = V_x^{(m)}(C) = \lambda_1 dissim(C) + \lambda_2\delta_P(C) \tag{4.12}$$

where $V_x^{(m)}(C)$ represents the energy after the merging process. $\lambda_2 \to \infty$. It inhibits the merging of nodes that have children themselves.

$c_i (i = 1..4)$ denotes the four child nodes of C. $\delta_N(c_i)$ is defined as

$$\delta_N(c_i) = \begin{cases} 1 & \text{if } \exists r \in \eta_{c_i}, r \text{ is similar to } c_i,\ \mathcal{P}(r) \neq \mathcal{P}(c_i), \mathcal{S}(r) \geq \mathcal{S}(c_i) \\ 0 & \text{otherwise} \end{cases}$$

where $\mathcal{P}(c)$ denotes the parent of node c, $\mathcal{S}(r)$ the size of the node r. If the children of node C have similar neighbours which are at least as large as them, then the splitting of node C is encouraged. This condition is used to prevent the splitting from forming too many small isolated regions. It is expected that the algorithm will be more noise resistant.

$dissim(C)$ is defined as

$$dissim(C) = \begin{cases} d_{node}(C) & \delta(d) = 1 \\ d_{node}(P), P = \mathcal{P}(C), & otherwise \end{cases}$$

$d_{node}(C)$ and $d_{node}(P)$ are dissimilarity measures of nodes C and P, respectively. It will be defined later.

$\delta_P(C)$ determines if there is a child of C which is also a parent itself.

$$\delta_P(C) = \begin{cases} 1 & \text{if any child } C_i \text{ of P is a parent node} \\ 0 & \text{otherwise} \end{cases}$$

and P is the parent of C.

If $V_x^{(s)}(C)$ is small, it encourages splitting. That is to say, if a node's children are very different, and after splitting, each of the child nodes is similar to one of its neighbours, then it is reasonable that this node should be split. If $V_x^{(m)}(C)$ is small, which means this node is similar to its siblings, and neither this node and its siblings have children, then merging would probably happen.

The following probabilities are used to determine if splitting or merging will occur.

$$p = \begin{cases} 1 & \text{if } V \leq V_t \\ e^{-\frac{V-V_t}{kT}} & \text{otherwise} \end{cases} \tag{4.13}$$

where V_t is a predefined constant, which can be different for splitting and merging. If $V \leq V_t$, splitting or merging (depend on $\delta(d)$) will happen ($p = 1$); otherwise, it will happen with probability p. With the decrease of temperate T, the chance of random splitting or merging will be lowered accordingly. The threshold V_t may be different for splitting and merging.

4.3 The Dissimilarity Measures

In MCIS, the split-and-merge scheme is used to perform segmentation. In order to determine if a split or a merge could take place, the dissimilarities between nodes must be measured. The dissimilarity inside one block and the dissimilarity between two blocks whose size may be different are to be measured.

The dissimilarity inside one block is measured by the dissimilarity between the children of the block. A node is said to be different inside if there exists a pair of its child nodes that have large distance:

$$d_{node}(C) = \delta[\exists(j,k) d(C_j, C_k) > d_t)] \tag{4.14}$$

where C_j and C_k are child nodes of C. $d(C_k, C_j)$ is a distance measure C_j and C_k. d_t is a predefined threshold. In MCIS, it is also required to measure the dissimilarity between two neighbouring nodes (not *siblings*). The size of the two nodes may be different.

In MCIS, three kinds of dissimilarity measures are to be examined:

1. The Kolmogorov-Smirnov distance

The Kolmogorov-Smirnov (KS) statistic is applied to measure the difference between two probability distributions. In this case, the probability distribution is the normalized cumulative distribution of the histogram of each pixel block [26]. The KS statistic is defined to be the maximum distance between the two cumulative distributions of the data sets P_1 and P_2

$$d_{KS}(P_1, P_2) = \max_{-\infty<t<\infty} \mid F_1(t) - F_2(t) \mid \tag{4.15}$$

where

$P = (v_1, v_2, \cdots, v_n)$ and

$F(t) = \frac{1}{n}\sharp\{k : v_k \leq t\}$, the cumulative distribution function. In colour images, the three R, G and B distances should be combined into one uniform measure. The distance d between two colour blocks is defined as follows:

$$d(P_1, P_2) = \sqrt{d_r(P_1, P_2)^2 + d_g(P_1, P_2)^2 + d_b(P_1, P_2)^2}$$

where d_r, d_g, d_b are defined as in equation (4.15).

One problem with the KS-distance is that when the size of the nodes is too small, e.g. 1×1, this statistical method can no long be applied to measure the dissimilarity of the nodes. For example, if the size of the nodes is 1×1, then the KS-distance between two nodes, i.e. pixels, which have very similar grey scale values (e.g. 200 and 201) is 1, which is the upper bound of the KS-distance. In this case, other dissimilarity measures must be used instead.

2. The Euclidean distance between the mean colour vectors

   The Euclidean distance between the mean colour vectors of the two nodes is:

$$d(C_k, C_j) = \sqrt{(\bar{R}_k - \bar{R}_j)^2 + (\bar{G}_k - \bar{G}_j)^2 + (\bar{B}_k - \bar{B}_j)^2}$$

where $\bar{R}_k, \bar{R}_j, \bar{G}_k, \bar{G}_j, \bar{B}_k, \bar{B}_j$ are the mean values of the nodes C_k and C_j in the R, G and B space, respectively.

3. The mean HV distance

   Using this dissimilarity measure, the mean vectors in the HSV space of the nodes to be compared are calculated. The two nodes are said to be dissimilar if both the H (hue) distance and the V (brightness) distance are greater than some preset thresholds.

Besides, the Bhattacharyya distance [8] was also investigated in MCIS. Due to the singularity problem of the Bhattacharyya distance when the variance is zero, this distance is not suitable in this approach.

4.4 Colour spaces

In the above discussions, the RGB colour space is used in the description of the algorithm. But actually, a set of colour spaces will be used in the SSF and MCIS algorithms. The objective is to find a colour space that has greater discriminatory capacity. That is to say, in such a colour space, the distance between different objects in the image should be large, while it is less sensitive to the geometry (e.g., the shading) of the object. The colour spaces be used and tested and their transformation from the RGB space are:

- RGB — original tristimuli red, green, and blue

- HSV — hue, saturation and value [4]

$$
\begin{aligned}
V &= Max \\
S &= \begin{cases} 0 & \text{if } Max = Min \\ \frac{Max-Min}{Max} & \text{otherwise} \end{cases} \\
H &= \begin{cases} undefined & \text{if } Max = Min \\ 120 \times (G-B)/(R+G-2B)] & \text{if } Min = B \\ 120 \times [1+(B-R)/(B+G-2R)] & \text{if } Min = R \\ 120 \times [2+(R-G)/(R+B-2G)] & \text{if } Min = G \end{cases}
\end{aligned}
$$

  where Max and Min are the maximum and the minimum value of the R, G and B components, respectively.

- $I1, I2', I3'$ — Ohta's colour space [28]

$$
\begin{aligned}
I1 &= (R+G+B)/3 \\
I2' &= R-B \\
I3' &= (2 \times G - R - B)/2
\end{aligned}
$$

- XYZ — C.I.E. X-Y-Z colour space [29]

$$
\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.607 & 0.174 & 0.201 \\ 0.299 & 0.587 & 0.114 \\ 0.000 & 0.066 & 1.117 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}
$$

- Lab — C.I.E. uniform colour coordinate space [28]

$$
\begin{aligned}
L &= 25[(100Y/Y_0)^{1/3}] - 16 \\
a &= 500[(X/X_0)^{1/3} - (Y/Y_0)^{1/3}]
\end{aligned}
$$

$$b = 200[(Y/Y_0)^{1/3} - (Z/Z_0)^{1/3}]$$

Where X_0, Y_0 and Z_0 are the values of X, Y and Z for the reference white, i.e., the values of X, Y and Z when the R, G and B values are all 255 in this thesis.

- YIQ — colour TV transmission primaries [29]

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.275 & -0.321 \\ 0.211 & -0.523 & 0.312 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Every transformed colour component is normalized to the range of [0,255] because each component is represented by an 8-bit unsigned character.

Chapter 5

Experiments

The goal of the experiments is to detect how the parameters settings and colour spaces affect the results of the segmentation. The parameters mainly consists of the thresholds for dissimilarity measures, the cooling schedule for simulated annealing, and the option for connecting neighbouring nodes. The selection of parameters for the SSF has been discussed in the section 4.1(?), and is not included in this chapter. The parameters to be investigated in the multiresolution part of MCIS are shown in Figure 5.1. In order to evaluate the performance of different parameters and different colour spaces quantitatively, a novel evaluation function is introduced in this chapter.

In the experiments, both real images and synthesized images are used to evaluate the performance of the proposed algorithm (Figure 5.2). The size of the images is 128×128. The real images, which are quite noisy, were scanned from magazines or fliers. In this thesis, results on using two real images, namely "girl" and "house", are described. Two synthesized images are used: one is "crossboard" and the other is "cylinder". The image "crossboard" is selected to avoid favouring the quadtree data structure used in MCIS, which has a bias for horizontally and vertically oriented edges. In this synthesized image, the distribution of each of the R, G and B colour components is a combination of two Gaussian distributions, one with a mean value

<table>
<tr><td rowspan="7">The Multi-resolution Scheme</td><td rowspan="6">Control parameters</td><td rowspan="4">Dissimilarity measures</td><td>KS-T</td></tr>
<tr><td>EU-T</td></tr>
<tr><td>SMALL</td></tr>
<tr><td>H-T, V-T</td></tr>
<tr><td>Simulated annealing</td><td>α</td></tr>
<tr><td>Connecting neighbouring nodes</td><td>CONNECT</td></tr>
<tr><td colspan="2">Colour spaces</td><td>RGB, HSV, Lab
I1-I2′-I3′, XYZ
YIQ</td></tr>
</table>

Figure 5.1: Parameters to be examined in the multiresolution scheme of MCIS.

of 70 and the other of 185, and both with a standard deviation of 50. The other synthesized image, “cylinder” is rendered with some specular reflection (highlight) using the ray tracing techniques. The synthesized images without noise and contours of regions are shown in Figure 5.3. The coarse segmentations of the images using SSF are shown in Figure 5.4.

