Colour Object Search

P. A. Walcott

Submitted for the Degree of

Doctor of Philosophy

from the

City University

London

Centre for Information Engineering

Department of Electrical, Electronic and

Information Engineering

City University

London, EC1V 0HB, UK

July 1998

Abstract

The visual search process is required when locating an object in some region of space. To perform this search two capabilities must be available: the ability to recognise the object when it comes into view; and a way of selecting these views. Visual search is often complicated by object occlusion and low spatial resolutions of the object. Although the human visual system performs this task effortlessly, the mechanisms of it are not properly understood. Object colour and geometry, however do play an important role. This thesis develops an object search methodology which assumes that a computer vision system captures both wide-angle and zoomed images of the scene containing the object. Since most of the research has focused on object recognition using geometry, this system is purely colour-based. It is not expected that object colour will always give a definitive solution, however database pruning will often occur leading to reduced search times.

The thesis argues that because colour is salient and more resilient than geometry to decreases in spatial resolution, it is more appropriate for visual search when the object occupies a small spatial resolution in an image with a large field of view. It also demonstrates that colour can be used to recognise objects when they occupy most of the field of view; as well as discriminate between database models with similar colour proportions but different region topologies. These conclusions are supported by the results produced by three algorithms, two of which perform colour object search and one that performs colour object recognition.

The first object search algorithm uses image locations containing salient object colours as a method of selecting views. Each of these views are ranked indicating which view most likely contains the object. The second object search algorithm identifies image regions with similar colour and topology as the object. These results are produced in a best-first order. The object recognition algorithm uses an invariant based on region area to identify three

corresponding model and image regions. A transformation is calculated to bring the model and object into the same viewpoint where region matches are based on position and colour.

Each of these methods produced good results in complex indoor scenes with fluorescence and/or tungsten filament lighting; also the search speeds were impressive.

Acknowledgements

There is but **one** who guides the willing, strengthens the weak and provides inspiration and ideas to the starving. I therefore praise His name because He has been my true source of the "light."

I would also like to thank Dr. Tim Ellis for all the help he gave me while working on my PhD; also the Commonwealth Scholarship Commission for providing the scholarship under which this work has been done. In addition, I would like to thank my colleagues Sheun, Brian, James and Dan for their help and encouragement and secretaries Anita, Joan and Linda for all their help. Special thanks to my mum and brothers who actually believed that I could do it all; Also, to the members of the Christian Union and Life City Church for their prayers. Thanks as well to my friends Urvashi, Theodora (Theo), Harriet, Anna, Lucy and Hannah. And last, but certainly not least, I would like to thank Laureen whose love and encouragement have helped me so much during my trying times in London while doing this degree.

Contents

Chapter 1

Using Colour in Object Search

1.1 Introduction

The visual search process is required when locating an object in some region of space. To perform this search two capabilities must be available: the ability to recognise the object when it comes into view; and a way of selecting these views. Visual search is often complicated by object occlusion and low spatial resolutions of the object. Although the human visual system performs this task effortlessly, the mechanisms of it are not properly understood. Object colour and geometry, however do play an important role. This thesis explores the problem of colour object search using a computer vision system and presents a colour object search/recognition methodology for efficient object search.

Although most of the research on object recognition uses object geometry (where the object occupies most of the field of view), object colour is better for object search — when the object is at a reduced spatial resolution. Object colour, which is region-based, is both salient and resistant to reduced spatial resolutions, unlike geometric primitives (e.g. lines) which are not well-defined at these resolutions. Since object colour is region-based, the object being represented must be partitioned into regions of constant reflectance. These regions must be identifiable at the reduced spatial resolutions if object search is to be successful. Objects containing textured regions are often difficult to model especially when the texture is both regular and irregular. Often it is appropriate to represent the object with a few of its non-textured regions; these regions may be disjoint. This affects the object representation used, for example a model based on region adjacency could not be used.

The most straightforward way to search for an object is to use linear search. In linear search all image regions are examined at high spatial resolution. If a geometric object recognition algorithm were used then it would have to search each view for the object. This process is computationally expensive. To reduce this search more efficient methods of selecting the views are required. In this work colour is used. Given an image containing an object at a reduced spatial resolution, image locations which contain salient object colours are identified (cues) — object localisation. Each of these cues is ranked in best-first order so that the view which most likely contains the object is examined first. This search strategy improves the overall search time. Two things must be ensured however: that the searcher does not produce a negative result when the object is actually in the image (false negative results); and that the number of times the searcher returns a positive result when the object is actually not present (false positive results) is kept to a minimum. What is also important is that these cues are generated quickly.

In recent years colour has become increasingly important in object recognition, object search and image retrieval. Although the colour of an object is often not unique, it can discriminate between objects of different colour [Swa90][SO95] — however geometric object recognition is required when the proportion and topology of the colour of objects under consideration are similar. Although colour is an attractive visual cue (it is both salient and stable), it was not exploited in object search and recognition until Wixson and Ballard [WB89] and Swain [Swa90]. The main reason for the slow progress in colour research is the lack of adequate and practical colour constancy algorithms — which are required when models and images are viewed under different illuminants. Recently, however many researchers [SB90] [Sye92] [SO95] [Sch94] [Mat96] [VM96a] [VM96b]. have simply ignored the problem by assuming that model and images are viewed under the same (or similar) illuminants.

The aim of this thesis is to develop a colour object search methodology which satisfies the following conditions:

1. The computational complexity of object search does not appreciably increase with an increase in the number of database models.
2. Object search is allowed in images where objects are (at most 50%) occluded.
3. Objects may be perspectively distorted (perspective, affine, shear, rotation, scale, translation).
4. Only indoor environments are considered with fluorescent and tungsten filament lighting.
5. Objects occupying low spatial resolutions are found accurately.
6. Both 2- and 3-dimensional objects are represented.

