COLOUR SEGMENTATION AND FIGURE-GROUND SEGREGATION OF NATURAL IMAGES

WONG SWEE SEONG

(B. Sc. (Comp. Sc.) (Hons. NUS))

A THESIS SUBMITTED
FOR THE DEGREE OF MASTER OF SCIENCE
DEPARTMENT OF COMPUTER SCIENCE
SCHOOL OF COMPUTING
NATIONAL UNIVERSITY OF SINGAPORE
2000

Acknowledgements

This Master thesis has been the result of many months of hard work. In the process, many people have been there for me and I would very much like to thank them. They have given me their support either directly by contributing useful ideas to my work, or simply be around when I need someone to hear me out.

First of all, my supervisor Dr. Leow Wee Kheng for withstanding my stubbornness and wrestling ideas with me. Dr. Sung Kah Kay for being my advisor and giving me directions for my research along the way. Also are my peers, just to name a few are David Ong my faithful roommate, Shen Tat the fun contributor, Seet Chong the ‘pressurizer”, Joseph Lee the ‘brainy” one, Wee Yeh the joyous fellow and many others who share the 4th floor of S15 building. My graditude to them for making my stay a memorable one.

Next, my wife Karen for pushing me on through my down time and most of all for her love. Not to forget, my family members who are most encouraging and for having faith in what I am doing.

Contents

List of Figures

Summary

To recognize the objects in an image and to understand the image content, a computer system has to first separate the foreground objects from the background. Image segmentation and figure-ground segregation are, therefore, essential for computer image understanding. A system called OLAG, for Object-LAyer Grouping, is developed for image segmentation and figure-ground segregation. OLAG consists of several incremental refinement steps which use colour and other visual cues such as size and compactness for grouping the image pixels. It produces, as an end result, a set of layers each containing an object or several object parts. Figure and ground relationships among the objects are inferred, giving their relative depths. OLAG provides a framework in which other visual cues such as texture and shape can be easily incorporated to identify objects. It is shown that the system can produce interesting and useful segmentation results.

Chapter 1

Introduction

1.1 Motivation

To recognize the objects in an image and to understand the image content, a computer system has to first separate the foreground objects from the background. Image segmentation and figure-ground segregation are, therefore, essential for computer image understanding. Image segmentation and figure-ground segregation are closely related problems. A good colour segmentation can be used as a preliminary result for grouping image pixels into regions. Regions identified in colour segmentation can be further grouped together based on other visual properties to isolate the objects in the image.

The objects and figure-ground information extracted from the images can also be used for content-based multimedia retrieval. A natural way for a user to specify the required images is to indicate the content and the spatial layout of the images. To accurately retrieve these images, a multimedia retrieval system needs to know what and where are the objects in the images. Figure-ground information provides, in addition to positional information in a 2-D plane, relative depths of the objects which are needed to fully describe the 3-D layout of the images.

1.2 Objective

The overall objective of this research is to develop a system called OLAG, for Object-LAyer Grouping, that performs image segmentation and figure-ground segregation of natural images. An image can contain many different types of objects and background regions. This research focuses on objects that are visually obvious and distinct from the background and neighbouring objects. The segmentation result should have the following properties:

1. The objects should have visual properties such as colour and texture that differ significantly from those of the neighbouring objects. The current implementation of OLAG focuses on using colour to identify objects. Other

properties like proximity and compactness are also incorporated in OLAG, demonstrating the flexibility of OLAG in including other features.

2. An object may consists of several disjoint parts and may be occluded by other objects. It is useful to group the disjoint parts of an object together and identify them as parts of a single object. On the other hand, an object part that is visually distinct from other parts may be considered as a separate object. For example, the windows and the roof are significant parts of a house and thus may be considered as separate objects (at a fine scale) and as parts of the house (at a coarse scale).

3. The objects and object parts should have significant sizes compared to the image.

4. The boundaries of the objects and regions should correspond to physical boundaries.

5. The colours in the objects and regions need not be strictly homogeneous. Instead, homogenuity in the regions is not overly emphasized since natural images usually contain much variation in colour even within the same object.

It is difficult to define a good measurement of the quality of the segmentation result. So far, we can only rely on visual inspection to assess the quality of the segmentation result based on the above guidelines.

1.3 Thesis Overview

To lay the ground for further discussion, Chapter 2 first introduces some background concepts such as colour spaces, noise in an image, common smoothing algorithms and general approaches to colour segmentation. With these concepts described, it is then appropriate to discuss the steps involved in OLAG in Chapter 3. In the next chapter, OLAG's segmented results are illustrated and compared with those of other existing segmentation methods. Finally, Chapter 5 discusses future works that can improve and enhance OLAG. In addition, Chapter 5 also summarizes the contributions of this research and concludes the thesis.

Chapter 2

Background Literature and Related Works

To obtain good colour segmentation result, many issues need to be addressed. For example, noise present in images needs to be removed as it causes error in grouping pixels into likely objects. There is also a need to determine the colour space in which colour similarity can be measured. Finally, current approaches taken to perform colour segmentation and figure-ground segregation are examined in this chapter.

2.1 Background Literature

2.1.1 Noise in Natural Images

Noise in images can be classified as random noise or natural noise. Random noise is caused by image digitization (e.g. scanning, paper print quality), lossy image format (e.g. jpeg) or colour reduction (dithered colours). Natural noise is caused by imperfection existing in natural scene itself, e.g., uneven colour tone of object surface. Such irregularity in colour tone may be due to stain, dust or varying length of exposure to daylight which causes fading of colour. In addition, real world objects may not be truly homogeneous in colour. They are often made up of varying fine grained colours (i.e. fine colour texture). Examples of these objects and regions include sand patches, roads and fabrics. Other sources of imperfection include varying illumination sources and contoured surfaces of objects which produce varied reflections (e.g. shadows and highlights).

It is essential that image noise be removed or reduced as much as possible. A common approach is to smooth the image. Smoothing also removes fine colour texture and makes the regions uniform in colour. In OLAG, smoothing is carried out before segmentation to remove noise in the image while retaining the colour edge information.

2.1.2 Smoothing

Gaussian filter is the most commonly used smoothing filter to date. It is a linear filter and the filtering result is easy to compute. However, the shortcoming of Gaussian filter is that smoothing is performed throughout the whole image, even across boundaries where there are sharp local changes in colour. This is undesirable as the boundaries define the outline of objects in the image and should be retained in the smoothing process.

To retain the boundaries while smoothing, many edge-preserving filters have been proposed. The general approach is to replace the value of the pixel to be smoothed by the weighted sum of the values of selected pixels from the neighbourhood. Ideally, the selected pixels should belong to the same region as the smoothed pixel. For example, the weighted-sum filter performs a weighted sum over all pixels within an $n \times n$ window centred at the pixel to be smoothed. The weights are usually inversely proportional to the magnitude of the gradient between the center pixel and its neighbouring pixels, or other measure of dissimilarity between the two pixels [19, 22]. The strength of weighted-sum filter is that those neighbours more similar to the center pixel contribute more to the replaced value of the center pixel. Taking all neighbouring pixels into consideration can incur errors when the center pixel is located at the colour boundary because its replaced value is corrupted by the values of the pixels across the boundary.

Another approach, the k-nearest-neighbour filtering, replaces the center pixel's value by the average of the k nearest neighbours located within an $n \times n$ window [6, 14]. Symmetric nearest-neighbour filter looks at a 3×3 window in which each neighbouring pixel is paired with the pixel on the opposite side across the center pixel. For each pair, the pixel closest in colour to the center pixel is chosen. The center pixel's colour is then replaced by the average of the colour of four chosen neighbours [12, 16].

In a note-worthy work by Rehrmann and Priese [17, 18], the authors reported their investigation of three non-linear filters namely, median-filter, k-nearest-neighbour filter (knn) and symmetric nearest-neighbour filter (snn) used for edge-preserving smoothing. Judging from the computational complexity and test results, it is reported that the snn filter is the fastest of the three. Based on visual inspection of several hundreds of test results, it is concluded that the snn filter and the knn filter have comparable performance and are better than the median-filter.

OLAG implements a smoothing filter that is a combination of the snn filter and the weighted-sum filter. This method takes advantage of both the pixel selection strategy of the snn filter and the weighted contribution policy of the weighted-sum filter (Section 3.3).

2.1.3 Colour Space

There exist many different colour spaces that are used in different applications or equipment. For example, for computer display, a colour is produced by the excitation of red, green and blue phosphors on the CRT faceplate. Red, green and blue (RGB) are the most native representation used for digitized images. RGB colour space takes the shape of a cube—its regularity in dimension makes it easy to use. For the purpose of colour segmentation, RGB colour space is not commonly used because it is not perceptually linear. A colour space is perceptually linear if the Euclidean distance between any two colour points in the colour space is proportional to human's perceptual difference. The RGB colour space is also not intuitive as human does not perceive colour as an additive composition of R, G and B components.

There are two colour spaces that are more commonly used in colour image processing, namely HSI and CIE L*u*v*. In hue, saturation and intensity space (HSI), hue refers to human's perception of what colour is, for example, red, blue or green. Saturation of a colour means the richness of the colour, whether the colour is strong or dull. Intensity measures the brightness of the colour. HSI colour space represents a collection of similar colour spaces including HSV (V for value), HCI (C for chroma/colourfulness), HVC, etc. Most of these colour spaces can be linearly transformed from the RGB space. This means that every point in the RGB space is mapped onto a unique point in the HSI colour space. HSI space generally has an inverted cone shape. The main cited reason for its use in colour image processing is the intuitive meaning of its various components. In particular, colour difference between any two colours is measured using the hue component. A difficulty in using the HSI space is that at low intensity, all colours are actually perceived by humans to be similar to black. Thus hue is not useful for determining colour difference at low intensity. Various methods are used by different researchers to handle this problem. Some methods even apply different rules for colour distance measurement to different parts of the colour space [18].

In CIE L*u*v* colour space, L* refers to luminosity, and u* and v* jointly define the chromatic components. The L*u*v* colour space is an empirical model based on visual experiments conducted on human subjects and hence is approximately perceptually uniform. Intuitive colour information like hue and saturation can be derived from the u* and v* components. The L*u*v* space does not have a uniform shape but there is a nonlinear reversible conversion between RGB and L*u*v* spaces. OLAG adopts the L*u*v* colour space because it is perceptually uniform. The Euclidean distance between any two points in the colour space correlate well with human's perception of the difference between the two colours.

2.2 Related Works

2.2.1 Colour Segmentation

Image segmentation methods can be grouped into several categories: *edge detection*, *region merging and splitting*, *optimization* and *clustering*.

Edge detection methods [3] look for sharp changes in intensity or colour in the image. Edges detected are usually accurately located but the edges may not form closed contours. Further processing such as boundary completion and merging are needed to close up the gaps between the edges.

Region merging techniques repeatedly merge neighbouring regions together into larger regions. The merging process usually starts by grouping neighbouring image pixels into many small regions [18]. Seed growing is a similar idea whereby only a few small seed regions are selected and other pixels are slowly grouped into them [1]. An alternative approach is to start with the image as a single region and repeatedly split larger regions into smaller regions. This technique is usually used as a corrective step to rectify over-merging. Region merging is the most popular approach in segmentation and is often used as part of more comprehensive approaches.

Optimization techniques define a global function that measures the goodness of the segmentation result and seek to optimize the result. Examples of these techniques include Bayesian/Minimum Description Length criteria [23] and Markov random fields. Other approaches such as the snake and the balloon methods define optimization function for local region boundaries or contours [5]. The function is optimized if the contour matches the desired region boundary. Optimization techniques for finding a global solution can be computationally intensive and do not necessarily give better results than do simpler methods.