5.1 Evaluation measures

Most image segmentation results are evaluated visually and qualitatively. The main problem is due to the lack of uniform criterion to evaluate the results. Also, different people with different experience may have different judgments. In order to evaluate experimental results scientifically and statistically, quantitative methods must be developed and used.

Miss-labeling rate is one of the measures that can be used to evaluate the accuracy

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

Figure 5.2: Colour images for segmentation, (a) crossboard; (b) cylinder; (c) girl; (d) house.

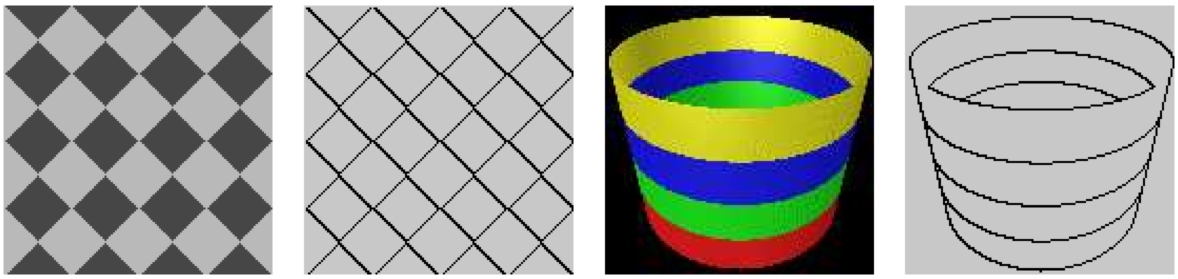


Figure 5.3: Perfect images of "crossboard" and "cylinder" and their contours.

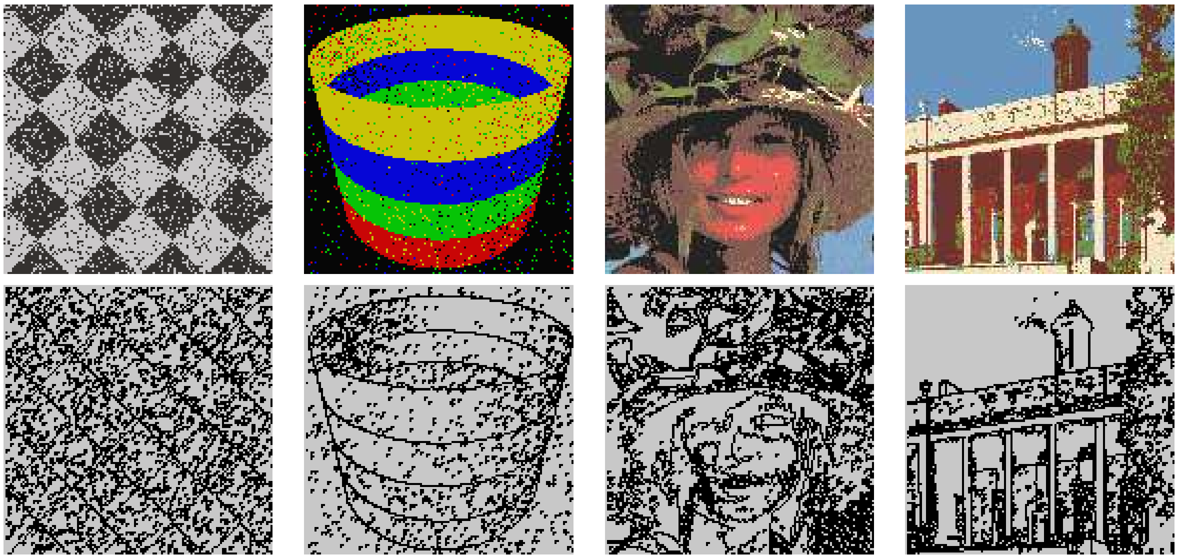


Figure 5.4: Coarse segmentations using SSF and their contours.

of segmentation. It is defined as the ratio of the number of pixels that are assigned the wrong cluster labels to the total number of pixels in the image. For synthesized images, the miss-labeling rate can be computed easily. One problem with using miss-labeling rate is that the result with a lower miss-labeling rate may not correspond to a better segmentation. As an example, in Figure 5.5, segmentation result (a) has a lower miss labeling rate than that of (b), but actually (b) is a better segmentation visually since there are less small holes. The reason for this inconsistency is because the miss-labeling rate is a global measure, while the evaluation of segmentation should account for some local information. In fact, the miss-labeling rate can be separated into two parts, one for pixels on both sides of the boundaries, and the other for pixels inside the boundaries. The miss-labeling rates for boundary pixels are 1.92% and 3.51% for results in Figure 5.5 (a) and (b) respectively, and the miss-labeling rates of interior pixels are 1.24% and 0.85%, respectively, for the same figure. When the noise is suppressed, the boundaries between regions become jagged mainly because of the quadtree structure. Hence, the miss-labeling rate of boundary pixels is sensitive

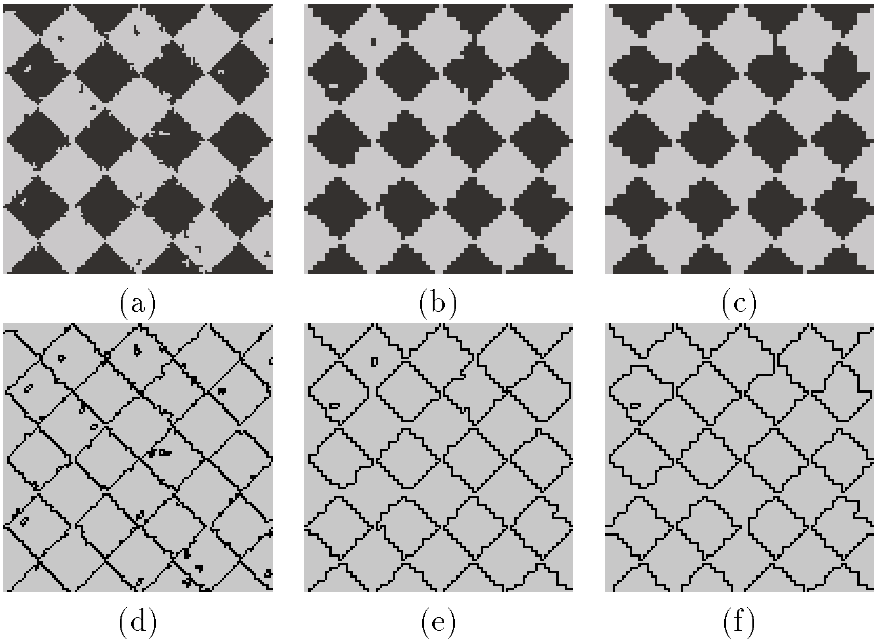


Figure 5.5: Some segmentation results of crossboard, (a) segmented image with miss-labeling rate 3.16%; (b) segmented image with miss-labeling rate 4.36%; (c) segmented image with miss-labeling rate 5.38%; (d) contour of (a); (e) contour of (b); (f) contour of (c);

to the data structure used by the segmentation procedure. The miss-labeling rate of interior pixels is more consistent with human intuition and is thus a better measure than the total miss-labeling rate.

Another problem with the miss-labeling rate is that it is applicable only to synthesized images, in which case the true labels of the image are known. But for real images, it is hard, if not impossible, to know the true label of each pixel. Some attempts have been made to compute the miss labeling rate based on manually segmented results [10]. Unfortunately, such an approach can be applied to simple scenes only. In addition, it is difficult to be convinced that the manually segmented result is the "correct" one.

In order to evaluate segmentation results of real images as well as synthesized ones, and to evaluate results both locally and globally, a novel evaluation function is proposed and developed. The evaluation function F is defined as

$$F(I) = \sqrt{R} \times \sum_{i=1}^{R} \frac{e_i^2}{\sqrt{A_i}} \tag{5.1}$$

where I is the image to be segmented, R the number of regions in the segmented image, A_i the area, or the number of pixels of the ith region, and e_i the colour error of region i. e_i is defined as:

$$e_i = \sum_{j=1}^{A_i} \sqrt{(C1^j_{X1} - C2^j_{X1})^2 + (C1^j_{X2} - C2^j_{X2})^2 + (C1^j_{X3} - C2^j_{X3})^2}$$

where $X1$, $X2$ and $X3$ denote each of the components in the colour space X1-X2-X3, e.g., R, G and B in the RGB space, $C1^j$ is the colour of pixel j of the original image I in the space X1-X2-X3, and $C2^j$ is the assigned colour of the same pixel in the segmented image. The smaller the value of F, the better is the segmentation result. The term $\sqrt{R}$ is a global measure which gives penalty to the segmentation which forms too many regions. The term $\frac{e_i^2}{\sqrt{A_i}}$ is a local measure which penalizes small regions or regions with a large colour error. e_i indicates whether a region is assigned an appropriate feature (colour) or not. A large value of e_i means that the feature of the region is not well captured during the SSF process. In practice, F is normalized by the size of the image and is scaled down by the factor 1/1000.

There is always a trade-off between preserving details and suppressing noise. If too many details are preserved, the segmented image may appear noisy. On the other hand, in order to suppress noise, some details may be lost. A good segmentation should make a compromise between these two requirements. This compromise is reflected in the evaluation measure F. If there are too many details in the segmented image, the colour error of each region may be smaller. But since many small regions are formed and the number of regions is large, the value of F is large which indicates that the segmentation result is not a good one. On the other hand, if very few details are retained and therefore noise is suppressed, then the number of regions is reduced, but the colour error of each region would be large. In this case, F would be large too. A good segmentation should be able to preserve appropriate details while suppress as much noise as possible. Figure 5.5 shows an example of such a compromise. Figure 5.5 (a) keeps most of the details of the original image, i.e., the shape of the squares is well retained, but the result is very noisy, i.e., there are many small holes in the regions. (b) loses some details around the boundary of the squares, but most noise is suppressed in this result. Although the miss-labeling rate of this segmentation is higher than (a), it is still more preferable. In (c), noise is further suppressed (notice that one more small region due to noise is removed compared with (b)), but too many details are lost, especially in the upper right part of the segmented image. By overall consideration, (b) is the best segmentation of the three. The values of F for these three segmentations are 378.6, 330.5 and 335.9, respectively. This means (b) is much better than (a), and slightly better than (c). Hence, the proposed evaluation measure is consistent with the segmentation criteria and also with what is observed visually.