Currently, no object search algorithm (to the author's knowledge) performs well in all six of these conditions. Most researchers, for example ignore 4 all together and assume that both models and images are viewed under the same illuminant (e.g. [Mat96]). Also, since most object search algorithms are model-based (the search complexity is proportional to the number of models in the database), 1 is often not satisfied either.

In the next section the use of colour as a visual cue is discussed. Section 1.3 describes the object recognition problem and some common geometric solutions. Finally, Section 1.4 outlines the organisation of this thesis.

1.2 Colour and Object Search

Visual cues are features which draw attention to the object in a scene. For example locating a large table in a room requires only a detailed search of large objects. In this example size (a geometric cue) is being used as the visual cue. The cue which is of great importance to this thesis is colour, which is a fundamental property of objects and is useful in their identification [Hil86].

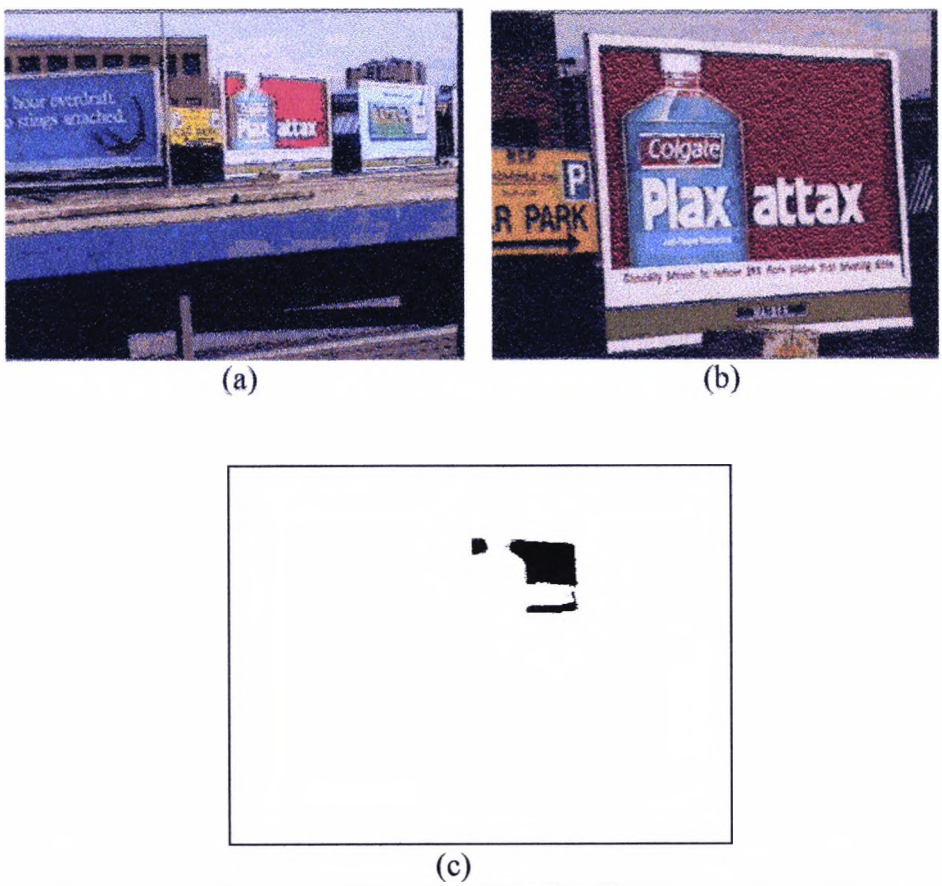
The use of colour as a visual cue is advocated by many researchers including Hilbert [Hil86], Healey and Binford [HB87] and Swain and Ballard [SB90]. Hilbert effectively captures the importance of colour in the object search process in the statement *"Of the properties objects of experience can appear to possess, color is the most salient. Everything we see is seen as having some color and the colors of objects play an important role in our abilities to visually identify and discriminate them."*[1] The colour of an object can be seen as having a number of additional properties which are also important, these are: the persistence of colour over time (important in colour object tracking), and the stability of colour [Hil86] [HB87]. Healey and Binford also argue that regions are more stable than geometric features (such as line segments) under reduced resolution and that the normalised colour of an object is more stable than image irradiance values [HB87].

Grimson [Gri86] describes the problem of object search and recognition as: *"It is usually convenient to pose the problem (identification of objects from sensory data) as one of search; that is, given a set of known models, we identify and locate the particular object that we are sensing by searching a large space of possible solutions until we find one (or all solutions) that matches the information available to us from the sensors."* Typically, the number of possible solutions is enormous; therefore methods to reduce the search space are necessary. From this definition of object search and recognition and the arguments put forward by Hilbert, Healey and Binford, one would admit that colour could indeed be used to reduce the search space.

The effectiveness of colour as a visual cue is demonstrated in Figure 1.1. The aim of this exercise is to locate the Colgate Plax billboard in the real world scene illustrated in Figure 1.1(a). By considering only red regions, three regions in the scene are identified (c.f. Figure 1.1(c)). These are effectively areas of interest which must be explored at a higher resolution. Since these

[1] Hilbert, D., R., "Color and Color Perception: A Study in Anthropocentric Realism", Center for the Study of Language and Information (CSLI), 1986, pp. 2.

regions are spatially close then a single high resolution image is required (c.f. Figure 1.1(b)) on which recognition of the billboard is performed. In this example colour reduced the search space by approximately seven-eighths. In the ideal case the number of areas of interest produced by a visual cue is one, however generally more than one might be produced; however, this does not reduce the usefulness of colour.