Clustering methods perform grouping in the feature space. Pixels in an image are mapped into points in a feature space (e.g. the colour space) [21]. These points are then clustered together based on the distance between the points measured in the feature space. The clusters are then re-mapped back to the image forming the various regions. Clustering methods based on colour histograms [15, 20] are commonly used. The recursive histogram splitting method is an example of these methods [4]. A histogram of colour pixels is constructed for each region in the image. Based on the histograms, the method determines whether to split larger regions into smaller regions. Although the colour histogram provides good global statistical information about the image in terms of colour, it does not capture the spatial distribution of the pixels in the image.

Two recent works on image segmentation are particularly interesting as they are applied to natural images. These works are described as follows:

1. **Region Competition**

   Chun Zhu and Yuille (1996) [23] presented a general algorithm for image segmentation called Region Competition. The algorithm seeks to find the local minimum of a global optimization function. The regions compete for the ownership of pixels adjacent to them. Neighbouring regions can be merged if the merger improves the global optimization function significantly. Additional seeds may be added in the process to absorb "unclaimed" pixels. The process is iterative and the number of iteration is difficult to determine. The number of seeds used also affects the segmentation result.

2. **JSEG colour segmentation**

   Ma and Manjunath (1999) [7] proposed a segmentation algorithm called "JSEG". Colours used in the image are first quantized into several representative classes based on vector quantization. Colours of image pixels are then replaced by their corresponding colour class labels. Based on the distribution pattern of labels, a statistical measurement of good segmentation is computed. Image segmentation is performed in two stages, namely region growing (similar to seed growing) and region merging. The quantization of colours in the image into representative classes is crucial for achieving good segmentation result.

In comparison, OLAG also adopts the region merging approach, but it applies different merging strategies. Colour edge information is used to perform an initial grouping of the pixels into regions. These regions are then merged together iteratively. Local histograms based on the colour difference of adjacent pixels, instead of colour alone, are used to determine suitable colour similarity for the merging process. In this way, spatial information is also incorporated into the histogram.

2.2.2 Figure-ground Segregation

Many factors contribute to figure-ground segregation, including differences in luminance, colour, size, binocular disparity and motion between a figure and its background [10]. Figure-ground segregation has not been adequately addressed in computer vision research. There are works that focus on formulating theories of human perception [2] but no actual system has been implemented. There are also several works on building neural network or mathematical models of figure-ground segregation but these models have been tested only on simple images containing uniformly coloured objects and background [9, 11, 13].

In contrast, OLAG integrates image segmentation and figure-ground segregation into a single system. The system is tested on natural images, and inferences are made about the depth relationships among the objects identified. In addition, it can also identify disjoint parts of the same object.

Chapter 3

The OLAG System

3.1 System Overview

The OLAG system defines a framework that performs colour segmentation and figure-ground segregation progressively. First, an input image is segmented by colour into *regions*, each consisting of a set of connected pixels. As an object may consist of several disjoint parts, regions that are likely to belong to the same object are grouped together into a *cluster* of possibly disconnected regions. Colour alone is not sufficient for identifying object parts. Therefore, other visual properties such as size and compactness are used to merge the clusters of regions together. Each cluster of regions corresponds to a possible object or parts of an object. Assuming that a full segmentation of the image into objects is achieved, it is then possible to infer the depth relationships among the objects. Each cluster of regions is placed in a separate *layer* which represents the relative depth position.

OLAG follows a conservative approach in pixel and region grouping. Pixels are grouped into regions, and regions into cluster, only if there is enough confidence that the grouping produces correct results. Otherwise, the grouping decision is postponed to the next step in OLAG. In the initial steps, simple but strict colour similarity criteria are used to quickly group pixels into regions. Due to the strictness of the criteria, a large number of regions with homogeneous colours are produced resulting in over-segmentation of the image. In subsequent steps, more complex criteria are used to further group the regions together. With the use of complex grouping criteria, the colour similarity criteria can be relaxed to encourage further merging of regions. This incremental refinement method avoids the need to split regions that are already merged and improves OLAG's efficiency because complex grouping criteria are applied only on large regions instead of all the pixels.

OLAG consists of four stages (Fig. 3.1), each of which is further divided into several steps:

1. The **Pre-Processing** stage prepares the underlying image for subsequent grouping by transforming the image into a suitable colour domain and removing the noise present in the image.

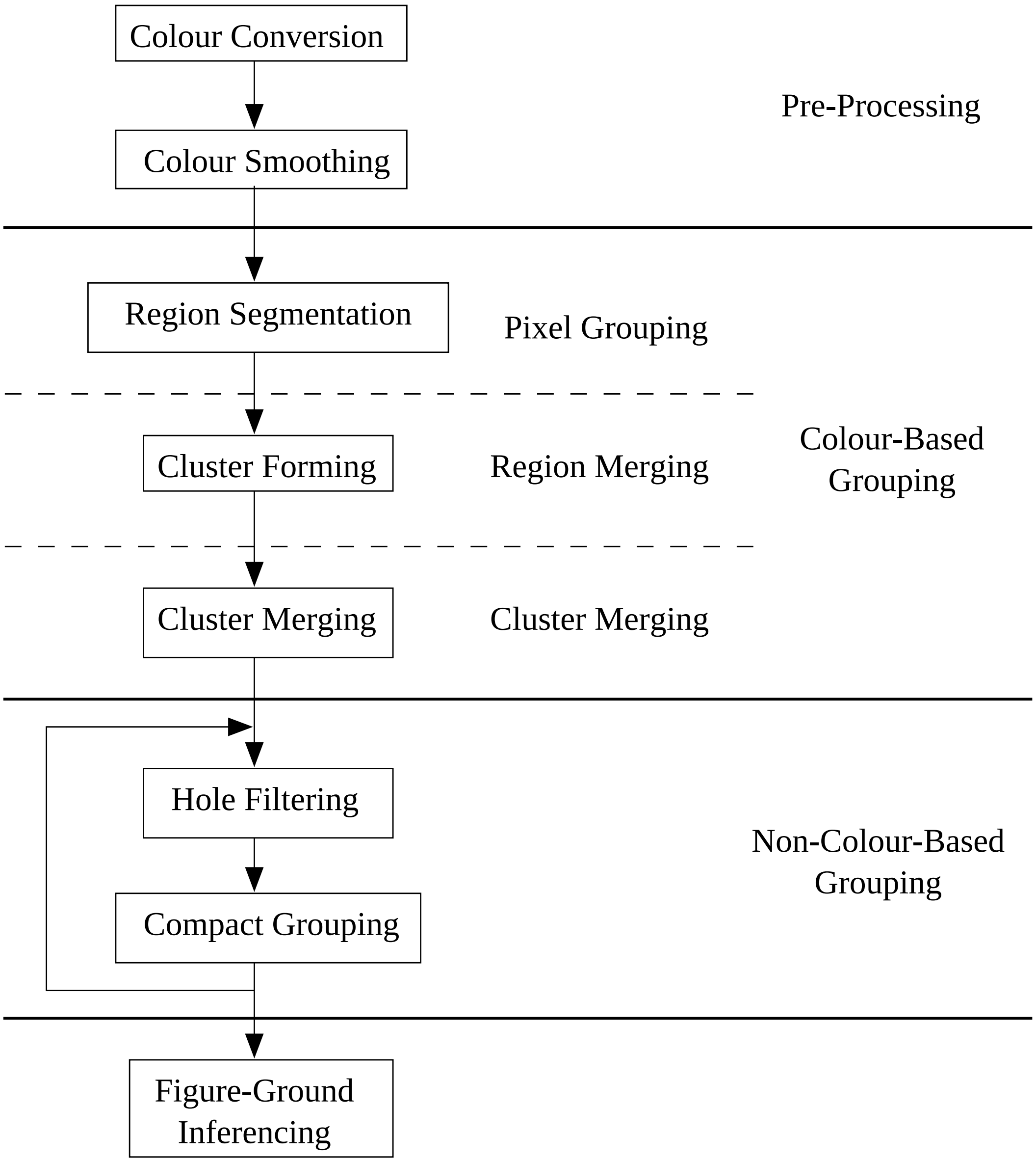


Figure 3.1: Structural flow of the OLAG system.

- The Colour Conversion step transforms the pixels from the RGB colour space to the L*u*v* space.
- The Colour Smoothing step performs edge-preserving smoothing which results in highly smoothed regions without blurring object boundaries.

2. The **Colour-Based Grouping** stage, as the name suggests, merges pixels and regions together based on their colour similarity.

   - The Region Segmentation step merges adjacent pixels into regions based on colour similarity. This step basically uses the colour difference of adjacent pixels to define the boundaries of regions.
   - The Cluster Forming step combines possibly disjoint regions with similar colours, which may be considered as parts of the same object, into a cluster of regions.
   - The Cluster Merging step further groups adjacent clusters of regions together by relaxing the colour similarity criteria. At the end this step, colour-based grouping is completed.

3. The **Non-Colour-Based Grouping** stage further groups clusters of regions together based on other non-colour visual properties to improve the segmentation result.

   - The Hole Filtering step examines the significance of the size of the regions obtained from the previous stage. If the size of a region is insignificant compared to its neighbouring regions, it is labelled as a "hole" which will be merged into one of its surrounding regions. This step effectively removes small regions and non-essential detailed features detected in the image.

- The Compact Grouping step combines clusters of regions to form more compact clusters. This step illustrates the ease of incorporating other visual properties into OLAG for image segmentation. As an illustration, the compactness criterion is used in the current implementation of OLAG.

4. The **Figure-Ground Inferencing** stage applies inference rules to determine the figure-ground relationship among the clusters of regions identified. Since each cluster of regions very likely correspond to an object or object parts, this step essentially infers the relative depths of the objects in the image.

3.2 Colour Conversion

The first step in OLAG is to convert the input image from the RGB colour space to the CIE L*u*v* space. The RGB values of a colour are first converted into the CIE XYZ tristimulus values:

$$
\begin{aligned}
X &= c_{11}R + c_{12}G + c_{13}B \\
Y &= c_{21}R + c_{22}G + c_{23}B \qquad (3.1) \\
Z &= c_{31}R + c_{32}G + c_{33}B.
\end{aligned}
$$

The coefficients c_{11} , . . . , c_{33} are equipment dependant. For OLAG, coefficients for the CIE colour-matching functions [8] are adopted:

$$
\begin{pmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{pmatrix} = \begin{pmatrix} 0.49 & 0.31 & 0.20 \\ 0.17697 & 0.81240 & 0.01063 \\ 0.0 & 0.01 & 0.99 \end{pmatrix}.
$$

Next, the XYZ tristimulus values are converted to L*, u* and v* values as follows:

$$
L^* = \begin{cases} 116\left(\frac{Y}{Y_n}\right)^{1/3} - 16 & \text{if } \frac{Y}{Y_n} > 0.008856 \\ 903.3\left(\frac{Y}{Y_n}\right) & \text{otherwise} \end{cases} \qquad (3.2)
$$

$$
\begin{aligned}
u^* &= 13L^*(u - u_n) \\
v^* &= 13L^*(v - v_n)
\end{aligned}
$$

where

$$
\begin{aligned}
u &= \frac{4X}{X + 15Y + 3Z} \\
v &= \frac{9Y}{X + 15Y + 3Z}.
\end{aligned}
$$

For the CIE colour-matching functions,

$$X_n = Y_n = Z_n = 1$$

$$\begin{aligned} u_n = & \frac{4X_n}{X_n + 15Y_n + 3Z_n} = \frac{4}{19} \\ v_n = & \frac{9Y_n}{X_n + 15Y_n + 3Z_n} = \frac{9}{19}. \end{aligned}$$

3.3 Colour Smoothing

The purpose of smoothing is to remove random noise and fine colour texture in the image. Ideally smoothing should even out the colour values of the pixels within the same region, and maintain or enhance the colour difference between different regions.