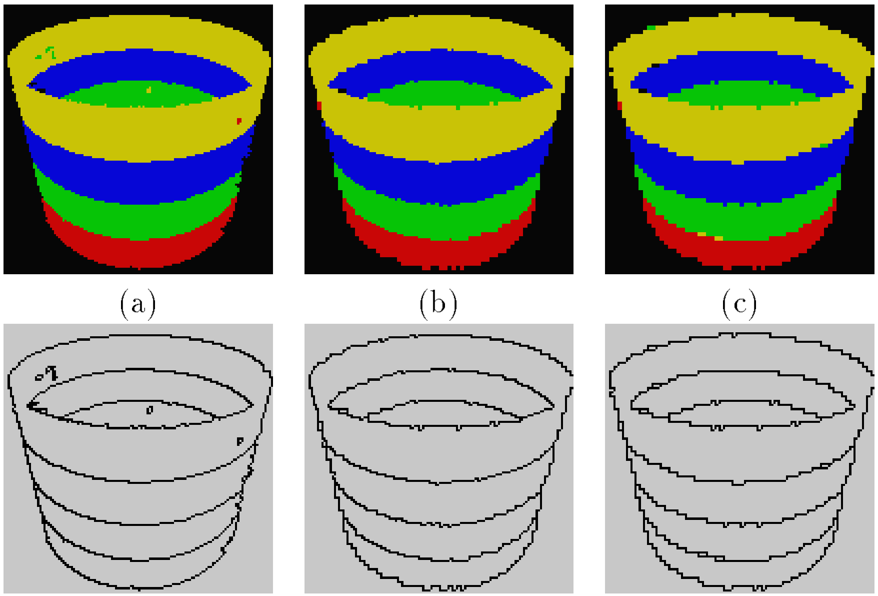
In Section 3.1, it was mentioned that a good segmentation result should satisfy four criteria: 1) regions should be homogeneous, 2) regions should be simple and without many holes inside, 3) adjacent regions should be significantly different, and 4) boundaries should be smoothed and accurate. The evaluation function, F, is a simplified model of these criteria. In fact, F incorporates these requirements either directly or indirectly, except criterion 4), the boundaries conditions. In this thesis, the jagged boundaries is assumed to be improved by some smoothing techniques, which is outside the scope of the thesis. F penalizes those segmentations which contain many holes. If a segmentation contains many holes, there must be many small regions, i.e., many small A_i, and the number of regions R must be large, and therefore, F is large. This conforms with criterion 2). Although criteria 1) and 3) are not handled directly in the evaluation function, they have some indirect relationship with it. If a region i in the segmented image is not homogeneous, the colour error e_i must be large because the variance of the colour in this region is large. For criterion 3), if the adjacent regions are not significantly different, this means such regions should be merged instead of forming separate regions, and there are more small regions A_i and larger number of regions R in the segmentation than it should be, which is also penalized in F. The reason that criteria 1) and 3) can not be incorporated directly in the proposed evaluation function is because these criteria are subjective to some extend. For example, how similar should the pixels in a region be in order to say that this region is "homogeneous"? And what is the meaning of "significantly different" adjacent regions? Some thresholds should be set to calculate the similarities, and it is very difficult for one to justify such thresholds. It will be shown experimentally that the proposed evaluation function F is a good measure for both synthesized and real images.

Figures 5.6 to 5.8 show some segmentation results for other three images. In each of these figures, (a) preserves more details but is too noisy, (b) is less noisy but has less details, and (c) has even less details than (b). Tables 5.1 to 5.4 show the values of F for these segmentation results. For synthesized images, the miss-labeling rate is used as one of the measures. For the real images, only the new measure F is used. In Tables 5.1 and 5.2, the first measure is the total miss-labeling rate, the second is the miss-labeling rate of the boundary pixels, and the third measure is the miss-labeling rate of interior pixels. "Average colour error" is the mean Euclidean colour distance of the whole image, i.e., the sum of Euclidean colour distance of each pixel between the original image and the segmented image divided by the number of pixels in the image. "number of regions" is the total number of regions in the segmented image. The four-connectedness system is used in counting the number of regions. For each set of segmentation results shown in Figures 5.5 to 5.8, (a) contains more details and the average colour error is smaller, but because of the small regions and the larger number of regions, the value of F is large. (c) is the opposite, many details are lost and therefore the colour error is large. Although there are less number of small regions and the total number of regions is less, the function F is still large. (b) has smaller colour error than (c) and has less regions than (a). According to the segmentation evaluation criteria, (b) is a good trade-off between suppressing noise and preserving details. This is also reflected in the proposed evaluation measure F. In each set of results, (b) always has the smallest F, which means that it is the best. It is noteworthy that normally, when a segmentation with less details is obtained, there is less small regions. But sometimes it is not the case. For example, Figure 5.6 (c) has a larger colour error than (b), while it also has more small regions than (b). The reason is that if the nodes on the boundary of some regions are merged, the new merged node has different features, i.e., mean colour vectors, than any of the adjacent regions. Therefore, a small miss-labeled region is formed.

The proposed evaluation function F is used in the later part of the experiments to quantitatively examine the performance of the MCIS algorithm using different parameters, e.g., dissimilarity measures, and colour systems. In order to evaluate the segmentation results in different colour spaces, all of the segmented images in the spaces other than RGB are transformed back to the RGB space for comparison because the colour errors in different colour spaces have different scales and are not comparable.

5.2 The control parameters

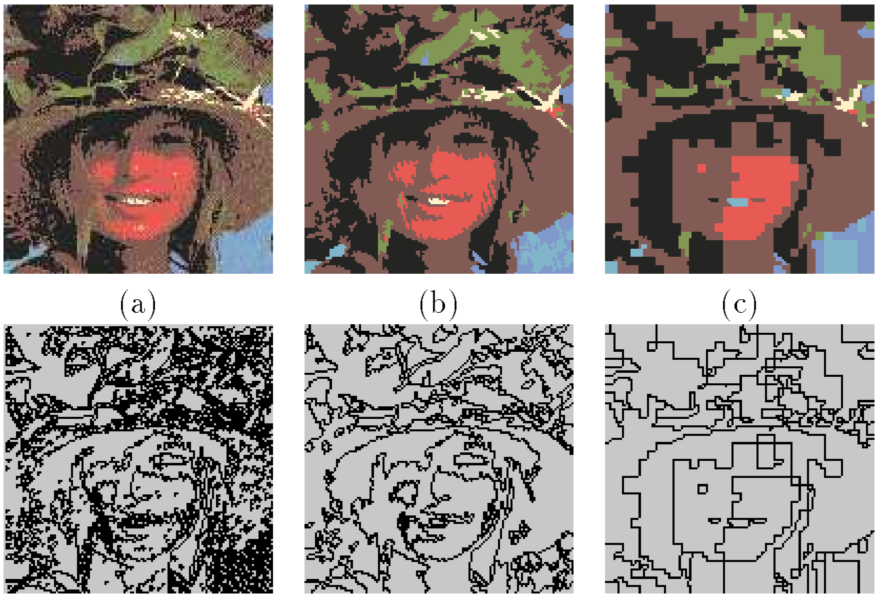
In the following experiments, the effect of dissimilarity measures, the effect of simulated annealing and the effect of connecting neighbouring nodes on the segmentation results are examined. In this section, the segmentation process is performed in the RGB space. The parameters evaluated experimentally are:



(a) (b) (c)

(d) (e) (f)

Figure 5.6: Segmentation results of "cylinder".



(a) (b) (c)

(d) (e) (f)

Figure 5.7: Segmentation results of "girl".

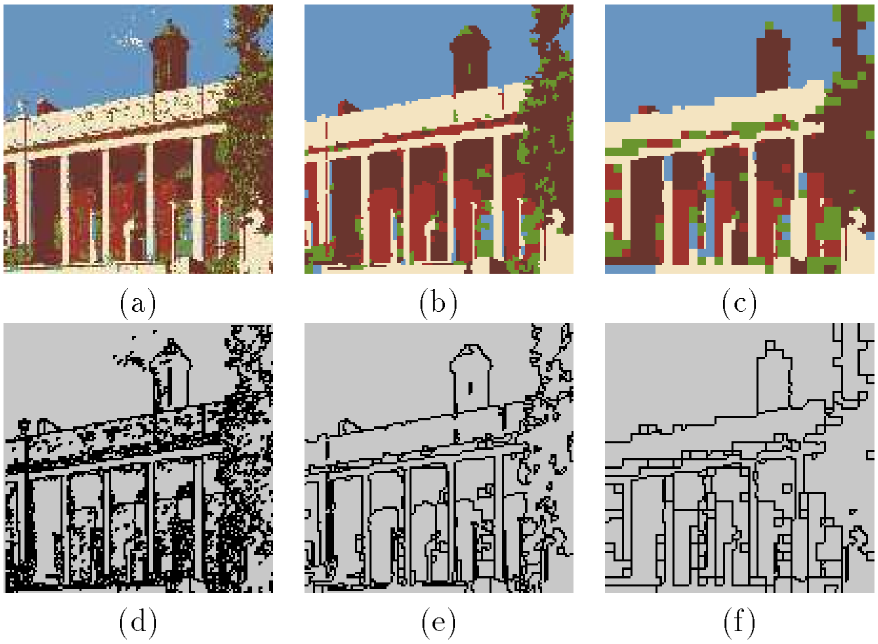


Figure 5.8: Segmentation results of "house".