(a) (b)

(c)

Figure 1.1: (a) A real world scene containing billboards. (b) The Colgate Plax billboard at high resolution resulting from a zooming process. (c) The red regions of (a) serve as areas of interest.

Recently Syeda [Sye92], Hachimura [Hac96] and Swain [Swa90] have described object search algorithms based on colour saliency. Syeda defined two types of saliency, relative-saliency and self-saliency; relative saliency measures the distinctiveness of a region with respect to surrounding regions; while self-saliency determines the *conspicuousness* of a region by itself, assuming some intrinsic characteristic — e.g. colour and size. Conversely, Hachimura described the *conspicuousness* of colour regions as regions which

have bright and brilliant colours, with large relative areas; the region shape is also included in this definition given that massive (large and compact) regions may be more *conspicuous* than elongated regions. On the other hand, Swain defined salient colours as those colours which are unique to an object thus distinguishing it from other objects in a database.

Both Swain's and Hachimura's definition of saliency (and *conspicuousness*) seem to be a subset of Syeda's self-saliency. Syeda's research however although much more general than Swain's and Hachimura's (Swain performed saliency experiments on white T-shirts with small coloured logos while Hachimura's work was restricted to the recognition of paintings) utilises a complicated saliency cost function which makes its use unappealing. Syeda's definitions, however are instructive.

1.3 Object Recognition

The goal of an object recognition system is to interpret sensory data in order to determine the location (and often orientation) of the object. No single recognition system is appropriate for all kinds of problems; therefore the complexity of the system is often proportional to the difficulty of the problem.

Object recognition is complicated by:

1. the number and complexity of the objects in the scene.
2. the number of objects in the database.
3. the amount of a priori information available about the scene.
4. the amount of object occlusion.

An object recognition algorithm should satisfy the following conditions:

1. gracefully degrade with increased noise.
2. identification should still take place in sparse data due to noise, occlusion and sensor sparseness.

3. should control the combinatoric explosion inherent in the search process.

A considerable amount of research has been done on object recognition, however geometric techniques have dominated. These techniques include the hough transform [Hou62], alignment [Ull86] and geometric hashing [GG92] which are discussed; but also more recently: geometric invariance [MZ92], surface pattern matching [ED94] and interpretation trees [Gri90].

The hough transform, due to Hough [Hou62], is a method for detecting curves by exploiting the duality between the parameters of the curves and the points on the curve. The transform maps feature points (in the image space) into feature space where concentrations of points reveal potential features of interest. The hough transform has been generalised to detect arbitrary non-analytic shapes [BB82] [Bal81] and several methods have been proposed to improve its efficiency by reducing the size of the parameter space (e.g. [Tho92]).

The alignment technique, due to Ullman [Ull86] searches the model and image for anchor points and calculates the viewpoint transformation to bring them into correspondence. By comparing the model and image in this canonical orientation the match is determined. Geometric hashing [GG92] [LW88] is a robust object recognition method which matches local features of objects. In geometric hashing, objects are represented by a set of points, called interest points, together with their geometric relation. The same interest points must be extracted from the scene which requires a search for a compatible transformation which maps the set of points representing the model into image points. Finding this transformation determines the position and pose of the object.

Many object recognition algorithms utilise a model-based, rather than a data-based object recognition philosophy. The difference between these philosophies is that model-based algorithms search the image for features that

match the models under consideration, while data-based algorithms combines groups of image features (e.g. all pairs of regions, all triplets etc.) to determine the presence of database models. The computation expense of the model-based approach tends to be a function of the number of models in the database (typically, the number of models times the complexity of the algorithm for a single model). Alternatively, the data-based approach does not vary (significantly) with the number of models but is of exponential order in complexity. Any algorithm which is based on the data-based approach must utilise techniques which prevent this combinatoric explosion.

The discussion of colour object recognition algorithms is deferred to Chapter 2. Most of these algorithms are model-based except (most notably) for Syeda [Sye92] who presents a colour saliency data-based object recognition algorithm. However, as described earlier the cost function used in that work is complicated, therefore difficult to apply to other scenarios.

1.4 Thesis Outline

In this section the organisation of subsequent chapters of this thesis is presented. In Chapter 2 several popular colour object recognition algorithms are reviewed. The algorithms discussed — which are colour histogram-based, statistical-based, and region-based — show the differences and limitations of the approaches. The colour histogram approach which is commonly used in image retrieval suffers from instabilities and is essentially 2-dimensional. The performance of algorithms based on the statistics of colour spaces (mean, variance, skewness and kurtosis) are also unstable in the presence of background clutter and are therefore inappropriate for object search. Conversely, region-based methods are much more suitable for modelling 3D objects. Also presented in Chapter 2 is the colour image segmentation and colour constancy problems and some common constrained solutions.

In Chapter 3 a colour object search algorithm which locates both 2- and 3-dimensional, planar, rigid objects which are affine distorted in the scene is

presented. In this algorithm areas of interest are generated from salient model colours. Using region size information provided by the cue a minimum and maximum object size is determined. Each area of interest is then grown to the maximum object size (where possible) by including only those pixels with model colours, in regions spatially close to the cue. At different object sizes a match measure (a histogram intersection measure) is calculated. If any of the calculated match measures exceed a predefined threshold then a match is recorded at the cue.

In Chapter 4 an object recognition algorithm is described which utilises colour, object region geometry and a single geometric invariant to recognise objects in cluttered scenes. It is assumed that an object in a scene has only some of its regions occluded allowing a ratio of region areas invariant and region colour to be used to identify three corresponding model/image regions. A geometric transformation is calculated and the model and object are transformed into the same viewpoint for region matching using colour and position. The algorithm returns the position and pose of the object, as well as a match measure (based on the number of corresponding model and image regions found).