OLAG uses a hybrid smoothing filter which combines the weighted-sum filter and the symmetric nearest-neighbour filter. The smoothing algorithm follows the pixel selection strategy of the symmetric-nearest neighbour filter (Eq. 3.4) but the sum of the chosen pixels are weighted according to the colour distance between the chosen pixels and the center pixel (Eq. 3.5). The closer the colour, the larger is the assigned weight.

Let $\mathbf{C}(i) = (L_i, u_i, v_i)$ denote the colour of pixel i, where L, u and v are the L*, u* and v* colour components of the pixel. The colour distance CD_{ij} between pixels i and j is given as:

$$CD_{ij} = \|\mathbf{C}(i) - \mathbf{C}(j)\| \tag{3.3}$$

$$= \sqrt{(L_i - L_j)^2 + (u_i - u_j)^2 + (v_i - v_j)^2}.$$

Consider a 3×3 window centred at pixel i (Fig. 3.2). The center pixel has 4 pairs of opposite neighbours: pixels j and $j+4$, $1 \leq j \leq 4$. From each pair of pixels, the one with the most similar colour compared to the center pixel is chosen:

$$P_{ij} = \begin{cases} j & \text{if } CD_{ij} < CD_{i,j+4} \\ j+4 & \text{otherwise} \end{cases} \tag{3.4}$$

Then, the center pixel is assigned the new colour

$$\mathbf{C}(i) = \frac{\sum_{j=1}^{4} w_{ij}\mathbf{C}(P_{ij})}{\sum_{j=1}^{4} w_{ij}}. \tag{3.5}$$

The weight w_{ij} is measured relative to the largest colour distance CD_{max} between any center pixel and its neighbouring pixels:

$$w_{ij} = \frac{CD_{max} - CD_{i\,P_{ij}}}{CD_{max}} \tag{3.6}$$

$$CD_{max} = \max_i \max_{j=1}^{8} CD_{ij}$$

The smoothing process iterates until the image is smooth enough. The measure of smoothness Θ of an image is defined as the average colour distance over all adjacent pixel pairs in the image:

$$\Theta = \frac{1}{N}\sum_i \sum_{j\in\{2,4,6,8\}} CD_{ij} \tag{3.7}$$

where pixels i and j are adjacent pixels (Fig. 3.2) and N is the number of adjacent pairs in the image. The average colour distance is used as the smoothness measure

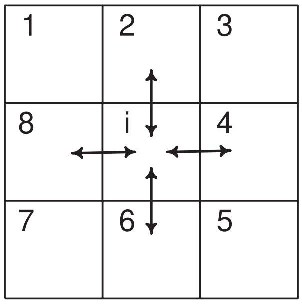
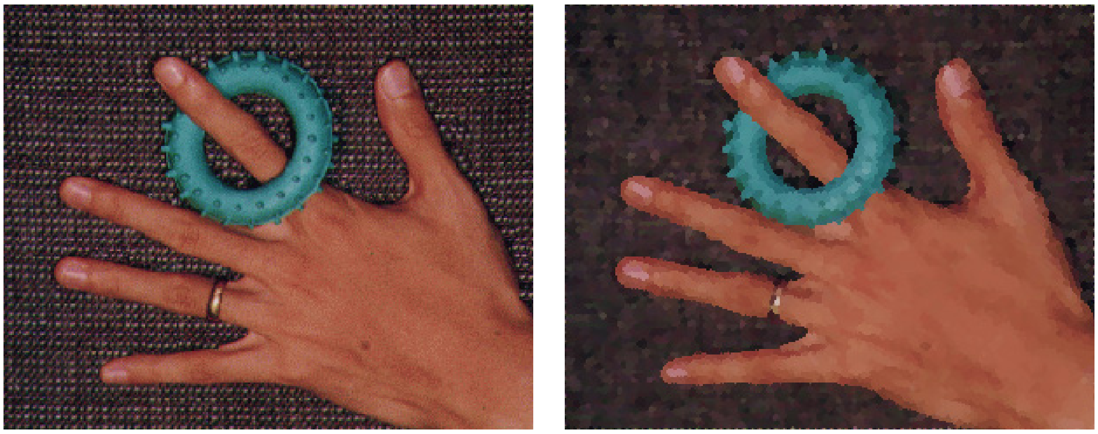


Figure 3.2: Neighbouring pixels and adjacent pixel pairs. All the eight neighbouring pixels, 1 to 8, are used in computing the smoothed colour value of the center pixel i (Eq. 3.5). On the other hand, only pixels 2, 4, 6 and 8 are defined as the *adjacent pixels* used in computing the smoothness of the image (Eq. 3.7) and in subsequent processing steps.

because as the iteration progresses, the pixel values will change and then stabilize, and the difference in the average colour distance will slowly converge to 0. In OLAG, the smoothing process terminates when the difference in smoothness measure between two successive iterations is smaller than 0.05. Experimental results show that any further smoothing will not improve the segmentation result. Typically, the number of iterations required lies between 5 and 15. Figure 3.3 illustrates an example of the image smoothing result. Note that the textured background is smoothed while the boundaries of the objects remain distinct.



(a) (b)

Figure 3.3: Image smoothing result. (a) The input image has a smoothness measure Θ of 17.93. (b) After smoothing the image for 14 iterations, its smoothness Θ drops to 3.38. The textured background is smoothed while the boundaries of the objects remain distinct.

3.4 Region Segmentation

The region segmentation step of OLAG performs a preliminary segmentation of the smoothed image to obtain regions that are generally homogeneous in colour. This segmentation step looks at the local colour similarity between adjacent pixels to decide if pairs of adjacent pixels are to be grouped into the same regions.

Before examining the actual algorithm, an illustration is given to show how the approach is derived. A pair of adjacent pixels can be classified as either an *inner* or *outer* pixel pair. An inner pixel pair is a pair of adjacent pixels that both belong to the same region. If the adjacent pixels belong to different regions, they are referred to as an outer pixel pair. Consider the following constraints:

1. number of inner pixel pairs > number of outer pixel pairs [a]

2. for any inner and outer pairs, CD (inner pair) < CD (outer pair). [b]

Given these constraints, the problem of colour segmentation can then be framed as finding the colour distance that differentiates the inner pairs from the outer pairs. Constraint [b] specifies that such a point of separation of colour distance exists, while constraint [a] restricts the point of separation to lie in the upper half of the histogram of colour distance between all pairs of adjacent pixels (Fig. 3.4). Constraint [a] can safely be taken to be true for all natural images. Otherwise, the underlying image will be so fragmented that it does not contain significantly large regions that correspond to objects.

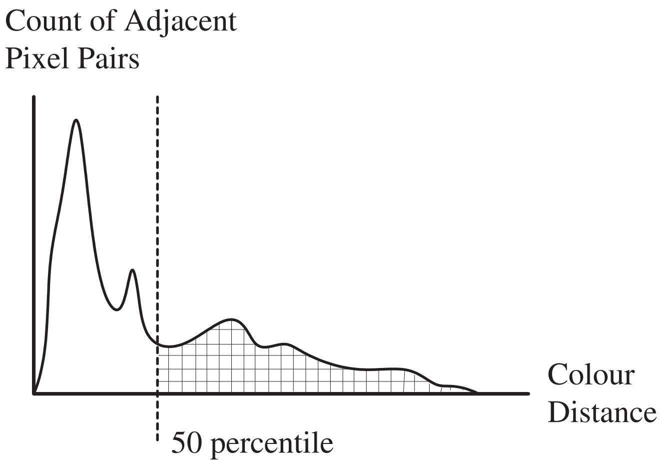
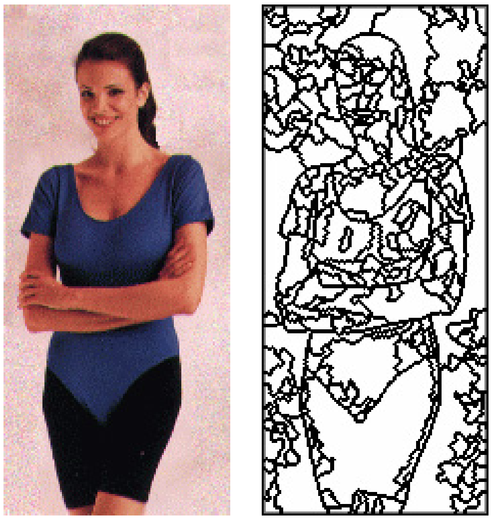


Figure 3.4: Histogram of colour distances between adjacent pixels. The colour distance that separates inner from outer pixel pairs lies in the upper half (50 percentile) given constraints [a] and [b].

In complex images, constraint [b] does not hold most of the time because the ranges of the colour distance of inner pairs and outer pairs tend to overlap. Constraint [b] needs to be relaxed for complex images. The approach taken is to divide the underlying image into many small blocks so that the likelihood of constraint [b] being true for each block is high. The point of separation can then be set at a point above the 50 percentile of the histogram built separately for each of the blocks.

In the actual implementation, the smoothed image is divided into blocks of size $m \times m$ pixels. The block size needs to be small and is determined empirically to be appropriate at 40×40 pixels. The colour distance at the point of separation



(a) (b)

Figure 3.5: Region Segmentation result. (a) The input image. (b) The boundaries of the segmented regions are accurately identified.

used for segmenting the pixels in each block is determined from the individual histogram of the block. It is found that the point of separation, named as Γ_0, can be set as high as 80 percentile and still yield a conservative segmentation (i.e. an over-segmented result).

In OLAG, the pixel grouping threshold percentile Γ_0 is fixed conservatively at 80%. An example of the result of the Region Segmentation step is shown in Figure 3.5. It can be seen that though the image is overly segmented, the boundaries where there are sharp local colour changes are identified accurately.

3.5 Cluster Forming

The purpose of Cluster Forming is to group disjoint regions with similar colour into a cluster of regions and place in the same layer so that they can be considered as being parts of the same object. An object may appear to have disjoint parts because it is occluded by other objects in front of it. In this step, it is necessary to establish decision criteria for grouping disjoint regions. Colour similarity of the regions is one such criterion. Other possible criteria are the proximity and the size of the two regions concerned. Large regions may be located further apart and still be considered as belonging to the same object. Another criterion is the similarity of the two regions' neighbourhood. Two regions are more likely to be parts of the same object if they have similar adjacent regions (i.e. their background).

Colour similarity between two regions is determined by the colour distance (CD) between the regions, and their colours are defined as the mean colours of the pixels in the regions. The threshold for which two regions' colours are considered similar is determined from a colour distance histogram. The histogram is constructed in the same manner as that in Section 3.4 except that it is based on the colour distance between adjacent pixels in the regions and in their neighbourhoods. Regions i and j are similar in colour if $CD_{ij} \leq CD(\Gamma_1)$. The parameter Γ_1 is a percentile value of the histogram. In other words, Γ_1 percent of the colour distances between adjacent pixels in the regions and their neighbourhoods are smaller than or equal to $CD(\Gamma_1)$. This way, attention is drawn to the colour variation in the regions and their surroundings instead of the whole image because different parts of an image may have different degrees of colour variation. This is especially so when the two regions concerned are much smaller than the image itself.