Measure	result (a)	result (b)	result (c)
Miss-labeling rate (%)	3.16	4.36	5.38
Miss-labeling rate (boundary)	1.92	3.51	3.79
Miss-labeling rate (interior)	1.24	0.85	1.59
Average colour error	104.8	109.1	110.2
Number of regions	35	22	21
F	378.6	330.5	335.9

Table 5.1: Evaluation measures for image "crossboard" in figure

Measure	result (a)	result (b)	result (c)
Miss-labeling rate (%)	0.80	2.15	3.57
Miss-labeling rate (boundary)	0.53	2.07	3.45
Miss-labeling rate (interior)	0.27	0.09	0.12
Average colour error	67.5	69.9	72.6
Number of regions	14	10	14
F	96.6	87.1	110.6

Table 5.2: Evaluation measures for image “cylinder” in figure

Measure	result (a)	result (b)	result (c)
Average colour error	31.8	35.5	42.1
Number of regions	2032	145	89
F	103.0	48.7	53.8

Table 5.3: Evaluation measures for image “girl” in figure

Measure	result (a)	result (b)	result (c)
Average colour error	40.4	44.7	55.7
Number of regions	1405	143	102
F	129.9	79.5	104.8

Table 5.4: Evaluation measures for image “house” in figure

1. Dissimilarity measures:

   - $KS - T$ — the threshold for the KS-distance, the range is from 0 to $\sqrt{3}$;
   - $EU - T$ — the threshold for the square of the Euclidean distance between the mean vectors of the two nodes. The range is from 0 to 195075 (3×255^2);
   - $H - T$, $V - T$ — the threshold for the hue and brightness distance respectively. The range of both parameters is from 0 to 255;
   - $SMALL$ — only the nodes with size larger than this parameter use the KS-distance as the dissimilarity measure.

2. Parameters defined in the simulated annealing scheme:

   - α - the cooling schedule. It determines the rate at which the temperature is lowered at each iteration of the annealing process. The larger the α, the slower is the cooling schedule. The default value of α is set to be zero. This means that instant cooling is adopted instead of the slower cooling schedule in simulated annealing. The reason for this is discussed later.
   - Ω - the number of configurations checked per iteration at each temperature value. The default value, 100, is used in the experiments. In case that the instant cooling schedule is used, this parameter does not have any significant effect on the performance because the temperature is always the same (zero) for each iteration. If Ω is small, the number of iterations is increased before the relaxation process converges; otherwise, if Ω is large, the total number of iterations is decreased. But normally, the total configuration changes, i.e., splitting and merging, is similar.

3. The parameter to control the connecting of neighbouring quadtree nodes:

   - $CONNECT$ — if 1, the similar neighbouring nodes are connected into one region before they are labeled.

4. Other parameters in the MCIS (the default values of these parameter are used)

    - λ_1 - the relative weight between the two terms in the energy function for splitting (equation 4.11). It determines the relative importance between the dissimilarity of a node and the neighbouring information when splitting is being considered. The default value is 4.

    - V_t - the energy threshold defined in equation 4.13. It determines how low the energy after splitting and merging should be in order to perform a definite splitting or merging, i.e., the probability is considered to be one. In the case of instant cooling, if the energy is higher than V_t, no configuration change is allowed. Otherwise, if the simulated annealing is adopted, the probability of taking a random change, i.e., random spliting or merging, is calculated. This parameter is set to be 0 for merging, i.e., only if four sibling nodes are all similar will they definitely be merged into a large node. Otherwise the probability of merging is computed in case of simulated annealing. For splitting, if the size of a node is larger than 2×2, V_t is set to be -5. That is to say, if a node is not homogeneous inside, and after splitting, at least one of the child nodes has a similar neighbour which is at least as large as this child node, then splitting will definitely happen. If the size of a node is as small as 2×2, the condition for splitting is more restrictive, and hence V_t is set to be -8. It means that a small node is split if after splitting, each of its child nodes has a similar neighbour, and the child nodes are not similar to each other.

5.2.1 Effect of the dissimilarity measures

Three kinds of dissimilarity measures are evaluated in the experiments, namely:

- The KS-distance;

- The Euclidean distance between the mean colour vectors of the two nodes;
- The distance of H (hue) and V (brightness value) between the mean colour vectors of the two nodes;

The KS-distance is used to measure the difference between two probability distributions. Figure 5.9 shows the effect of using KS-distance only to compare the dissimilarity between two nodes in all resolution levels, i.e., for nodes with any size. The range of $KS-T$ is [0.0, 1.732]. From this figure, it is clear that $KS-T$ has no effect on almost all the results, except that when $KS-T > 1.4$ the result of "cylinder" become a little worse. The reason is that the coarse segmentation of using SSF is very noisy, and therefore the quadtree is initialized mainly at the pixel level, i.e., the size of most nodes is 1×1. As mentioned in Section 4.3(?), the KS distance is not suitable for such small nodes. In order to solve this problem, the Euclidean distance between the mean colour vectors of the nodes are used when the size of the node is small.

Figures 5.10 and 5.11 show the values of the F when the mean Euclidean distance is used for nodes with size 1×1 and the KS-distance is used otherwise. Figure 5.10 shows the curves of F versus the Euclidean distance threshold, $EU-T$, when the KS-distance threshold $KS-T$ has different values. Notice that for both of the synthesized images which has a high level of Gaussian noise, it is preferable to select a larger $EU-T$. For "cylinder", the value of F has the lowest value when $EU-T$ is 80000. For "crossboard", the larger the $EU-T$, the smaller is the evaluation function. This means that when all the pixels are merged into larger nodes, i.e., 2×2 or larger, the noise is suppressed and a better segmentation is obtained. In the image "girl", small $EU-T$ produces better results. There are some wigglings in the curve because $EU-T$ is a global parameter while the evaluation function accounts for the local performance. Sometimes when $EU-T$ is changed by a small amount, the evaluation function may change drastically. For the image "house", when $EU-T$

Figure 5.9: Effect of using KS-distance in all resolution level.

is very small or very large, the evaluation function is large. The optimal results are obtained when $EU-T$ is in the range from 6000 to 16000. When the thresholds for large nodes are different, the curves of F versus $EU-T$ are also different. By these curves, the best set of thresholds can be picked that minimize the evaluation function.

Figure 5.11 shows the effect of $KS-T$ when $EU-T$ is fixed. Two values of $EU-T$ are used in each image. These values are selected in such a way to show that the shape of the curves of F versus $KS-T$ can be different when $EU-T$ is different. When $EU-T$ is small, the curves are relatively plain. For images other than the image "house", even if the value of $KS-T$ is unreasonably large, i.e., larger than the upper bound of $KS-T$ ($\sqrt{3}$), the results are still good, or even better (for "cylinder"). The reason is that the coarse segmentations after using SSF are typically noisy. In MCIS, when the quadtree is initialized according to the coarse segmentation, there are many 1×1 nodes in the quadtree. If the distance threshold for small nodes, $EU-T$, is set to be small, those small nodes can not be merged into bigger nodes, and therefore the chance of using $KS-T$ to measure larger nodes is very small, and so $KS-T$ does not have much effect on the performance. On the other hand, if $EU-T$ is larger, the result would be worse if $KS-T$ is also very large, because most nodes are merged and therefore many details are lost. In the extreme case, all nodes are merged and the final segmentation is only one big region. Notice that the curve always remain flat when $KS-T \leq 0.7$. This means that most colour KS-distances between two nodes are greater than 0.7, and it is not necessary to search $KS-T$ in the range of [0, 0.7] in order to find the optimal value.

One problem that should be examined is how small the nodes are when the KS-distance fails. The parameter "SMALL" is used to denote that for nodes equal to or smaller than this parameter, the Euclidean distance between the mean vectors will be used instead of the KS-distance. For example, $SMALL = 2$ means that nodes

Figure 5.10: Effect of EU-T

Figure 5.11: Effect of KS-T

with size 1×1 and 2×2 will use Euclidean distance.

Figure 5.12 shows the effect of using KS-distance and the Euclidean distance between the mean RGB vectors. In each of the images, the effect of $SMALL$ =1, 2 and 4 are tested. The curves show that for noisy synthesized images, "crossboard" and "cylinder", and for the image with more details, "house", $SMALL = 1$ produces better results, and large $EU-T$ can make it easier for the 1×1 nodes (pixels) to merge into 2×2 nodes, so that noise can be suppressed. But if $SMALL = 2$ or larger, large $EU - T$ forces too many mergings and segmentation results are obtained with most details lost. For image with less details, e.g., "girl", $SMALL = 2$ produces the best results, but the results with $SMALL = 1$ are also not bad. Therefore, $SMALL = 1$ is a good scheme for most images. This means that for the nodes with size 1×1, it is better to use the Euclidean distance between the colour vectors, and for the nodes larger than 1×1, the KS-distance is good.

Another dissimilarity measure is the hue-brightness (HV) distance. There are two thresholds in this measure, one for the hue and the other for the brightness. The mean HSV vectors of the two nodes are calculated and the distance of H and V between the two nodes are used. If both the absolute difference values of H and V are less than the corresponding thresholds, the two nodes are considered to be similar, otherwise they are different. Figure 5.13 shows the effect of using the HV distance. In this set of experiments, $SMALL = 1$, and the KS-distance is still used for measuring the larger nodes. The thresholds for the KS-distance, $KS - T$, are selected to be the best from the curves shown in Figure 5.10. They are 1.0, 1.4, 1.4 and 1.0 for the images "crossboard", "cylinder", "girl" and "house" respectively. Each curve in Figure 5.13 is a function of F versus the threshold for V. A set of thresholds for H (20, 40, 60 and 100) is used in each image. Compared with Figure 5.10, it can be shown that the best result of using the RGB distance is better than the best results of using the HV distance in "crossboard" and "house", but worse in "cylinder" and

Figure 5.12: Effect of using KS-distance and mean Euclidean distance in different resolution level.