Chapter 5 describes a model-based object search algorithm which groups image regions based on their colour and topological relationships. For each database model a modified syntactic neural network is used to determine in best-first order the combination of image regions which satisfy the model's region colour and topological relationships. The robustness of the network makes it extremely fast; however, large amounts of memory are typically required. Methods of reducing the memory required are discussed, as well as the modifications made to the syntactic neural network.

Finally in Chapter 6 the conclusions are discussed.

Chapter 2

Colour Constancy, Image Segmentation, and Object Recognition

2.1 Introduction

There are two important requirements for a general colour object search algorithm: colour constancy and colour image segmentation. These two processes are so important that they require a thorough discussion (which is provided at the beginning of this chapter) before reviewing some of the more common object recognition/search algorithms. If model and test images presented to an object recognition/search algorithm are sensed under different illuminations then colour constancy is required. However, the colour constancy problem is under-constrained thus restricting the type of images used by object recognition/search algorithms. The colour constancy problem is formalised in Section 2.2 where some of the common algorithms are presented, such as Land's Retinex, Hung's Spectral Adaptation, Forsyth's CRULE and Finlayson's Colour in Perspective.

The more flexible object models describe objects as sets of spatially related regions; therefore they require an image segmentation algorithm to partition the image into regions of uniform colour characteristics (constant reflectance). Numerous algorithms have been devised to perform this task and are typically based on clustering in colour space, region splitting, region growing or a combination of these techniques. Some of these algorithms will be discussed in Section 2.3.

In Section 2.4 some common object recognition (as well as object localisation and object search) algorithms are reviewed and their advantages and disadvantages discussed. Finally, in Section 2.5 the properties of object recognition algorithms which are not constrained are discussed in preparation for the object search algorithm presented in Chapter 3.

2.2 Colour Constancy

The ability of an observer (human or otherwise) to perceive the colours of a given surface in a consistent way, under illuminants of different spectral distribution, is known as colour constancy. This definition was extended by Brainard and Wandell [BW86] to include: the maintenance of the colour appearance despite variations in the colour of nearby objects. Despite the human visual system not maintaining perfect colour constancy, it is better than any man-made system currently available.

Colour constancy is not only confounded by the spectral power distribution (***SPD***) of the ambient light and the object's surface reflectance — most researchers only consider these two factors in the design of colour constancy algorithms — but also by specular highlights (or specularities which cause a saturation of the sensors) and mutual illumination (an object illuminated by the light reflected from another object). In general, a specular highlight removal stage should be incorporated into the colour constancy algorithms; however in many images highlights are present over only a small part of the image. Therefore, highlight removal is often not necessary in the case of inhomogeneous dielectrics (materials with the property that both surface reflection and colorant-layer scattering from the body of the material are important optical processes). It is important to note however that specular highlights do bias the results of colour constancy algorithms. On the other hand, mutual illumination complicates the recovery process and is generally ignored, however Funt et al. [FDH91] have studied the effects of mutual

illumination and used it to determine the ambient illumination and discount variations in it.

The ***SPD*** of the ambient light and reflectance function of the surface can not be separated for all possible viewing conditions. For example no algorithm can correctly determine the surface reflectance of a single unknown object illuminated by an unknown illuminant. Therefore, colour constancy algorithms in general require several different objects in the scene. Typically, colour constancy algorithms are developed in the simplified Mondrian world which consists of planar, overlapping matte patches. In this world, surface descriptors can be determined from the ambient light which is assumed to be locally constant. The light reflected from a Mondrian patch falls on a sensor array at location x where there are s distinct sensor classes. The response $R_k(\lambda)$ registered by the kth sensor p_k^x is:

$$p_k^x = \int_\omega C^x(\lambda) R_k(\lambda) d\lambda \qquad (2.1)$$

given that $C^x(\lambda)$ is the colour signal at x which is given by:

$$C(\lambda) = E(\lambda) S(\lambda) \qquad (2.2)$$

where $E(\lambda)$: is the spectral distribution of the illumination

$S(\lambda)$: the surface reflectance function

The integral is taken over the visible spectrum ω $(\lambda_1 \ldots \lambda_2)$.

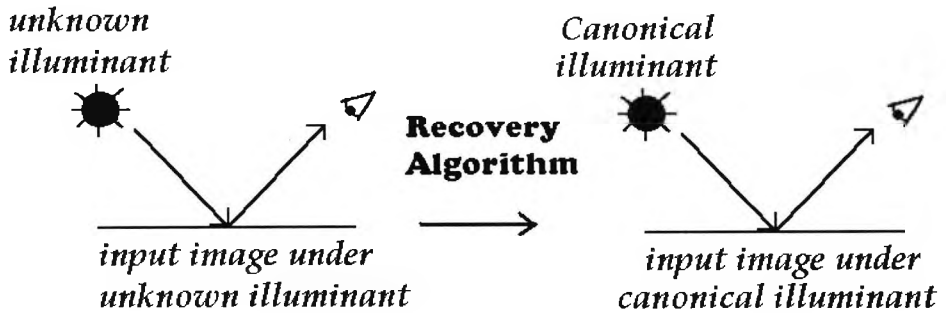


Figure 2.1: The Colour Constancy Problem

Figure 2.1 illustrates the colour constancy problem where an input image is illuminated by an unknown source (illumination dependent observations) and is transformed by a recovery algorithm into a known illuminant (illumination independent descriptors).