The colour distance histogram is constructed from *regional area* of a region so as to capture enough context of the region and its surrounding. The regional area is defined as the minimum bounding box of the region extended out in all 4 directions by a magnitude equal to the width and height of the minimum bounding box. In other words, the regional area has three times the width and height of the minimum bounding box. Figure 3.6 shows examples of regional areas in shaded boxes. When regions R_1 and R_2 are considered for merging, the histogram is built from the pixels in the bounding box containing the regional areas of R_1 and R_2.

The second criterion, proximity, is defined based on the regional areas as well. For example in Figure 3.6, R_1 and R_2 are considered close enough for merging because their regional areas overlapped. On the other hand, R_1 and R_3 will not be considered for merging as their regional areas do not overlap. Therefore, the proximity criterion is related to region size. The larger the regions, the larger are the regional areas, and the further they can be apart and still be merged together.

The third criterion is the similarity of the neighbourhoods of the two regions, which consist of those regions that are adjacent to the two regions. Consider two regions i and j to be merged. Let N_i denote the set of regions adjacent to region i,

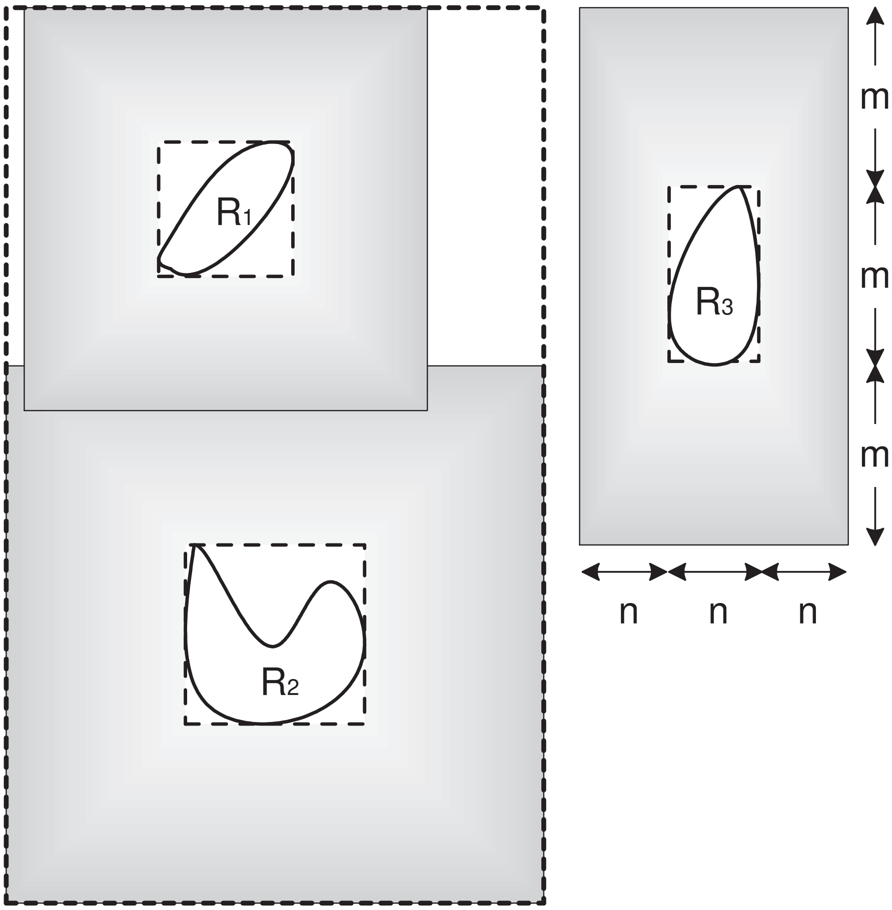


Figure 3.6: Illustration of the regional area for construction of histogram. The bounding boxes (dashed lines) of the regions R_1, R_2 and R_3 are extended outward to form the regional areas (shaded boxes). When R_1 and R_2 are considered for merging, a histogram is constructed from the pixels in the bounding box (bold-dashed lines) containing both the regional areas of R_1 and R_2.

similarly for N_j. Let K_i denote the set of regions such that for each region $r \in K_i$,

(1) $r \in N_i$ and

(2) $\exists$ region $s \in N_j$ such that $CD_{rs} \leq CD(\Gamma_1)$.

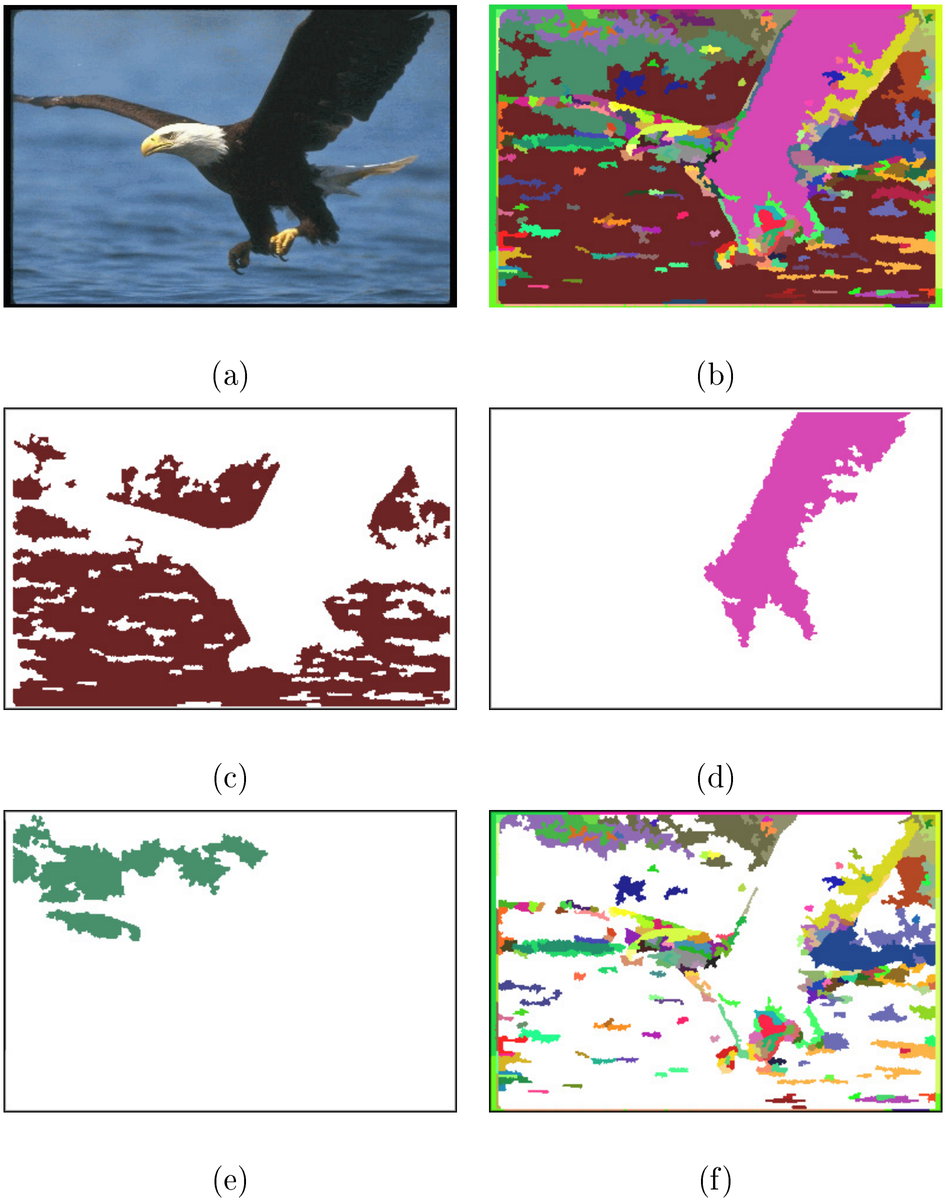
The neighbourhood of regions i and j are considered similar if

$$\frac{\mid K_i \mid}{\mid N_i \mid} \geq 0.5 \quad \text{and} \quad \frac{\mid K_j \mid}{\mid N_j \mid} \geq 0.5.$$

Intuitively, it means that the neighbourhood of two regions are similar if the majority of the regions in one neighbourhood is similar in colour to at least one of the regions in the other neighbourhood.

Before Cluster Forming is actually performed, the input segmented result from the previous step needs to be further polished. Since OLAG focuses on finding significantly large regions, those regions that are too small to be of any significance need to be removed by being absorbed into the larger neighbouring regions. In the current implementation, regions of size 30 pixels or less are merged into the neighbouring regions that are most similar in colour.

Cluster Forming is performed iteratively and in each iteration, neighbouring regions that satisfy the three criteria specified above are grouped into the same cluster. This is performed until no more grouping is possible. An example of region grouping result of the Cluster Forming processing is shown in Figure 3.7.



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

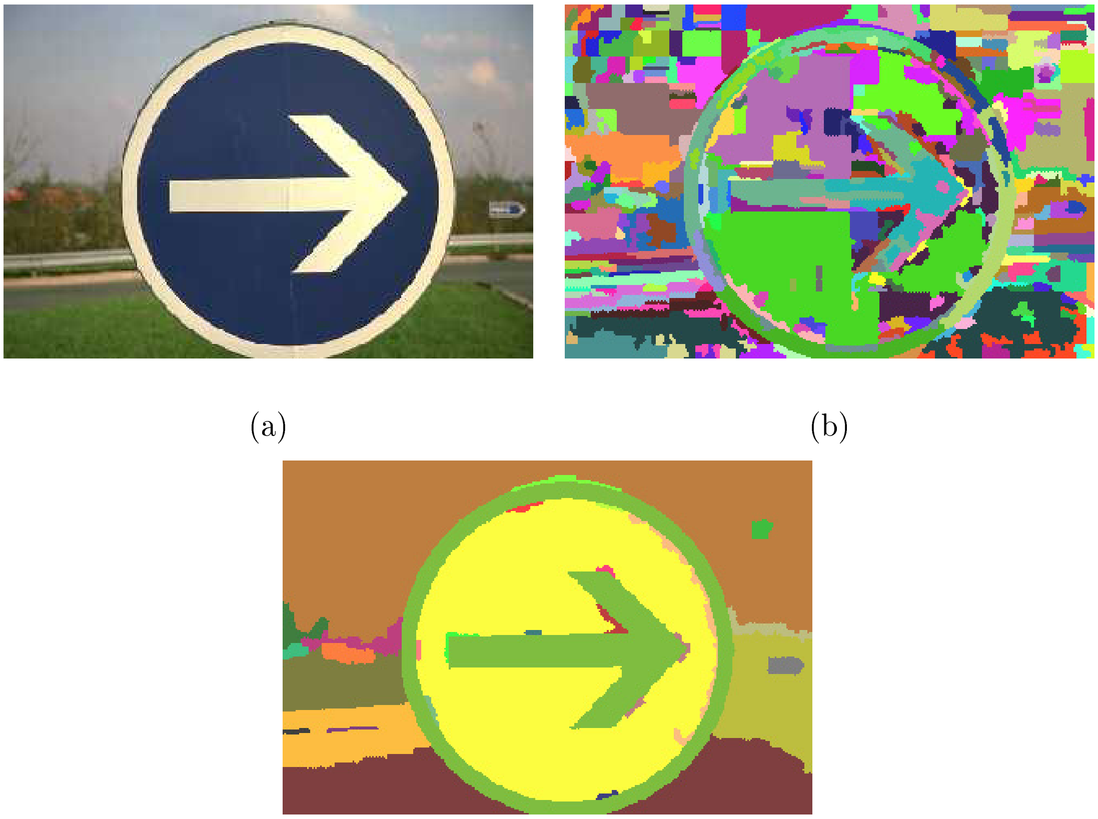
Figure 3.7: Cluster Forming result. (a) The input image. (b) The Cluster Forming result with all the clusters of regions shown in the same image. (c, d, e) The layers containing the three largest clusters of regions (shown in coloured patches). (f) All the remaining clusters shown in the same image. Regions with the same colour belong to the same cluster. Note that the upper and lower parts of the sea (i.e. background), separated by the eagle, are correctly grouped into the same layer.