"girl". It is noteworthy that since the noise in "crossboard" and "cylinder" is added in the RGB space, it may not be appropriate to use the HV distance to measure the dissimilarity in these two images. Generally speaking, the results of using these two measures are comparable. But since it needs two thresholds to use the HV distance, it would be more difficult to find the optimal result than using the RGB Euclidean distance. Hence, the RGB distance is preferred. Figure 5.14 compares the effect of using the HV distance with $SMALL = 1$ and $SMALL = 2$. $SMALL = 1$ produces slightly better results than $SMALL = 2$.

In summary, the KS-distance is good for measuring nodes with size greater than 1×1, and the Euclidean distance between the mean R, G and B values of two nodes are preferable to the mean HV distance when the size is 1×1.

5.2.2 Effect of simulated annealing

Simulated annealing is a technique for large scale optimization. It is supposed to be able to avoid local optimum and to get to a state of global optimum. In the experiments, the effect of using simulated annealing in image segmentation is examined by changing the cooling schedule parameter, α. The range of α is from 0 to 0.99. If α is zero, which means the instant cooling schedule is used, there is no random change of configuration at all. The larger the α, the slower is the cooling, and the more random changes of configurations in each iteration. It was expected that a large α would produce better results because of the slow cooling. Experimentally, this is found not to be true. Figure 5.15 shows the curve of F versus α. α is sampled at 0, 0.1, 0.2, ..., 0.9 and 0.99. Surprisingly, the results when $\alpha = 0.99$ are always very bad. At points of α other than 0, the result is occasionally better than the result when α is 0. But the result with α equal to 0 is always good. Figure 5.16 shows the curves of the CPU time (in seconds) for completing the segmentation versus α. When α is greater than 0.9, the time consumed increases drastically. For the α that is less than 0.9, the

Figure 5.13: Effect of using HV distance

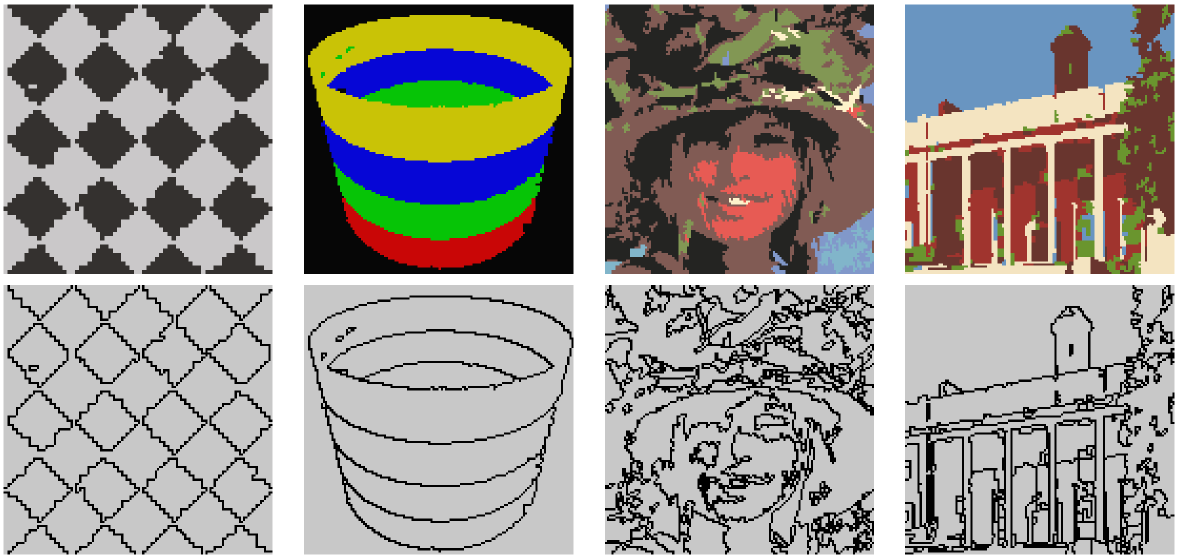
Figure 5.14: Effect of using HV distance with different 'SMALL' values.

time used does not vary significantly with α and is relatively constant. From this set of experiments, it seems that simulated annealing does not produce better results, and therefore the instant cooling scheme is applied in MCIS. Figure 5.17 shows the segmentation results when α =0.0 and 0.99.

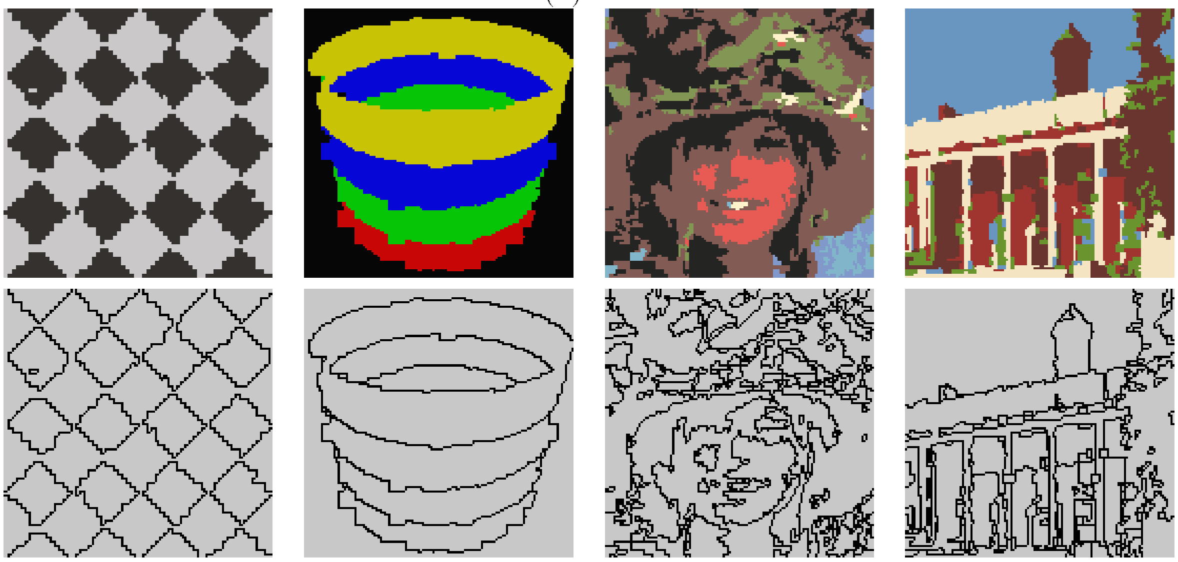
The reason that simulated annealing may not produce better results is that the function to be optimized, or the energy function to be minimized in MCIS, is not the function to be evaluated in the final segmentation. It is non-trivial to find such a function to be optimized so that the final result is the best when the function is optimized. Similar findings are also reported by different researchers [7] [20] [26]. In [7], it is reported that the energy function is not always minimized at the visually optimal results, and for a certain result, simulated annealing produces the smallest energy, but the segmentation is the worst visually. In the experiments presented by Huang et al. [20], segmentation results of using simulated annealing and ICM (Iterated Conditional Modes) [3] are very similar, but the time of using simulated annealing is much more than that of using ICM (4 hours versus 10 minutes). In the paper of Muzzolini et al. [26], some synthesized images and three real ultrasound images are used in the texture segmentation algorithm. For the synthesized images, the results using slower cooling comparison schedule have higher miss-labeling rates than the ones using very fast cooling schedule. For the three real ultrasound images, simulated annealing produces one better result, one worse result and one comparable result compared with the results using fast cooling. The time of using slow cooling is also longer than that of using fast cooling. It seems that there is no convincing evidence to demonstrate that simulated annealing can produce better results in image segmentation.

Figure 5.15: F versus α.

Figure 5.16: CPU time versus α.



(a)



(b)

Figure 5.17: Results comparison of using simulated annealing or not. (a) segmentation results of instant cooling ($\alpha = 0.0$); (b) segmentation results of slow cooling ($\alpha = 0.99$).

5.2.3 Effect of connecting neighbouring nodes

As was discussed, the quadtree structure is used in the MCIS algorithm, and the splitting and merging process is performed according to the similarity of the sibling nodes. When the relaxation process converges, each node is assigned a label, or a colour. The problem of this approach is that a node can not be merged with other nodes which are not its siblings even if they are similar, and sometimes the nodes are assigned the wrong labels because the information used for labeling is too local. In order to solve this problem, a parameter, called "CONNECT", is added into MCIS. If "CONNECT" is set to be 0, the above approach is applied. Otherwise, similar nodes are connected in the following way. After the relaxation process converges, each leaf node in the quadtree is scanned. The first leaf forms a region, and the feature of this region is its mean colour vector. For the successive leaf nodes, each node is compared with the region features of all of its neighbouring nodes. If the neighbour has been visited, and the feature of the node is similar to the feature of this neighbour, then the node being visited is added to the region to which its neighbour belongs, and the region feature is revised by the mean vector of all the nodes that is already in this region. Otherwise, the node being visited forms a new region. Finally, each region is assigned a label according to its feature. This new option can produce very good results for some images, especially those images with highlights (Figure 5.18). The problem with this method is that when a node which includes pixels from two regions is not split, it will sometimes become a transition node to connect these two regions because this node may have feature that is similar to both of these regions.