There are three basic linear image models used by typical recovery algorithms, these are trivial, coefficient and general [Fin95]. In the trivial (Eqn. (2.3)) case, a single coefficient (α) is used to map the observations (r,g,b) to descriptors (r',g',b'). This, in effect, is a global intensity scaling of the sensor channels. Conversely, in the coefficient model (Eqn. (2.4)) each sensor channel is scaled with a separate coefficient (α,β,χ). This was first proposed as a model for human vision by Von Kries [Kri78] who formulated the coefficient rule (This model is only valid however if narrow band filters are used). Conversely, in the general model (Eqn. 2.5) each descriptor value is a weighted sum of the observation responses (in Eqn. (2.5) nine coefficients are required).

$$\begin{bmatrix} r' \\ g' \\ b' \end{bmatrix} = \alpha \begin{bmatrix} r \\ g \\ b \end{bmatrix} \tag{2.3}$$

$$\begin{bmatrix} r' \\ g' \\ b' \end{bmatrix} = \begin{bmatrix} \alpha & 0 & 0 \\ 0 & \beta & 0 \\ 0 & 0 & \chi \end{bmatrix} \begin{bmatrix} r \\ g \\ b \end{bmatrix} \tag{2.4}$$

$$\begin{bmatrix} r' \\ g' \\ b' \end{bmatrix} = \begin{bmatrix} \alpha & \beta & \chi \\ \delta & \varepsilon & \phi \\ \varphi & \gamma & \eta \end{bmatrix} \begin{bmatrix} r \\ g \\ b \end{bmatrix} \tag{2.5}$$

2.2.1 The Retinex Theory

In [LM71], Edwin Land (and McCann) described a set of experiments which showed that humans are capable of perceiving the reflectance of a scene in a way that is largely independent of illumination (the illumination may be non-

uniform or unknown for that matter). He subsequently coined the term Retinex which was intended to describe the function of the retina and cerebral cortex during the processing of fluxes.

The set of experiments described involved the illumination of a set of Mondrians of different shapes using three independent sources of long, middle and short wavelength light. By using a telescopic photometer, the amount of radiation reflected from the surface of any Mondrian (which effectively measures the flux reaching the eye) could be measured. The test concluded that although the intensities of the incident illumination were changed, the colour of the Mondrians remained the same. Thus, the sensation of colour is not simply related to the product of the illumination and reflectance.

Land's Retinex algorithm ([LM71] and its modifications [Lan77] [Lan86] [BW86]) determine the surface colour without knowledge of the illuminant by using the coefficient rule[ii] and a contrast process; however, the contrast process is poorly understood. Also, since comprehensive results have not been published, it is difficult to determine the effectiveness of the algorithm.

2.2.2 Forsyth's CRULE Algorithm

Forsyth's CRULE algorithm is intended to solve the colour constancy problem for 2D objects under constant illumination with a small number of total distinct colours. Forsyth models (using the coefficient model described earlier) the effects of an illuminant on a scene by mapping the colour descriptor (receptor responses under a canonical illuminant) of a patch to the observed receptor responses under the given illuminant and discusses the conditions under which this mapping is invertible.

The algorithm proceeds by:

[ii] Von Kries [Kri78] coefficient rule requires a coefficient to be calculated for each class of receptor; the colour descriptor is then the output of each receptor multiplied by its coefficient.

1. Constructing the gamut under the canonical illuminant by imaging a large number of receptor responses of the illuminant which results in a bounded gamut. (The canonical gamut is defined as the convex set of *rgb* response vectors obtained by imaging a maximal set of reflectances, which are representative of all surfaces under a canonical illuminant).
2. Constructing the set of feasible mappings (the feasibility set) for any patch imaged under a constant illuminant.
3. Determining, using an estimator, the map most likely to correspond to the illuminant.
4. Applying the chosen map to obtain the colour descriptor.

In the experiments presented, the canonical hull (the convex hull of the gamut under the canonical illuminant) was formed by imaging 180 out of 202 coloured papers under white light. Colour constancy was then performed on 60 different papers in a Mondrian under six different illuminants with good results.

Forsyth's Crule algorithm however suffers from several restrictions including:

1. the requirement that all surfaces must be flat which is often not the case in real scenes.
2. no specularites can be present in the scene, however most surfaces often have a specular component.
3. the illumination power must be everywhere uniform.
4. the image must contain a diverse set of colours otherwise a large number of maps will result.

2.2.3 Hung's Spectral Adaptation

Hung [HE95] incorporates the coefficient model with a spectral adaptation process to achieve colour constancy. The adaptation process requires the selection of a set of reference colours — in a canonical colour space — which are used to describe the colour space. By assuming that a prominent image

colour is equivalent to a reference colour, the parameters for a hypothesis transformation are calculated. This hypothesis transformation is used to transform all the image colours into the canonical space and the distance (a Mahalanobis distance measure is used) between the transformed colours and the closest reference colour is determined. This distance is summed for all image/reference colour pairs. This process is repeated by assuming that the same prominent image colour is equivalent to another reference colour, until it is matched with all the reference colours. This entire process is repeated for all the reference colours. The hypothesis transformation which yields the smallest summed distance (in a least-square sense) is assumed to be the best transformation and is used to produce the colour constant image.

Some of the assumptions that are inherent within the algorithm are:

1. The spectral distribution of the illumination is everywhere constant.
2. The image surface patches are lambertian as in the Mondrian world (a 3-dimensional illuminant, and a 2-dimensional surface reflectance — the *3-2 case* described in [Fin95]).

As with the other colour constancy algorithms described there are problems with this approach including:

1. The process of generating and matching hypothesis transformations is exhaustive ($O(nm^2)$ for n database colours and m image colours).[ii]
2. No checks are made for channel saturation in the transformed (colour constant) image; i.e., no ranking of the best hypothesis transformations is performed, which is necessary since the real solution may not have the smallest least square distance.
3. It is not clear how the choice of reference colours affect the algorithm's performance. In one sense, the smaller the number of reference colours the faster the algorithm, but this might produce a poor result — what is the number of reference colours threshold for an optimal solution?