3.6 Cluster Merging

In this step, adjacent clusters are merged if the distance between the mean colours of the clusters satisfies the colour similarity criterion. The colour similarity measure used here is the same as that used in the previous step. A similar threshold percentile, Γ_2, based on the histogram is used to define the colour similarity threshold. The threshold percentile for Cluster Merging, Γ_2, is larger than the threshold percentile for Cluster Forming, Γ_1. Correspondingly, the colour similarity threshold $CD(\Gamma_2)$ is more relaxed than $CD(\Gamma_1)$. Cluster Merging is performed iteratively with the initial merging threshold set at $CD(\Gamma_1)$. At each iteration, the threshold is further relaxed by raising the percentile by a small increment until $CD(\Gamma_2)$ is reached. An example of Cluster Merging result is shown in Figure 3.8.

3.7 Hole Filtering

In the first 5 steps of OLAG, image segmentation is performed based on colour. From this point onward, non-colour visual properties are used to perform further grouping. In this step, region sizes are examined to determine if they are significant. As described in Chapter 1, interesting objects must have significant sizes and OLAG aims at identifying these objects. A region with an insignificantly small size is referred to as a "hole". A hole is removed by absorbing it into the adjacent region that shares the most contact with the hole. The contact between two adjacent



(a) (b)

(c)

Figure 3.8: Cluster Merging result. (a) The input image. As the merging threshold is relaxed from (b) $\Gamma_1 = 87$ to (c) $\Gamma_2 = 97$, more regions are grouped together with the sky portion merged into almost a single large cluster of regions. Regions with the same colour belong to the same cluster.

regions is defined as the pixel count of the shared boundary. Holes usually exist as fine details within objects or as noise at the boundaries of the objects.

Size is a relative concept. The size of a region is compared against the total size of the surrounding regions. By visual inspection, it is observed that given a large region of size 250×250 pixels, a small region of size 15×15 pixels or less will appear insignificant within the large region. As the enclosing region's size is reduced, the minimum significant size of the enclosed region is reduced exponentially. The following relation is assumed to hold: for a region i whose size is denoted by $\mid i \mid$,

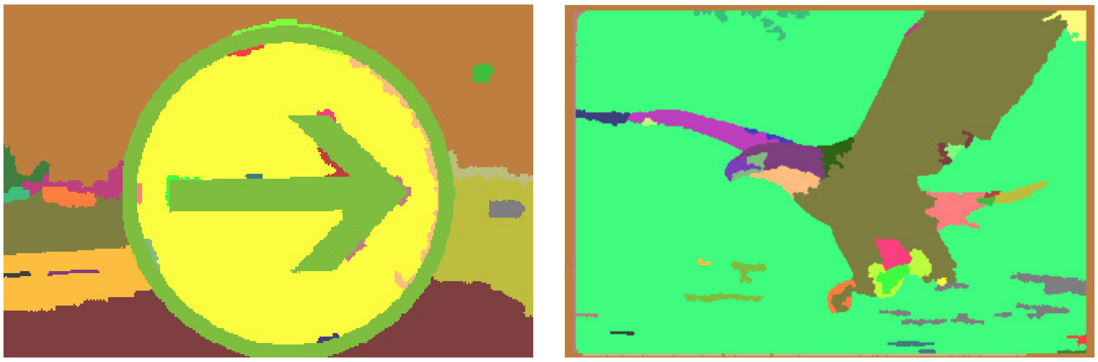
$$\begin{aligned} \frac{\mid i \mid^n}{A_i} &= constant \qquad (3.8) \\ &= \frac{(15 \times 15)^n}{250 \times 250} \\ &= \frac{225^n}{62500} \end{aligned}$$

where A_i is the total size of all the regions surrounding region i, including the size of region i itself. It is determined that a suitable value for n is 4. Hence, a region i is considered to be a hole if

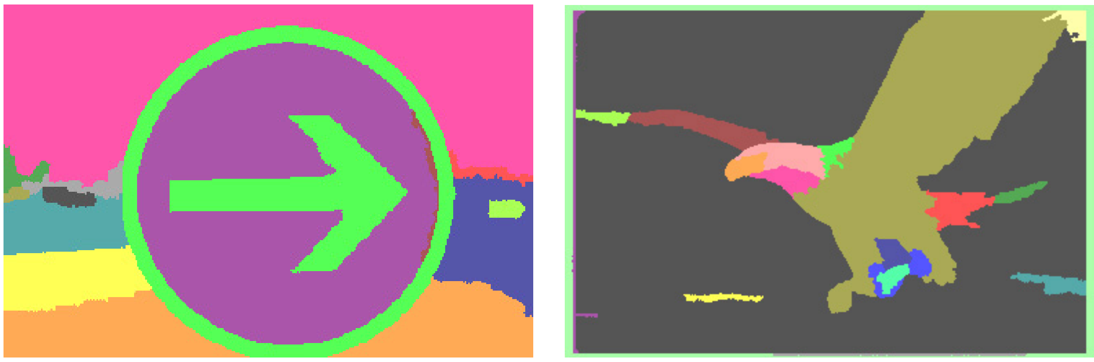
$$\begin{aligned} \mid i \mid &\leq \left(\left(\frac{225^4}{62500} \right) \cdot A_i \right)^{\frac{1}{4}} \\ &\approx 14 \cdot (A_i)^{\frac{1}{4}} \end{aligned}$$

Hole Filtering is an iterative step in which holes are repeatedly seeked and merged into their neighbouring regions as the regions grow larger. Figure 3.9 shows example results in which most of the small regions that contain uninteresting details are merged into larger neighbouring regions.

(a) (b)



(c) (d)



(e) (f)

Figure 3.9: Hole Filtering results. (a, b) The input images. (c, d) Cluster Merging results. (e, f) Hole Filtering results. Regions with the same colour in a segmented image belong to the same cluster.

3.8 Compact Grouping

The OLAG system is designed to facilitate the inclusion of other visual properties that can improve the segmentation result. In the current implementation, the compact grouping step is included to demonstrate the ease of introducing compactness into OLAG. Compactness is a measurement of good shape and is defined as the ratio of the square of the object's perimeter to its area. The smaller the compactness, the more compact is the object. A circle is the most compact shape with a compactness of 4π.

In general, the approach adopted here is to merge two clusters if the compactness of the merged cluster is smaller than those of the individual clusters. The compactness of a cluster is defined as the weighted sum of the compactness of the regions within the cluster. The weights are determined by the respective sizes of the regions. The compactness of a cluster is given by:

$$\begin{aligned} CP &= \sum_i \frac{A_i}{\mid A \mid} \cdot \left(\frac{P_i^2}{A_i}\right) \\ &= \frac{\sum_i P_i^2}{\mid A \mid} \end{aligned} \tag{3.9}$$

where P_i and A_i are the perimeter and the area of region i in the cluster, and $\mid A \mid$ is the area of the cluster.

Clusters Y_i and Y_j can be merged if the compactness of the merged cluster Y_{ij}

is smaller than a merging threshold:

$$CP_{ij} \quad < \quad (1 - \frac{\mid Y_{ij} \mid}{I_m}) \min(CP_i, CP_j) \tag{3.10}$$

where $\mid Y_{ij} \mid$ is the area of the merged cluster Y_{ij} and I_m is the size of the image. That is, the compactness of the merged cluster Y_{ij} must be significantly smaller than those of the clusters Y_i and Y_j. The significance of improving compactness is dependant on the size of the clusters to be merged. The larger the cluster sizes, the more visually obvious are the clusters in the image and hence, the more improvement is required to merge them.

An exception to the Compact Grouping process is that if a cluster contains a region that is fully enclosed within a region of another cluster, then the two clusters are not merged. This exception takes care of the case in which an object is contained in another object, for example, cups on the table and fruits in a basket. The enclosed objects are interesting details that should be retained and not to be merged into the enclosing object.

Note that the Compact Grouping step iterates with the Hole Filtering step (Fig. 3.1) until no more merging is possible. After each Compact Grouping step, Hole Filtering is applied to filter out newly formed holes. In the segmented result shown in Figure 3.10, the Hole Filtering result consists of many clusters which are grouped by this iterative process into the final result. Figure 3.11 shows another result of Compact Grouping in which the sky is segmented as a large region and

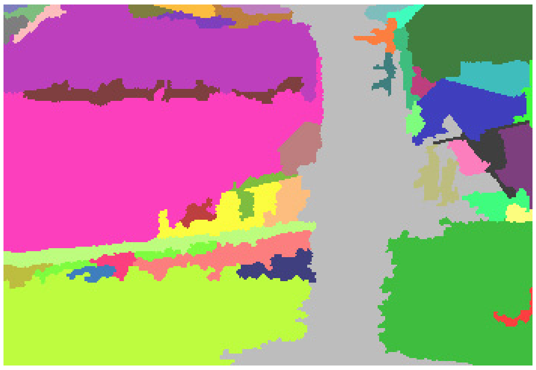
(a)

(b)

(c)

Figure 3.10: Compact Grouping result of an image of a flower. (a) The input image. (b) The result after the first Hole Filtering iteration and (c) the result after the last Compact Grouping iteration. Regions with the same colour belong to the same cluster. The flower petals are merged into a single region by Compact Grouping.

fine details of the ground are grouped together as well.



(a) (b)

(c)

Figure 3.11: Compact Grouping result of an outdoor image. (a) The input image. (b) The result after the first Hole Filtering iteration and (c) the result after the last Compact Grouping iteration. Regions with the same colour belong to the same cluster. The sky is segmented as a single large region and fine details of the ground are grouped together.

3.9 Figure-Ground Inferencing

After the Compact Grouping step, each cluster of regions is placed in a layer. Given these clusters, inference rules are used to infer the figure-ground relationships among the clusters. These inference rules attempt to identify the background of the image, and the enclosure and occlusion relationships among the objects.

1. Background: A background region typically has extensive coverage over the width or the height of the image. Therefore, a cluster with a minimum bounding box that covers the full width or height of the image is identified as a background region in the image.