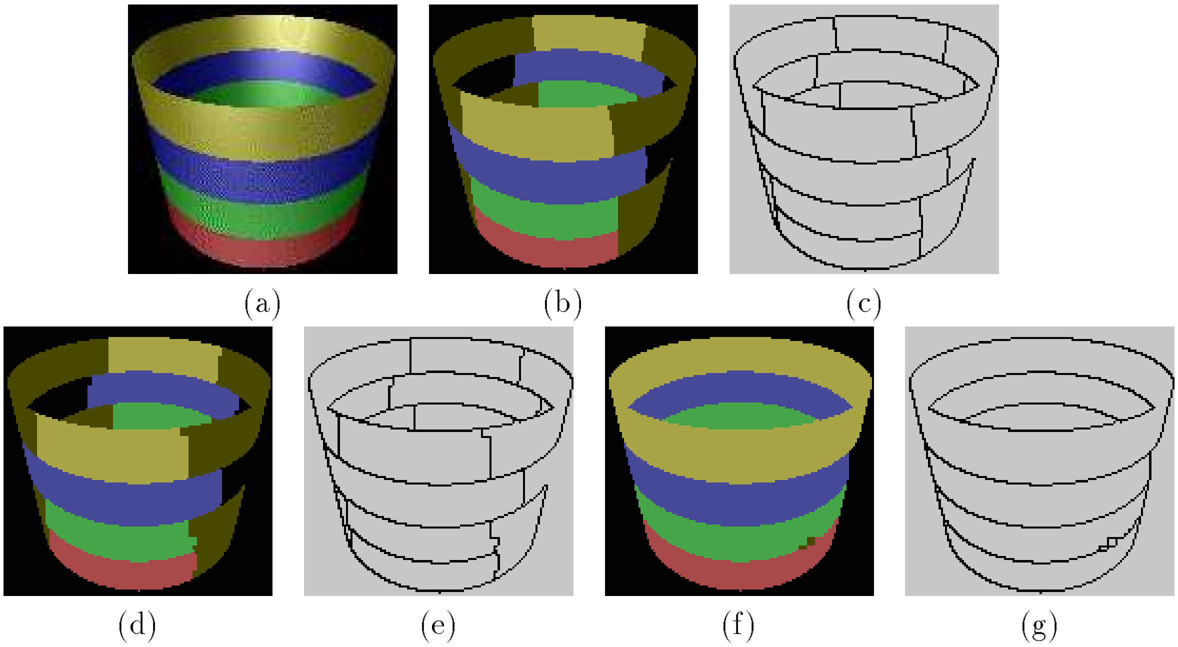


Figure 5.18: Effect of connecting neighbouring nodes, (a) original image with higher level of specular reflection; (b) segmentation of using SSF; (c) contour of (b); (d) segmentation of MCIS without connecting neighbouring nodes; (e) contour of (d); (f) segmentation of MCIS after connecting neighbouring nodes; (g) contour of (f).

5.3 Effect of colour

5.3.1 Motivation of using colour

Colour is a useful property which adds information to images. Compared with monochromatic images, colour images not only has the information about the brightness of each pixel as the monochromatic images do, but also has additional information such as hue and saturation. It has been argued for a long time whether this additional information in colour is really useful in image segmentation. Unfortunately, no consensus has been reached. In order to test this, an image "gloves", in both monochromatic and colour version, is used for segmentation. There are seven gloves in the picture with the colour of red, pink, white, purple, black and two shades of blue. Figure 5.19 shows the results of segmentation. In the segmentation of the colour image, almost all of the gloves can be separated except that the two shades of blue can not be distinguished completely. There is a clear boundary between the red and the pink gloves. But in the monochromatic image, the red and the pink gloves can not be distinguished because their grey scales are too close. This confirms that colour is useful in image segmentation.

5.3.2 Effect of colour spaces

In the experiments, the effect of using different colour spaces in MCIS is examined. The colour spaces compared in the experiments are RGB, HSV, I1-I2′-I3′, Lab, XYZ and YIQ. The images "crossboard", "cylinder", "girl" and "house" are also used in this set of experiments. For each image, the same number of clusters are extracted in SSF for the convenience of comparison. In each colour space, a series of experiments are performed with different parameters, and the best result, i.e., the result with lowest value of F, is picked. Figures 5.20 shows the optimal value of F versus the colour space for each image. Figure 5.21 and 5.22 show the corresponding

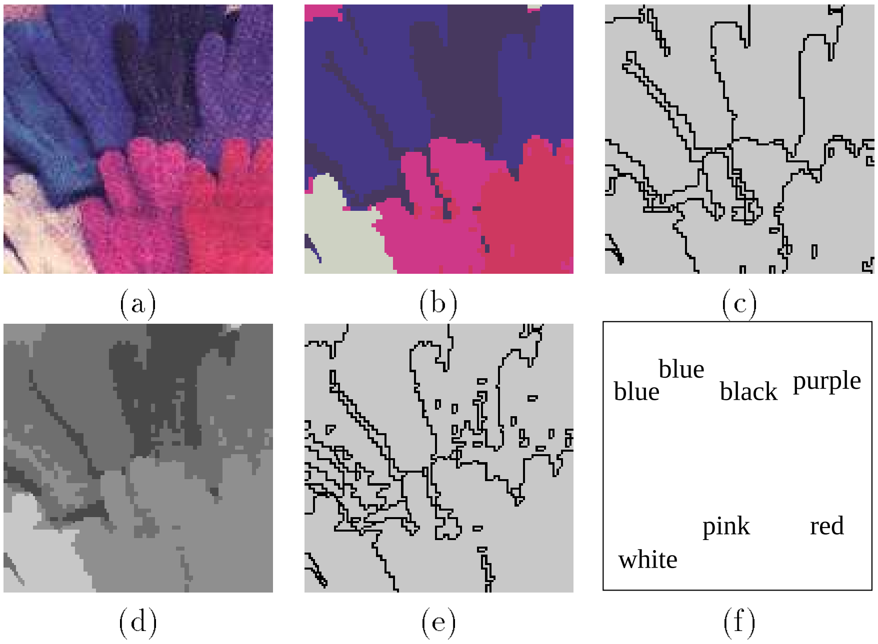


Figure 5.19: Segmentation results of colour and monochromatic images. (a) Original colour image; (b) segmentation of colour image; (c) contour of (b); (d) segmentation of monochromatic image; (e) contour of (d); (f) colour order of the gloves.

average colour error and the number of regions. Figures 5.23 to 5.28 show the best segmentation results, i.e., results with the lowest value of F, in different colour spaces. All of the segmented images are transformed back to the RGB space for comparison.

For the image "crossboard", the RGB space produces the least noisy segmentation, which is the best if the contours are examined. But obviously, the colour in the segmentation result of the RGB space is not quite right, because by comparing with the original image, the contrast in the segmented image is much higher. In Figure 5.21, it can be seen that the average colour error in this space is very high (next to the HSV). The result in the HSV space is the worst. Both the colour error and the number of regions are large, and thus the value of F is also large. In Figure 5.24, this result looks noisy and with too low contrast. The segmented images using the Lab, XYZ and YIQ spaces have small average colour errors, but they are more noisy and have a larger number of regions than that using the RGB and I1-I2′-I3′ spaces. The F values of these three spaces are similar. The result using the space I1-I2′-I3′ space has a small colour error and is not very noisy, and it has the smallest value of F. So it is the best result for the image "crossboard".

Since the image "cylinder" is very noisy, and the colour composition is more complex than that of the image "crossboard", the SSF fails to find the right clusters in colour spaces other than the RGB space if the original noisy image is used. In order to solve this problem, the image in each colour space is smoothed by taking the average colour vector in a 2×2 pixel window before SSF is used. After the smoothing, the noise is suppressed and the right clusters can be found correctly in the I1-I2′-I3′, XYZ and YIQ spaces. But in the spaces HSV and Lab, clustering information extracted is still not quite right and therefore the results are bad. In the final segmentation, the spaces RGB, I1-I2′-I3′ and YIQ produce comparable low values of F for the image "cylinder", and the result in the I1-I2′-I3′ is slightly better than the other two. The results in the three spaces successfully distinguish regions

Figure 5.20: Evaluation function F versus the colour spaces. 1: RGB, 2: HSV, 3: I1-I2′-I3′, 4: Lab, 5: XYZ, 6: YIQ.

Figure 5.21: Average colour distance versus the colour spaces. 1: RGB, 2: HSV, 3: I1-I2′-I3′, 4: Lab, 5: XYZ, 6: YIQ.

Figure 5.22: Region numbers versus the colour spaces. 1: RGB, 2: HSV, 3: I1-I2′-I3′, 4: Lab, 5: XYZ, 6: YIQ.

with different colours. The result in XYZ space has a larger value of F than that in the above three spaces because both the colour error and the number of regions in the XYZ space are larger than those in the above three spaces. In Figure 5.27 it can be seen that there are some miss labeled regions inside the top part of the cylinder. The results in the HSV and Lab spaces are very bad because the colour error and the number of regions are very large. In Figure 5.24, the segmented image for "cylinder" is very noisy, especially in the top and the bottom part of the cylinder. In Figure 5.26, the shape of the cylinder become very rough, and the dark background is assigned two colours (hardly distinguishable in the RGB space).

For the image "girl", the RGB space has the best result. The colour features are well captured during the SSF process and so the colour error is the smallest. This result also has the smallest number of regions. The HSV space has the worst result mainly because the colour error is too large, in particular, the colour in the face of the girl is much lighter than the original image. The result in the I1-I2′-I3′ space is not very good. Although the colour error in this space is also small, the value of F is large because there are too many small regions. The colour error in the Lab space is larger than that of I1-I2′-I3′ space, but the number of regions is smaller, and hence, the F value in the Lab space is similar to that in the I1-I2′-I3′ space. The segmented images in the I1-I2′-I3′ and Lab spaces have many artifacts due to the quadtree structure. It is noteworthy that although both the average colour error and the number of regions in the XYZ space are larger than that of the Lab space, the value of F is smaller. This is because the positions of the regions in the XYZ space are more accurate than those in the Lab space, i.e., the segmentation result in the XYZ space has much less artifacts that is due to the quadtree structure and has smoother boundaries. This suggests that the value of F also accounts for the smoothness of boundaries indirectly. In the YIQ space, the colour error is smaller than that of the XYZ space, and the number of regions is much less than that of the XYZ space, but the value of F is just slightly smaller than that of the XYZ space. The reason is also because the region accuracy, or the boundary smoothness, of the result in the YIQ space is not as good as the one in the XYZ space.