[ii] Take out one image colour and map that in turn to each database colour, computing the transformation and applying it to all remaining image colours. Repeat this for all image colours [HE95].

2.2.4 Finlayson's Colour-in-Perspective Algorithm

Finlayson [Fin95] proposed a new colour constancy algorithm which extended Forsyth's CRULE algorithm by placing simple constraints on the set of possible reflectances and illuminants. Forsyth's algorithm is based on two constraints:

1. surface colours under a canonical illuminant all fall within an established convex gamut of possible colours.
2. a diagonal matrix accurately maps colours between illuminants.

These restrictions force strong constraints on the scene (as described earlier). Finlayson however shows that these restrictions were only necessary because Forsyth attempted to recover the intensity of descriptors. Instead, Finlayson maps 3-dimensional (r,g,b) co-ordinates to a 2-dimensional chromaticity space r/b, g/b. It is in this diagonal chromaticity space that Forsyth's CRULE algorithm is applied and the 3D descriptor orientations derived. Finlayson subsequently extends the algorithm by placing a maximal gamut constraint on the set of illuminants (which is analogous to a gamut constraint on surface colours). The results show good colour constancy.

One consequence of the perspective transformation of the sensor data (mapping from 3D to 2D space) is that the calculated feasibility maps are also distorted. Finlayson [FH97] however later addresses this problem by removing the distortion prior to the map selection process. This colour constancy algorithm appears to be the most general to date.

2.3 Colour Image Segmentation

Colour image segmentation algorithms can assume two forms: Form 1, the identification of image regions of a known colour, and Form 2, the partitioning of images into regions of uniform colour characteristics. Generally, solutions fall into one of three classes: characteristic feature thresholding or clustering,

edge detection and region extraction. However some segmentation algorithms combine these methods to achieve more robust segmentation, e.g. [MK95].

Many of the colour image segmentation algorithms described in the literature are based on clustering (or thresholding) in colour space. These include Sarabi et al. [SA81], Andreadis et al. [ABS90], Gong et al. [GS95], Celenk [Cel90] and Khotanzad et al. [KB90]. Sarabi et al. [SA81] describes both a Form 1 and a Form 2 segmentation algorithm. In the Form 1 algorithm decision surfaces, defined interactively using straight lines, parabolas, and ellipses, are used to model the chromatic distributions. The results of the segmentation are the pixels of the given image which fall within the chosen decision surfaces. In the Form 2 algorithm, clusters are detected in the normalised colour space and their boundaries used as decision surfaces. Khotanzad et al. [KB90] also uses mode analysis of multidimensional histograms to effect a Form 2 segmentation with good results. Andreadis et al. [ABS90] used decision surfaces characterised by the mean and standard deviation of the colour in a normalised colour space to effect Form 1 segmentation. This algorithm was capable of accurately discriminating 1000 colours. Finally, Gong et al. [GS95] described pixels of a given colour using a second-order basis functions in the HVC (Hue/Value/Chroma) colour space and Celenk [Cel90] detected clusters in the 1976 CIE (L*, a*, b*) colour space using circular-cylindrical decision elements (for a Form 2 segmentation).

Colour space clustering however does not guarantee spatial coherence of the pixels from the cluster. If measurements overlap in colour space then the results may be poor and noisy. Alternatively, region growing methods (and sequential labelling [SHB93]) stress spatial coherence.

Tominaga [Tom90] utilised cluster-based and region merging techniques in the creation of a Form 2 segmentation algorithm. The input image is mapped into a uniform perceptual colour space where spectral cluster detection is performed. Discrimination between spectral clusters is achieved through

principal component analysis of the colour data. Image regions are extracted from these spectral clusters until no clusters are left. A post-processing process is used to merge smaller regions with larger ones based on a colour distance. Finally, all clusters classified earlier are reclassified, resulting in the merging of spectrally close clusters.

Matas et al. [MK95] described an image segmentation algorithm based on both spatial and feature space clustering (called FSD). The method proceeds by identifying unimodal clusters in the histogram of the image (in the case of a colour image, a colour histogram) and backprojecting the pixels contributing to the bins in the largest unimodal cluster to identify the connected components. The unimodal cluster of the largest connected component serves as a model which is used to statistically test for connected components with the same (feature space) characteristics. Each accepted connected component is grown to include pixels which are close in both the spatial and feature space domains. The pixels contributing to the accepted components are subtracted from the image histogram and the process repeated. The results presented demonstrate both the power and flexibility of this algorithm over traditional cluster-based methods.

The colour segmentation algorithms presented here are only a subset of the available algorithms. Other algorithms are due to Healey [Hea89] [Hea89b] [Hea92] [HDMN96] and Ohta et al. [OKS80].

2.4 Object Recognition Algorithms

Numerous colour object recognition algorithms have been described in the literature to-date; however, several of them are based on techniques such as histogram intersection and histogram backprojection. In this section a cross-section of these algorithms are presented which will include histogram intersection and backprojection as well as statistical and region-based techniques.