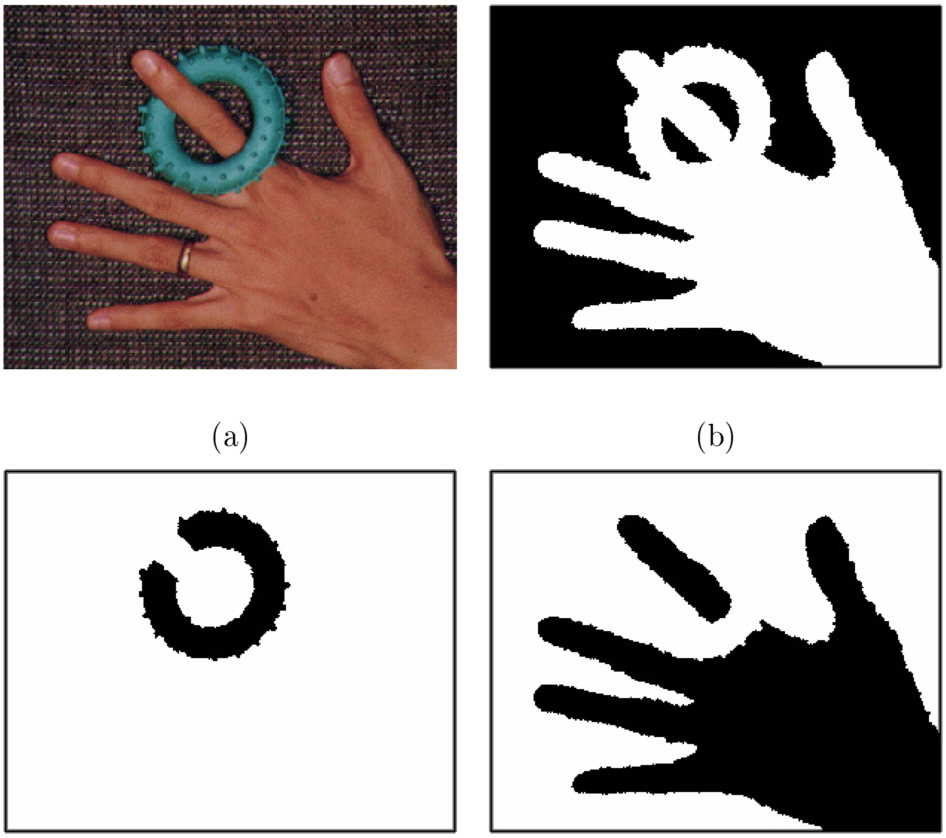
2. Enclosure: A cluster whose regions are fully enclosed within the region of another cluster is said to be contained within the enclosing cluster. Based on the relative size, the smaller cluster is considered to be located in front of the enclosing cluster. This rule conforms to human's figure-ground perception.

3. Occlusion: A cluster may contain two or more regions separated by the regions in another smaller cluster. It can be inferred that the separated cluster is behind or occluded by the smaller cluster. This rule also conforms with human's perception.

In Figure 3.12, the image is segregated into 3 layers corresponding to the textured background, the rubber ring and the hand. Note that the background consists of 3 regions and the second finger is disjoint from the rest of the hand. The textured background has an extensive cover across both the width and height of the image. Thus it can be identified as the background of the image. The hand and the ring are then identified as foreground objects located in front of the background. Further inferencing made on the results shows that the ring occludes the hand and separates it into two disjoint parts (Fig. 3.13). Similarly, the ring and the hand separate the background into disjoint parts. Therefore, it can be concluded that the ring is located in front of the hand and the hand is, in turn, in front of the textured background.

3.10 Summary

OLAG is designed to identify objects in natural images. It encompasses several steps that made use of specific properties in each step to group pixels into regions and clusters of regions. OLAG focuses mainly on making colour comparisons at different level of abstractions, from pixel, region to cluster. At the pixel level, adjacent pixels are merged if they have similar colours. At the region level, disjoint regions are grouped into clusters based on the colour of the regions. Adjacent clusters are then merged by further relaxing the colour similarity criterion. Non-colour property such as region size is used to merge those regions that are small and, therefore, uninteresting. An attempt to include shape property is made by



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

Figure 3.12: Figure-Ground Inferencing. (a) The input image. (b) The background layer. (c, d) The foreground object layers. It is inferred that the ring and the hand are located in front of the background.

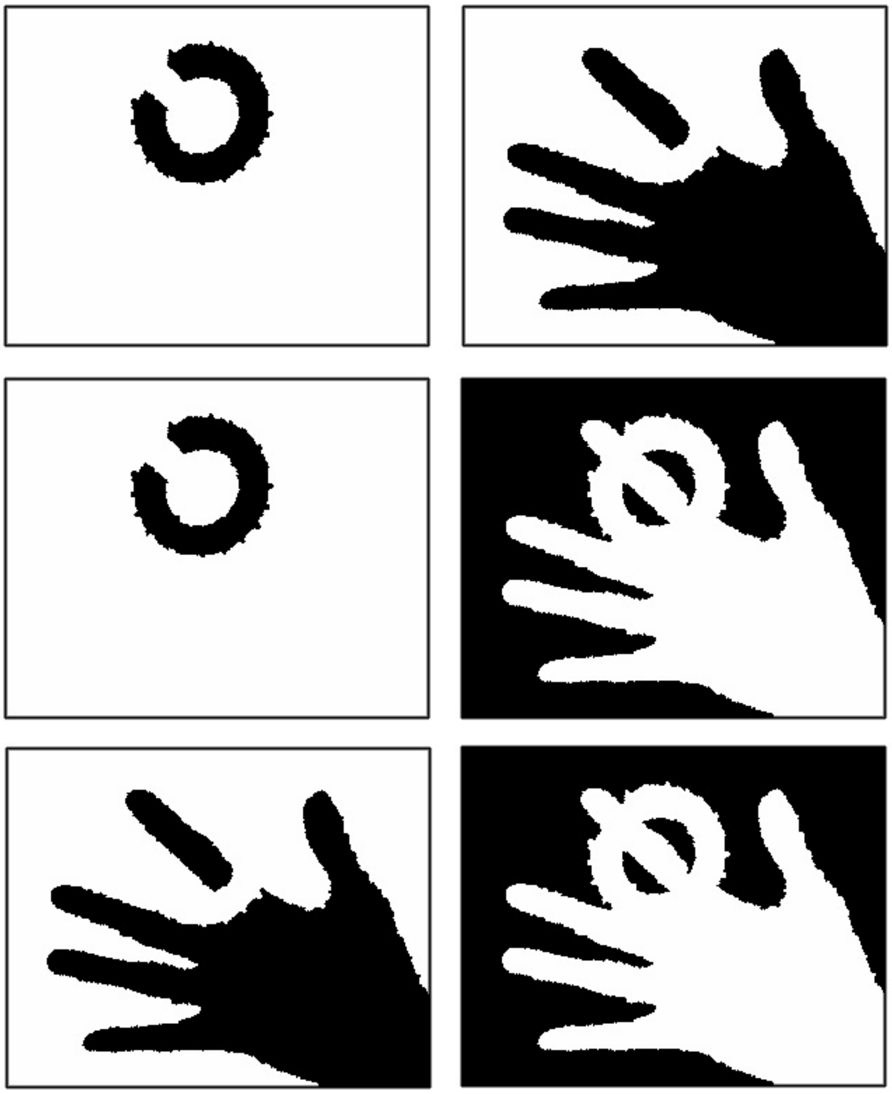


Figure 3.13: Occlusion detection. The object (in black) on the left is in front of (i.e., occludes) the object or region in the images on the right.

considering the compactness of objects in merging the clusters. Inferencing of figure-ground relationship is made using rules that help to identify the background of the image, the occluding and the occluded objects, and the enclosing and the enclosed objects.

A uniform approach is adopted throughout OLAG. Grouping information such as histograms and thresholds are region-based and image-dependant. For example, region histogram thresholding is used to determine the colour similarity in the colour based grouping. The threshold percentiles Γ_1 and Γ_2 are defined in terms of the histograms, and are well bounded: $80 \leq \Gamma_1 < \Gamma_2 < 100$. Empirical study shows that the useful ranges of the parameters are: $80 \leq \Gamma_1 \leq 88$ and $90 \leq \Gamma_2 \leq 98$. For the segmented results shown in this thesis, the parameters are set automatically though fine tuning of the parameters is possible to obtain results that are more suitable for different applications. The rules for setting the parameters are described in the next chapter together with comparisons of OLAG's segmentation results and those of other existing methods.

Chapter 4

Segmentation Results

4.1 Experimental Settings

The segmentation results of OLAG presented in this thesis are fully automated with the parameter settings determined by a selection scheme. The selection scheme classifies all images based on the smoothness measure Θ of the images after smoothing (Section 3.3). Images are classified into 3 categories, each with a different set of parameter values (Table 4.1), which are selected based on the following guideline. In a very smoothed image, the ranges of the colour distance of inner pixel pairs and outer pixel pairs do not overlap significantly (Section 3.4). Hence, a high thresholding percentile can be used to separate the inner and outer pixel pairs. The

Table 4.1: Selection of values of threshold percentiles Γ_1 and Γ_2 based on the smoothness measure Θ.

Smoothness measure Θ	Cluster Forming percentile Γ_1	Cluster Merging percentile Γ_2
≤ 3.2	87	97
> 3.2 and ≤ 3.7	85	95
> 3.7	80	90

difference between the percentiles Γ_1 and Γ_2 should be around 10 percent.

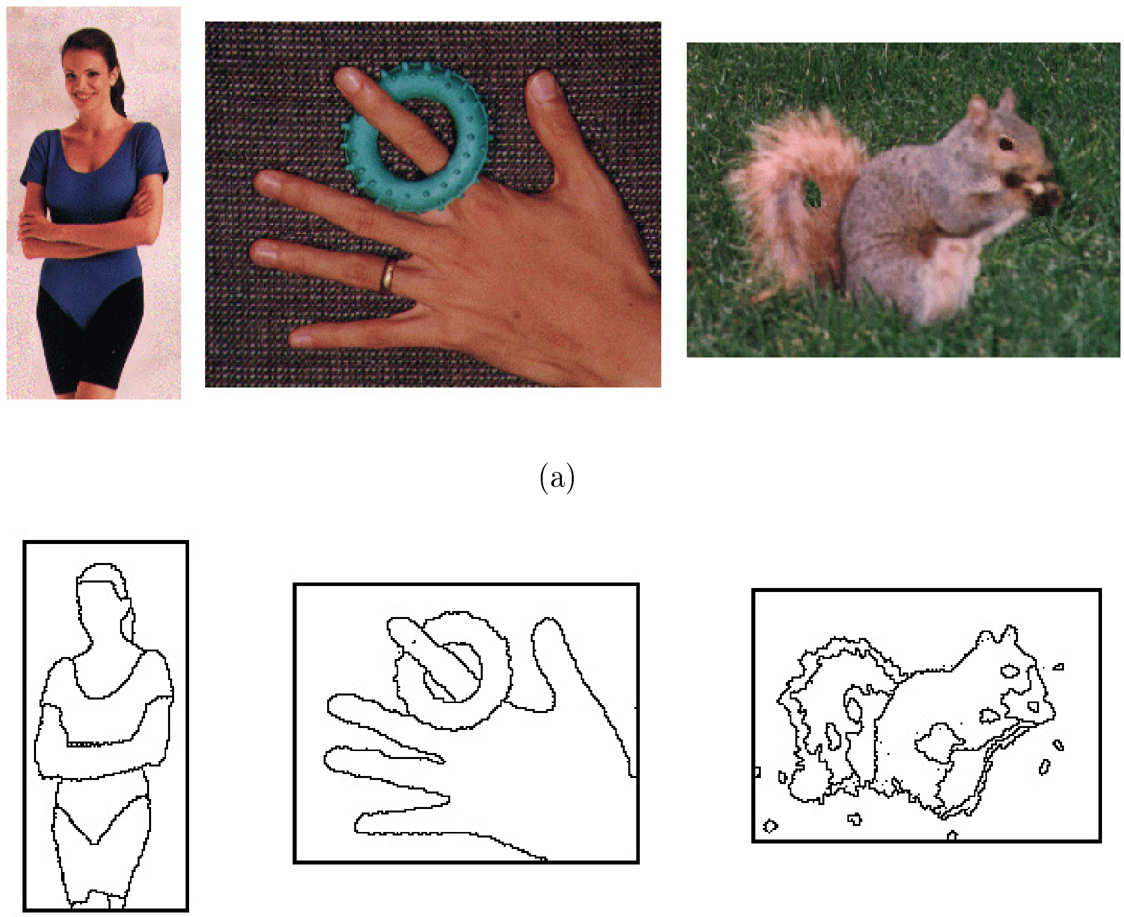
In the following, segmentation results of OLAG are compared against results of two existing methods, namely Region Competition and JSEG colour segmentation. These existing approaches have be described in Section 2.2.1. Note that the segmentation results from Region Competition is dependant on the initial number of seeds and the number of segmentation iterations. On the other hand, JSEG colour segmentation is similar to OLAG in that the parameter settings are determined internally by the system based on the input image.