The space I1-I2′-I3′ also produces the best result for the image "house". Although the number of regions is not small, this result gets "bonus" for its colour accuracy and its preservation of details. The YIQ space has the second best result. Unlike the result in the I1-I2′-I3′ space, this result does not preserve as much details, e.g., in the upper face of the wall, but it has a relatively small colour error and a small number of regions and therefore has a small value of F. The results in the spaces RGB, XYZ and Lab have similar values of F. The colour errors in the three colour spaces are similar to that of the YIQ space but much larger than that of the I1-I2′-I3′ space, and the numbers of regions are similar to that of the I1-I2′-I3′ but much larger than that of the YIQ. Therefore, the values of F in the three spaces are larger than that in that I1-I2′-I3′ and YIQ spaces. Again, the HSV produced the worst result. Both the colour error and the number of regions are the largest. The result looks noisy, and unlike the result in the I1-I2′-I3′ space, the details in the segmented image does not conform to those in the original image.

From the above discussions, one can deduce that there does not exist a single colour space that is good for all images. The RGB spaces produces the best results with the image "girl", and the I1-I2′-I3′ space has the best results with the other three images. Generally speaking, the I1-I2′-I3′ space produces very good results in most cases. The HSV space produces the worst results because of the singularity problem inherent to this space. When the values of the R, G and B component of a pixel are identical, the hue (H) of this pixel is undefined, and in this thesis, it is assigned the value of 0 in such case. Besides, the saturation (S) becomes very unstable when the value of $R + G + B$ is small. As a result, it is hard to find the right clusters in this colour space during the SSF process and thus the average colour error is always

Figure 5.23: Segmentation results in the RGB space.

very high. In the multiresolution process of MCIS, it is also difficult to compare the similarity between nodes. The results in the HSV space are not discussed any more in the following comparison. Ohta et al. [28] use a similar colour space, DSH, which is similar to the HSV space. They try to avoid this problem by not using the S when the value of $R+G+B$ is less than 30 and not using the H if the value of $S \times (R+G+B)$ is less than 9. But this does not solve the problem. If there is a big dark cluster in the image, such as in the image "cylinder", this cluster will be lost. How to use the colour spaces with singularity is still an open problem.

5.4 Summary

In this chapter, a set of experiments are performed to examine the performance of MCIS under different parameters, different dissimilarity measures and different colour systems. In order to evaluate the segmentation results quantitatively, a new evaluation function F is proposed. This function can evaluate segmentation results

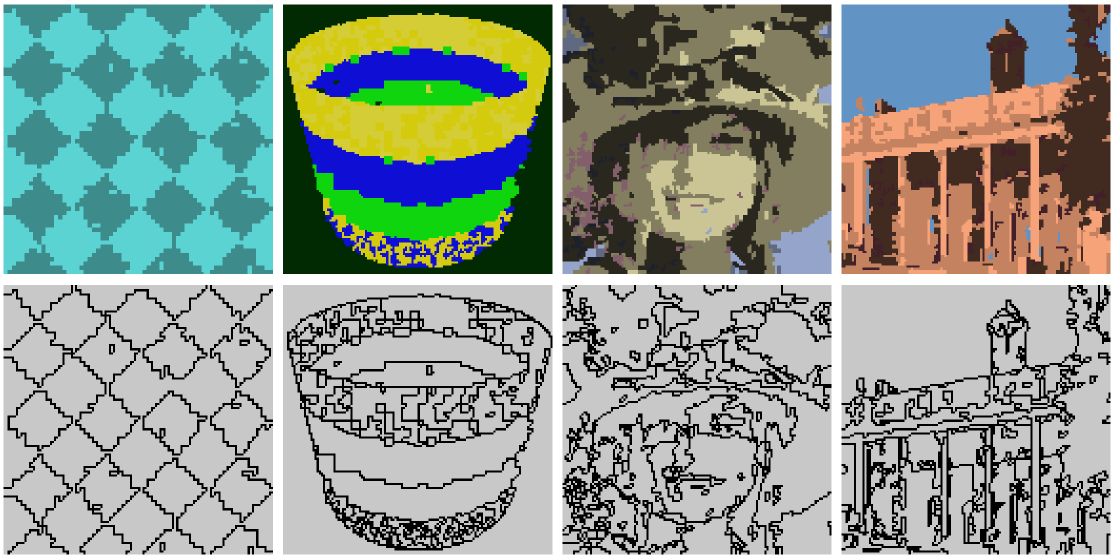


Figure 5.24: Segmentation results in the HSV space.

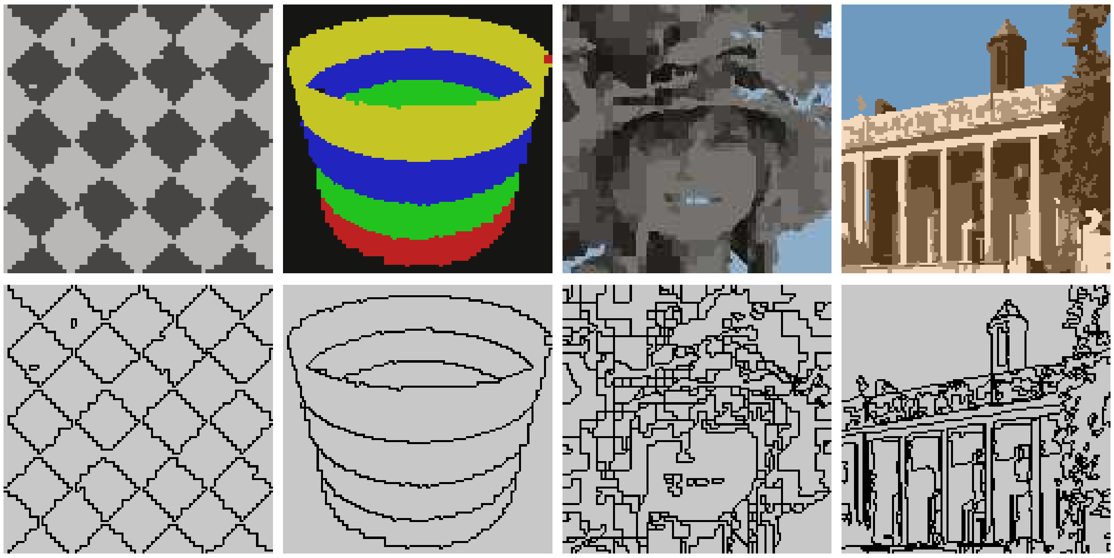


Figure 5.25: Segmentation results in the I1-I2-I3 space.

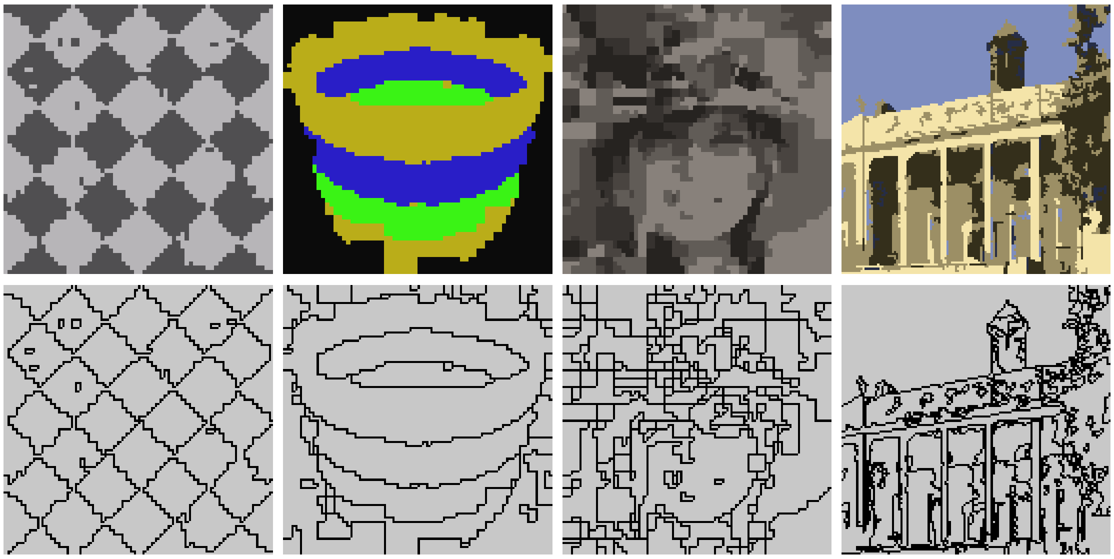


Figure 5.26: Segmentation results in the Lab space.

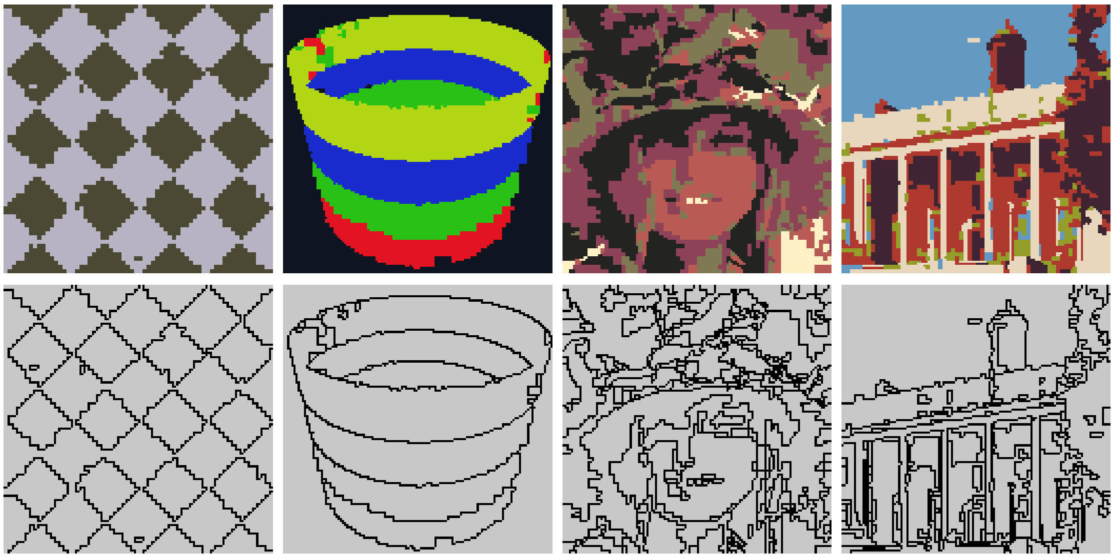


Figure 5.27: Segmentation results in the XYZ space.