2.4.1 Histogram Intersection Based Algorithms

Swain and Ballard [SB90] [Swa90] first proposed the use of colour histograms and a process called ***histogram intersection*** to determine the identity of an object when its location is known. Histogram intersection matches the n bins of a model M and image I histogram using an L_1 metric:

$$H(I,M) = \frac{\sum_{j=1}^{n} min(I_j, M_j)}{\sum_{j=1}^{n} M_j} \qquad (2.6)$$

However, Stricker et al. [SO95] concluded that colour histograms are the major source of instability when used in similarity metrics induced by the L_1 [SB90] and L_2-related norms [NB93]. This instability occurs when there are deformations in the shape of the colour histogram which are due to changes in the ***SPD*** of the illumination (and illumination intensity), position of the light source, changes in viewpoint, and changes in the acquisition chain. Swain [Swa90] suggests that a small number of histograms are sufficient to describe a 3D object; however, as Matas [Mat96] shows the process of selecting representative viewpoints is non-trivial. Since similarity metrics are based on histogram bin counts, these conditions cause drastic changes in the metrics. Also, colour histograms are essentially 2-dimensional and only affine (shear, rotation, scale and translation) invariant. Alternatively, similarity functions based on modes (local maxima) in colour histograms are much more stable [Mat96]. Wixson et al. [WB89] demonstrates this for models comprising everyday household objects (cereal boxes, detergent containers, books and magazines) using an invariant based on the ratios of histogram mode populations.

Schettini [Sch94] presents a 2D algorithm for object recognition based on shape matching and colour distribution verification. A polygonal type approximation is used to describe the object boundary and a 3D colour histogram (of size 4x4x4) in CIELUV space is used for colour match verification. Since the boundary descriptor utilises angles and length ratios of

segments the method is only invariant to translations, rotations and scale. Clearly, it also inherits all the problems of the histogram intersection metric. In addition, the segmentation of objects from their background has been grossly simplified because black backgrounds were used.

Finlayson et al. [FF95] [Fin92] described a modification to Swain and Ballard colour histogram intersection which allows for variations in the SPD and intensity of the illuminant. The technique is based on the histogramming of colour ratios in neighbouring regions. Although the method works well on both real and synthetic images it inherits all the problems of the histogram intersection metric.

2.4.2 Histogram Backprojection Based Algorithms

One of the earlier and more popular solutions to the object localisation problem — a known model but unknown position — was proposed by Swain and Ballard [SB90] [Swa90]. The localisation algorithm, called ***histogram backprojection***, determines a confidence value for each pixel in the image. Given an image H^I and a model H^M histogram, the confidence value assigned to pixel P_{xy} is given by:

$$C(x,y) = \frac{H_i^M}{H_i^I} \qquad (2.7)$$

where pixel P_{xy} maps to histogram bin i. Peaks in the confidence space after smoothing correspond to object hypotheses.

Histogram backprojection, however suffers from several restrictions:

1. The algorithm is 2-dimensional and only affine invariant
2. The size of the object must be known a priori (to determine the size of the smoothing mask).
3. The object hypotheses are not ranked.

Vinod et al. [VM96a] describes an object localisation algorithm based on Swain and Ballard's ***histogram backprojection*** (***BP***) or ***focused colour intersection*** (***FCI***) [VM96b] and ***focused discrete cosine transform*** (***FDCT***) (the discrete cosine transform (***DCT***) is used by image compression algorithms [PTVF88]). The algorithm uses ***BP*** (or ***FCI***) to localise regions in the image with high match confidence then applies the ***DCT*** to the cued local image regions.

Although this algorithm appears to work well it suffers from several problems:

1. ***BP*** + ***FDCT*** (or ***FCI*** + ***FDCT***) still suffers from the problems of histogram intersection and backprojection.
2. The technique is not viewpoint invariant.

Ennesser et al. [EM95] describes an object localisation algorithm based on matching local histograms using a weighted histogram intersection measure. It is shown that in the worst case this algorithm degenerates to Swain et al.'s histogram backprojection. The test set are digitised pictures from the "Where is Waldo" books. The scale of the model is determined by gradually increasing the size of the local region (under consideration) until there is no change in the confidence (the ***histogram intersection*** measure) value. To tolerate colour variations a co-occurrence histogram was formulated which modelled the colour of each pixel and its neighbours. However, this increased the complexity of the algorithm, as well as reduced invariance to simple transformations (e.g. scale and rotation).

2.4.3 Wixson et al.'s Localisation Algorithm

Wixson et al. [WB89] described an active vision system used in the real time detection of multicoloured objects. The technique assumed that a colour histogram of an image, containing a model object, contains a spectral signature which is invariant over a large number of conditions. By successively adjusting the gaze of a camera mounted on a robot arm in the middle of a 16' x

24' room and rotating 360° in 15° increments, (with pitch angles to allow the examination of the upper and lower walls) a set of significant gazes were determined — using a confidence measure. The confidence measure used was based on the ratio of the populations of the modes in the colour histogram, which forms an invariant. Wixson [Wix94] subsequently reduces the number of gazes examined through the use of a priori knowledge of the scene. By associating each object with some larger, intermediate object and recording the spatial relationship, the object search problem was reduced to searching for the larger object, which is less difficult computationally. For indoor scenes Wixson reported an up to eight-fold improvement in search time when intermediate objects were used.

2.4.4 The Colour Region Adjacency Graph

Syeda-Mahmood [Sye92], Matas et al. [Mat96] [MMK95] and Olatunbosun et al. [ODE96] have presented colour localisation/recognition algorithms based on the colour region adjacency graph (***CRAG***). In the ***CRAG*** each region is a node in the graph and the line connecting two nodes a graph edge. The ***CRAG*** algorithm presented by Syeda-Mahmood [Sye92] was computationally restrictive due to the complexity of the sub-graph search process (when attempting to locate a model ***CRAG*** within an image ***CRAG)***. As a result Matas et al. [Mat96] [MMK95] improved its performance by augmenting the ***CRAG*** with a Colour Adjacency Graph (***CAG***) — to simplify the graph search process — whose nodes represent a single image colour and edges the reflectance ratio (a photometric invariant). The localisation is then reduced to locating the model ***CAG*** in the image ***CAG*** and backprojecting it into the image ***CRAG***. Olatunbosun et al. [ODE96] augments the ***CRAG*** with geometry invariants (Euclidean invariants) such as the ratio of distances between region triplets and the angle between any two graph edges. The quality of match between a model and image is determined by the maximal clique — the maximum number of matching graph edge pairs.