4.2 Comparison of Segmentation Results

Figure 4.1 shows the results of applying OLAG, Region Competition and JSEG to segment images of various objects. Their results are similar in most cases whereby large regions have all been identified correctly. Region Competition produces more fragmented results for the image of the squirrel while JSEG over-merges part of the head portion of the squirrel with the background. Moreover, JSEG's result identifies most of the nail portions of the hand image as separate regions. In terms of colour segmentation, OLAG is comparable to the two existing methods. On top of that, OLAG also identifies the disjoint parts of the objects. For example, the face and the arms of the woman, the separated portions of her garments, as well as her hair are all identified correctly as objects containing disjoint parts.

Results of segmenting more complex images are shown in Figures 4.2 and 4.3. Only the results of OLAG and JSEG are shown because the results of Region Competition are not available for these images. For the image of the outdoor scene (Fig. 4.2), OLAG produces better grouping results with larger regions being formed than JSEG. Note that though OLAG has not been designed to take into consideration colour texture, it can handle colour texture quite well. For example, the flower texture on the ground is well segmented into large regions. Insignificant details such as the small trees in the far background and the lamp pole are removed in OLAG's segmentation result.

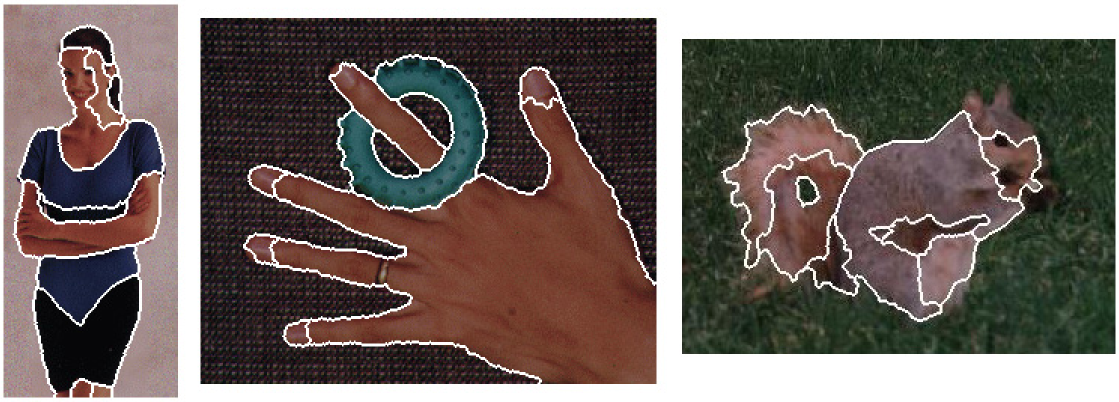
In Figure 4.3, the lizard's body colour is close to that of the background. As a result, over-merging occurs for both OLAG and JSEG. However, OLAG gives a better composition of the shape of the lizard. On the other hand, JSEG gives a cleaner segmentation result with fewer regions being identified.



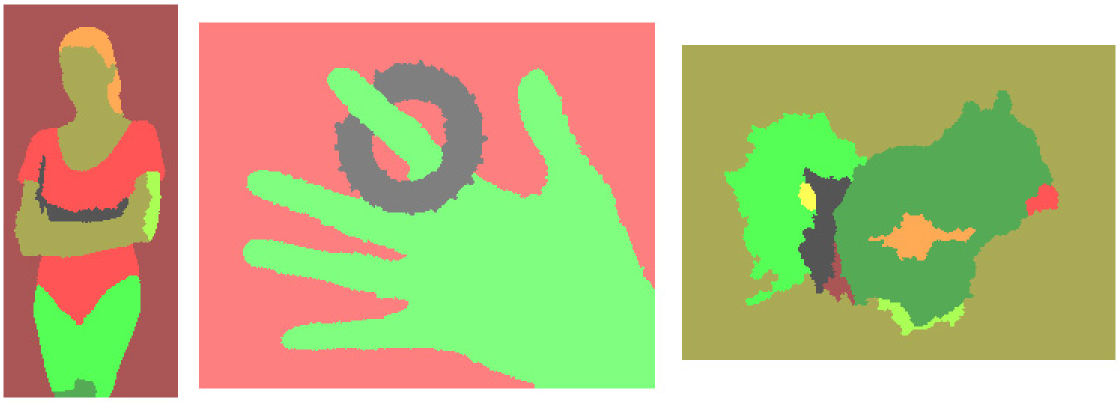
(a)

(b)

Figure 4.1: (a) The input images. (b) Segmentation results of the Region Competition method.



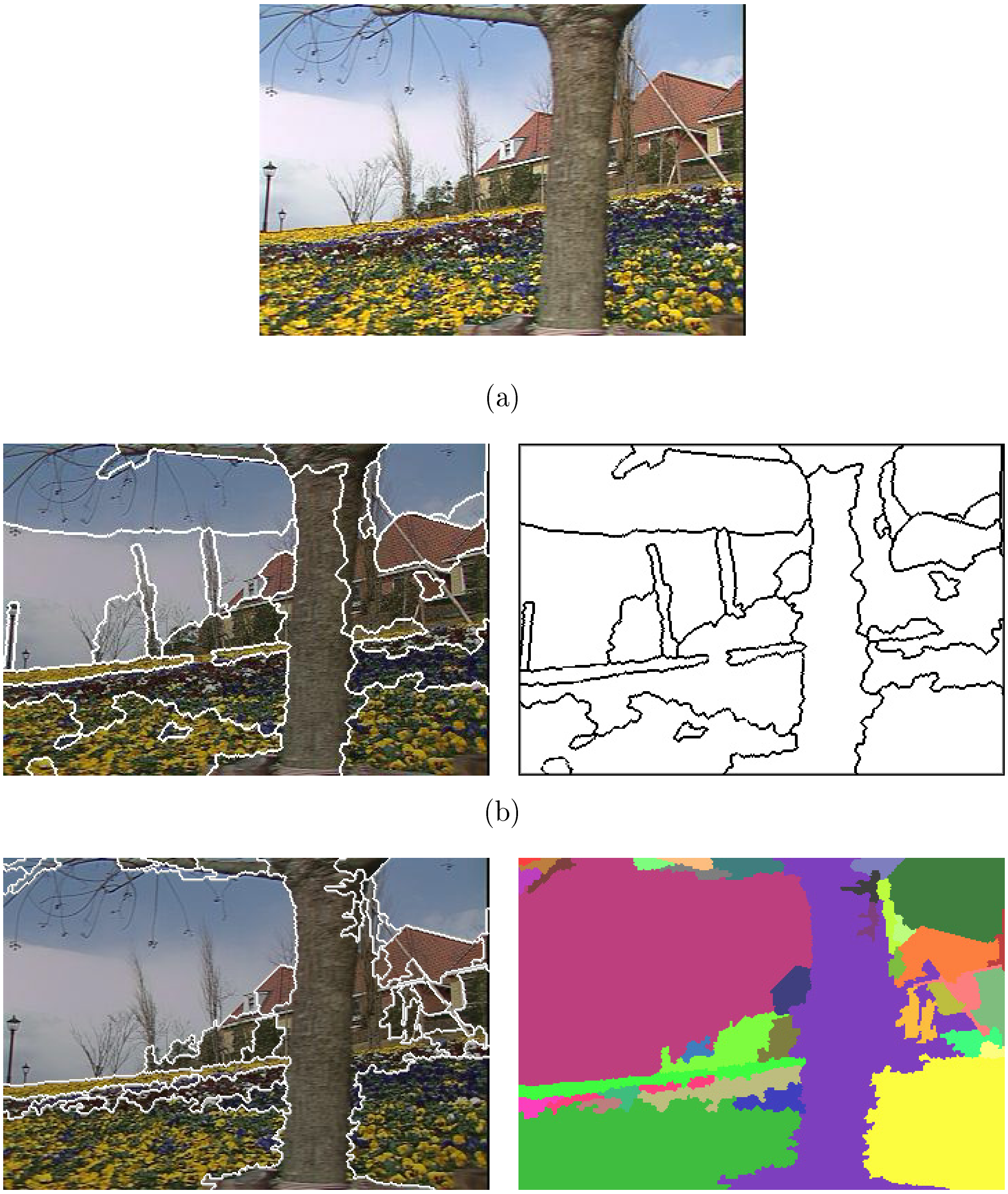
(c)



(d)

Figure 4.1 (continued): (c) Segmentation results of JSEG. (d) In the results of OLAG, regions with similar colours belong the same clusters (i.e., objects).

Figure 4.4 shows another comparison between the results of OLAG and JSEG. Figure 4.4 (c) and (d) are obtained with OLAG running in the default mode with parameter values set by OLAG itself. It can be seen that with default parameter values, JSEG performed better than OLAG. In OLAG's result, the hands are merged into the background which has a similar colour. An adjustment of OLAG's parameter values produced a better segmentation result with the hands recovered (Fig. 4.4(e)). The result shows that OLAG was able to retain fine distinct details accurately, for instance the pingpong ball and the border of table. In addition, OLAG produced a more accurate outline of the regions' boundaries.



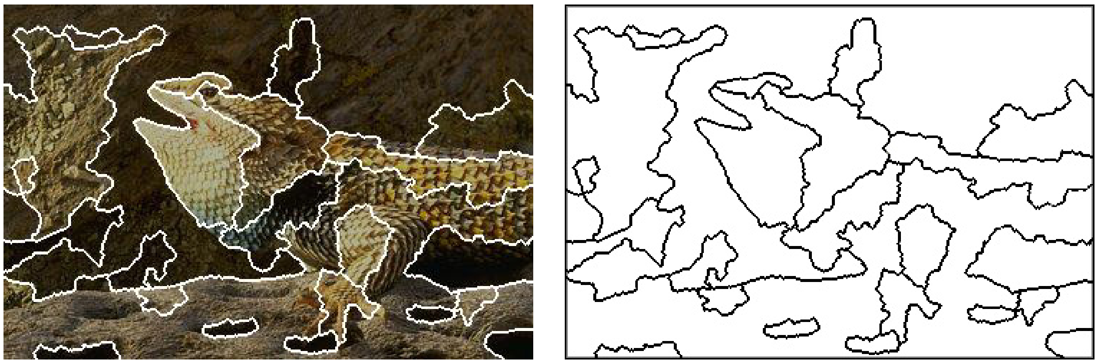
(a)

(b)

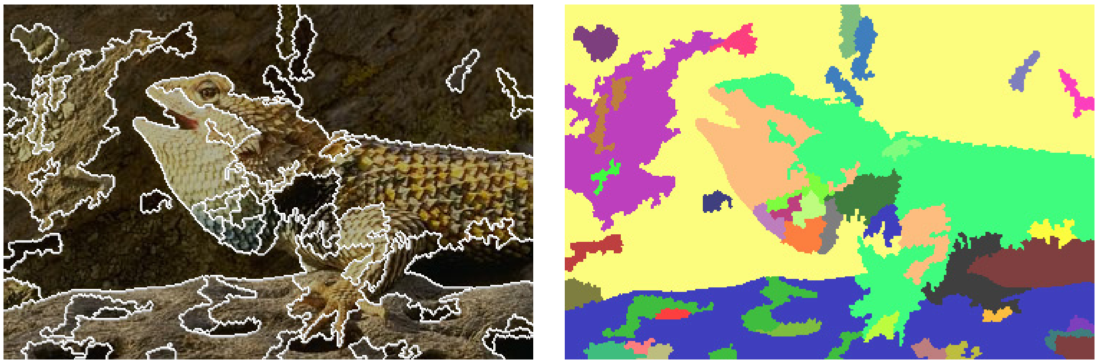
(c)

Figure 4.2: Segmentation result of an outdoor image. (a) The input image. (b) The result of JSEG. (c) The result of OLAG contains larger regions than does JSEG's result.