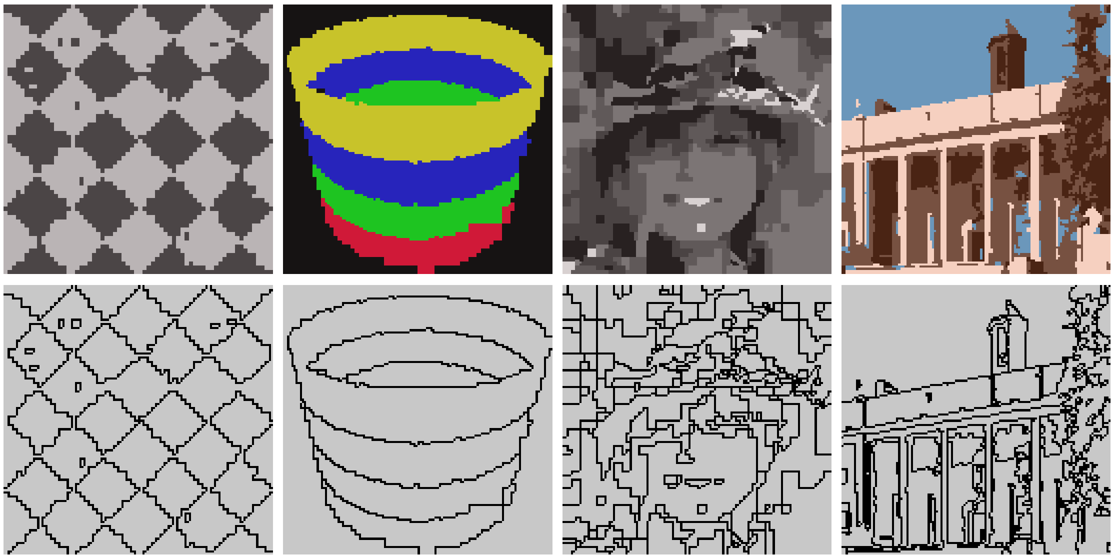


Figure 5.28: Segmentation results in the YIQ space.

both locally and globally for synthesized and real images, and it incorporates the segmentation criteria either directly or indirectly. F is used as a criterion in other parts of the experiments. Given a set of clusters of an image, F can be used to find the best segmentation result, i.e. one with the lowest value of F. It is noteworthy that the effect of simulated annealing has been evaluated, and the conclusion is that simulated annealing does not produce better results than that produced with instant cooling. Three dissimilarity measures in MCIS are examined, and the effect of these measures with respect to the size of the nodes are also studied. It was found that the KS-distance can be applied only when the size of the blocks are equal to or larger than 2×2. For nodes of size 1×1, the Euclidean distance is more appropriate. For noisy images, it is advisable to use large dissimilarity thresholds at the pixel level to suppress the noise while preserving details for larger nodes. Six colour systems, namely, RGB, HSV, I1-I2′-I3′, Lab, XYZ and YIQ, are compared for their effectiveness in segmentation. The I1-I2′-I3′ space produces very good results, and the HSV space has the worst results because of its inherent singularity problem.

Chapter 6

Conclusion

A new multiresolution color image segmentation (MCIS) algorithm is presented in this thesis. The first step of the proposed algorithm uses the SSF to get the clustering information, and to segment the image coarsely. Then the novel multiresolution scheme is used to refine the segmentation. The proposed approach has very good performance on both synthesized and real images. In order to evaluate the segmentation results quantitatively, a new evaluation function is proposed and developed. This evaluation function incorporates most of the factors in the segmentation result evaluation criteria. The effect of using different dissimilarity measures, the effect of simulated annealing and the effect of connecting neighbouring nodes are examined using the new evaluation function. The performance of using different colour spaces in segmentation is also studied both quantitatively and qualitatively, or visually. It is found that the I1-I2′-I3′ space produces the best results in most cases, and the RGB also produces very good results. The HSV space has the worst results because of the inherent singularity problem in this space.

References

[1] C. O. Acuna, "Texture modeling using Gibbs distribution," *CVGIP: Graphical Models and Image Processing*, Vol. 54, pp. 210-222, May 1992.

[2] D. H. Ballard and C. M. Brown, *Computer Vision*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1982.

[3] J. Besag, "On the statistical analysis of dirty pictures," *J. Royal. Statist. Soc.* B, Vol. 48, pp. 259-302, 1986.

[4] P. Burger and Duncan Gillies, *Interactive computer graphics,"* Addison-Wesley publishing company, Wokingham, England, 1989.

[5] M. Celenk, "A colour clustering technique for image segmentation," *Computer Vision, Graphics and Image Processing*, Vol. 52, pp.145-170, Nov. 1990.

[6] M. J. Daily, "Colour image segmentation using Markov Random Fields," *IEEE Proc. Computer Society Conference on Computer Vision and Pattern Recognition*, San Diego, California, June 1989, pp. 304-312.

[7] R. C. Dubes, A. K. Jain, S. G. Nadabar and C. C. Chen, "MRF model-based algorithms for image segmentation," *???*.

[8] K. Fukunaga, *Introduction to statistical pattern recognition*, Academic Press Inc., San Diego, California, 1982.

[9] J. A. Gartaganis and J. Tartar, *Wave-based spectrum rendering,* Technical report TR 90-13, Dept. of Computing Science, University of Alberta, 1990.

[10] J. Gauch and C-W. Hsia, "A comparison of three colour image segmentation algorithms in four colour spaces," *???????????*

[11] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 6, pp. 721-741, November 1984.

[12] D. Geman, S. Geman, C. Graffigne, and P. Dong, "Boundary detection by constrained optimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 12, pp. 609-628, July 1990.

[13] R. Gershon, *Survey on colour: aspects of perception and computation,* Technical Report RCBV-TR-84-4, Dept. of Computer Science, Univ. of Toronto, July 1984.

[14] R. Gershon, *The use of colour in computational vision,* Technical Report RBCV-TR-87-15, Dept. of Computer Science, Univ. of Toronto, June 1987.

[15] M. Gonzalez-Rodreguez, D. Benitez-Diaz, and C. P. Suarez-Araujo, "Segmentation and recognition in visual chromatic spaces," *Cybernetics and Systems: An International Journal*, Vol. 21, pp. 241-247, 1990.

[16] R. M. Haralick and L. G. Shapiro, "Survey — image segmentation techniques," *Computer Vision, Graphics, and Image Processing*, Vol. 29, pp. 100-132, 1985.

[17] G. Healey and T. O. Binford, "A colour metric for computer vision," *IEEE Proc. of Computer Vision and Pattern Recognition*, Ann Arbor, Michigan, June 1988, pp. 10-17.

[18] G. Healey, "Using colour for geometry-insensitive segmentation," *J. Opt. Soc. Am.A*, Vol. 6, pp. 920-937, June 1989.

[19] G. Healey, "Segmenting images using normalize colour", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 22, pp. 64-73, Jan./Feb., 1992.

[20] C. L. Huang, T. Y. Cheng and C. C. Chen, "Colour images segmentation using Scale Space Filter and Markov Random Field," *Pattern Recognition,* Vol. 25, pp. 1217-1229, 1992.

[21] J. Kender, *Saturation, hue, and normalized colour: calculation, digitization effects, and use,* Technical Report, Dept. of Computer Science, Carnegie-Mellon Univ., November 1976.

[22] G. J. Klinker, S. A. Shafer, and T. Kanade, "Image segmentation and reflection analysis through colour," *Proc. of Image Understanding Workshop*, Cambridge, Massachusetts, April 1988, pp. 838-853.

[23] G. J. Klinker, S. A. Shafer, and T. Kanade, "The measurement of highlights in colour images," *International Journal of Computer Vision*, Vol. 2, pp. 7-32, 1988.

[24] G. J. Klinker, S. A. Shafer, and T. Kanade, "A physical approach to colour image understanding," *International Journal of Computer Vision*, Vol. 4, pp. 7-38, 1990.

[25] Y. W. Lim and S. U. Lee, "On the colour image segmentation algorithm based on the thresholding and the fuzzy c-means techniques," *IEEE Transactions on Pattern Analysis and Machine Intelligence.* Vol. 12, pp. 609-628, July 1990.

[26] R. Muzzolini, Y. H. Yang, and R. Pierson, "Multiresolution texture segmentation with application to diagnostic ultrasound images," TR-91-13, Computer Vision Lab, Dept. of Computational Sci., Univ. of Saskatchewan, 1991, to appear in *IEEE Trans. on Medical Imaging.*

[27] R. Ohlander, K. Price, and D. R. Reddy, "Picture segmentation using a recursive region splitting method," *Computer Graphics and Image Processing*, Vol. 8, pp. 313-333, 1978.

[28] Y. Ohta, T. Kanade, and T. Sakai, "Colour information for region segmentation," *Computer Graphics and Image Processing*, Vol. 13, pp. 222-241, 1980.

[29] W. K. Pratt, *Digital Image Processing*, John Wiley & Sons, Inc., New York, 1978.

[30] A. Sarabi and J. K. Aggarwal, "Segmentation of chromatic images," *Pattern Recognition,* Vol. 13, pp. 417-427, 1981.

[31] M. J. Swain, *Colour indexing,* Technical Report 360, Dept. of Computer science, Univ. of Rochester, 1990.

[32] T. F. Syeda, "Finding distinctive coloured regions in images," *SPIE Intelligent Robots and Computer Vision IX: Algorithms and Techniques*, Vol. 1381, pp.574-581, 1990.

[33] T. Taxt and E. Bolviken, "Relaxation using models from quantum mechanics", *Pattern Recognition*, Vol. 24, pp. 695-709, 1991.

[34] A. P. Witkin, "Scale space filtering: a new approach to multi-scale description," *Image understanding*, S. Ullman and W. Richards, Eds., pp. 79-95, Ablex Publishing, N.J., 1984.