(a)

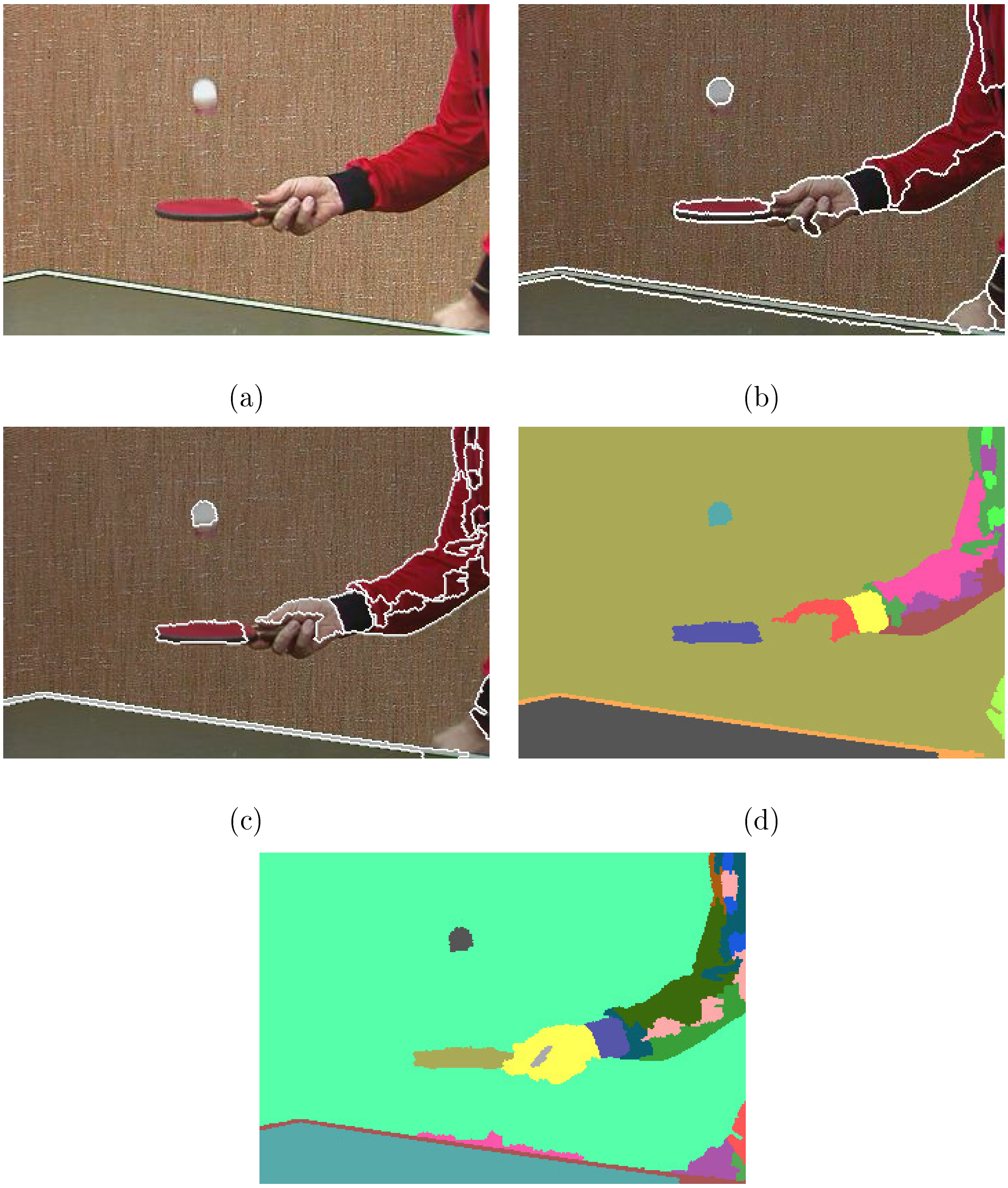


(b)



(c)

Figure 4.3: Segmentation result of an image of a lizard. (a) The input image. (b) The result of JSEG and (c) the result of OLAG.



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

Figure 4.4: Segmentation result of an image of table tennis. (a) The input image. (b) The result of JSEG. (c, d) The result of OLAG with $\Gamma_1 = 87$ and $\Gamma_2 = 97$. (e) The result of OLAG is improved with $\Gamma_1 = 87$ and $\Gamma_2 = 96$.

Chapter 5

Conclusion and Future Work

5.1 Future Works

In the current implementation, OLAG makes use of colour, size and compactness to perform image segmentation. There exists many other properties including boundary continuity, texture and shape that are useful for identifying objects in an image.

Boundary continuity can be used to group clusters together if there is continuation of boundary at the joint of adjacent clusters. Inclusion of boundary continuity in OLAG should improve its segmentation result. One aspect of shape is compactness which has already been incorporated into OLAG. Symmetry is another aspect of shape that can be used. Texture can be used to group clusters that overlap a lot if there is evidence that the combined cluster forms a textured pattern. Texture information can also be used in the Cluster Merging step to select candidate clusters for merging. Currently, Cluster Merging is performed based on the mean colour of the clusters alone. Variation in colour in the cluster is not taken into consideration and hence could mistakenly merge clusters that are spatially close and similar in overall colour but differs in texture pattern.

The selection scheme used to determine suitable values for the threshold percentiles Γ_1 and Γ_2 (Chap. 4) could be fine-tuned to give better segmentation results. Currently the scheme classifies images broadly into 3 categories based on the smoothness measure of the smoothed images. For each category, a different set of parameter values is used. An example-based machine learning approach can be considered to determine more specific parameters to be used for each image category.

5.2 Contributions

One of the main contributions of this Masters research is the building of an image segmentation framework that includes figure-ground segregation. Depth relationships among the objects can be inferred by OLAG. It is easy to include additional visual properties into the framework to improve the segmentation results. The inclusion of size and compactness exemplifies the flexibility of the OLAG framework. Size and compactness help OLAG in grouping insignificant regions into neighbouring regions.

Another contribution is the grouping of disjoint regions. Existing clustering algorithms are able to perform disjoint grouping but they have not been applied specifically to colour segmentation. Another important feature of OLAG is that colour similarity measurement is neither fixed nor dependant on the global colour information of the image. Instead, it is dependant on the colour distribution of the neighbourhood surrounding a region. This method can thus adapt to regional colour variation of the image.

5.3 Conclusion

The objective of the OLAG system is to perform colour segmentation and figure-ground segregation of natural scene images. The segregated results show that OLAG can not only group disjoint regions that belong to an object together, but can also infer foreground-background relationships between the objects. The range of parameter values is tightly bounded and automation of the parameter value selection is possible.

Computationally, OLAG is fast enough to be executed on most machines. The computational time required depends on the size of the image as well as the complexity of the image. In general, it takes less than 10 minutes to process an image of about 400×300 pixels in size on a laptop (P266MMX Intel chip, 32MB SDRAM and Windows 98 operation system). Therefore, OLAG can be used in many applications that require image segmentation and figure-ground segregation.

Bibliography

[1] R. Adams and L. Bischof. Seeded region growing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16(6):641–647, 1994.

[2] I. Biederman. Recognition-by-components: A theory of human image understanding. *Psychological Review*, vol. 94(2):115–147, 1987.

[3] J.F. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8(6):679–698, 1986.

[4] M. Celenk. A color clustering technique for image segmentation. *Computer Vision, Graphics, and Image Processing*, vol. 52:145–170, 1990.

[5] L.D. Cohen. On active contour models and balloons. *CVGIP: Image Understanding*, vol. 53(2):211–218, 1991.

[6] L.S. Davis and A. Rosenfeld. Noise cleaning by iterated local averaging. *IEEE Transactions on Systems, Man and Cybernetics*, vol. 8:705–710, 1978.

[7] Y. Deng, B.S. Manjunath, and H. Shin. Color image segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1999.

[8] H.R.W. Gainer. *Measuring Colour*. Ellis Horwood series in Applied Science and Industrial Technology. New York: Ellis Horwood, 2nd edition, 1991.

[9] S. Grossberg. A solution of the figure-ground problem for biological vision. *Neural Nerworks*, vol. 6(4):463–483, 1993.

[10] S. Grossberg. Figure-ground separation. In M. A. Arbib, editor, *The Handbook of Brain Theory and Neural Networks*, pages 395–399. The MIT Press, 1995.

[11] L. Hérault and R. Horaud. Figure-ground discrimination: A combinatorial optimization approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15(9):899–914, 1993.

[12] D. Harwood, M. Subbarao, H. Hakalahti, and L.S. Davis. A new class of edge-preserving smoothing filters. Technical Report CAR TR-59, Center for Automation Research, University of Maryland, 1984.

[13] Y. Hirakura, Y. Yamaguchi, H. Shimizu, and S. Nagai. Dynamic linking among neural oscillators leads to flexible pattern recognition with figure-ground separation. *Neural Networks*, vol. 9(2):189–209, 1996.

[14] G.A. Mastin. Adaptive filters for digital image noise smoothing: An evaluation. *Computer Vision, Graphics, and Image Processing*, vol. 31:103–121, 1985.

[15] Y. Ohta. *Knowledge-based Interpretation of Outdoor Natural Color Scenes.* Pitman Advanced Publishing Program, Boston, Massachusetts, 1985.

[16] M. Pietikäinen and D. Harwood. Segmentation of colour images using edge-preserving filters. In V. Cappellini and R. Marconi, editors, *Advances in Image Processing and Pattern Recognition*, pages 94–99. North-Holland, 1986.

[17] V. Rehrmann and L. Priese. Fast and robust segmentation of natural color scenes. Technical report, Computer Science Department, University of Koblenz-Landau, 1997. http://www.uni-koblenz.de/~lb/lb_publications.e.html.

[18] V. Rehrmann and L. Priese. Fast and robust segmentation of natural color scenes. In *Asian Conference on Computer Vision*, 1998.

[19] P. Saint-Marc, J.S. Chen, and G. Medioni. Adaptive smoothing: A general tool for early vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13(6):514–529, 1991.

[20] R. Schettini. A segmentation algorithm for color images. *Pattern Recognition Letters*, vol. 14:499–506, 1993.

[21] T. Uchiyama and M.A. Arbib. Color image segmentation using competitve learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16(12):1197–1206, 1994.

[22] D.C.C. Wang, A.H. Vagnucci, and C.C. Li. Gradient inverse weighted smoothing scheme and the evaluation of its performance. *Computing Graphics Image Processing*, vol. 15:167–181, 1981.

[23] S.C. Zhu and A. Yuille. Region competition: Unifying snakes, region growing and Bayes/MDL for multiband image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18(9):884–900, 1996.