Although the ***CRAG*** model is extremely powerful it has some important limitations:

1. It does not implicitly model non-adjacent regions.
2. At low spatial resolutions the graph size becomes large and more difficult to search.

2.4.5 Statistical Based Algorithms

Keller et al. [KCUU86], Mehtre et al. [MKNM95] (extended in [KMW96]) and Stricker and Orengo [SO95] utilise statistical shape descriptors of 1-dimensional colour space components to describe images. [KCUU86] produced the first of these algorithms which used the mean, standard deviation, skewness and kurtosis of the 1-D components of the RGB space to describe the degree of doneness in beefsteak. Stricker and Orengo [SO95] used a remarkably similar set of shape descriptors (mean, standard deviation and skewness) in the HSV colour space to describe colour images for an image retrieval application. Similarly, Mehtre et al. used the mean of the 1-dimensional components of the RGB colour space to describe images. Each of these techniques suffer from similar problems, that of background clutter. If there is a lot of background clutter then the statistics returned by these algorithms are incorrect (in the case of [KCUU86] segmentation would be complicated by background clutter). Also, higher order shape descriptors (moments) are unstable. Illumination changes also affects these algorithms significantly although hue (in the HSV colour space) is more invariant to these changes than the RGB colour components. One major advantage of these algorithms however is that an image is described by only a few floating point numbers (nine in the case of [SO95], three in [MKNM95]).

2.4.6 Other Localisation/Recognition Algorithms

Several other localisation/recognition algorithms currently exist. An earlier algorithm [Ber87] utilised nine features (the 1-dimensional components of the normalised colour space and the hue and saturation values from a variety of

other colour spaces) to describe the colour of spray can caps under different illuminants. Weill and Yair [WN90] produced an Orange fruit recognition algorithm based on colour segmentation, using 1-dimensional components of the RGB colour space, and a hough transform for circle detection.

Strachan [Str93] describes an algorithm for fish recognition using shape and colour features. As a result the algorithm is invariant to fish bending and deformations. Finlayson et al. [FCF96] describes images by their three inter-band angles allowing images to be represented by just six numbers. This technique produced good recognition rates.

Gevers et al. [GS96] describes a photometric colour invariant based on hue-hue edges. Using a similarity function the algorithm is then applied to the image retrieval problem. Matas et al. [MMK93] models the illuminant of a given environment in order to recognise objects present in it. And finally, [HS94] describes a histogram descriptor which is invariant to changes in intensity and the ***SPD*** of the illumination.

2.5 Discussion

In this chapter the colour constancy problem was discussed and common constrained solutions presented including Land's Retinex, Forsyth's Crule, Hung's Spectral Adaptation, and Finlayson's Colour-in-Perspective algorithms. The performance of Hung's algorithm was accessed (since it was available in-house) for an indoor environment (tungsten and fluorescent lighting) and was shown to improve the image colour marginally (c.f. Appendix 2). It is expected that algorithms such as Finlayson's Colour-in-Perspective would further improve the performance of the colour matching measure used throughout this thesis. However, it is beyond the scope of this thesis to provide empirical data for any other colour constancy algorithm.

Also discussed in this chapter were several image segmentation algorithms which varied in performance and complexity. There has always been a debate in the vision community as to which image segmentation should be used in a particular application (since no generic image segmentation algorithms have been developed). However, little progress has been made in this debate. Therefore a relatively simple (low computational complexity) colour image segmentation algorithm was used throughout this thesis. This method, which is based on Khotanzad et al. [KB90] technique, is similar to Matas' FSD clustering algorithm. This algorithm is detailed in Chapter 3.

The choice of object recognition algorithm is dependent on the application. If objects can be represented by a set of regions then a ***CRAG*** model would be invariant to many more conditions (e.g. viewpoint and moderate occlusion) than say a colour histogram-based technique. However, if objects are highly textured then colour histogram methods or statistical based methods might be more attractive. If the size of the model representation (i.e. the number of floating point numbers required to represent the model) is crucial then a statistical-based method is better since an entire image can be represented by a few numbers. The problem with statistical-based methods, however is that they are highly sensitive to background clutter and illumination changes.

There is a general lack of generic colour object search algorithms (invariant to conditions such as: moderate occlusion, illumination changes, image clutter etc.). Algorithms based on Swain's histogram backprojection can only represent 2-dimensional objects and typically require that the object size is known a priori to search. Wixson [Wix94], on the other hand exploits the intermediate objects philosophy and effectively reduces the object search space; however his emphasis was more on the defining of the relationship between the immediate and actual object rather than the modelling and recognition of the object being searched.

There is a need for an object search algorithm capable of representing both 2 and 3-dimensional objects (which may be perspectively deformed) that can perform searches in complex, cluttered environments with moderate object occlusion; this thesis will present such an algorithm.

In Chapter 3 the first of two object search algorithms is presented. Although this algorithm is only capable of modelling 2-dimensional or 3-dimensional planar objects that are affine distorted, it is able to perform searches in complex cluttered scenes. A more general search algorithm will be presented in Chapter 5.

Chapter 3

A Colour-Based Object Search Algorithm

3.1 Introduction

Some researchers treat the task of object search as a one step process; that is given the identity of a model, the entire image is searched until the model is found. However, if the object search task is decomposed into object localisation followed by object recognition, cueing mechanisms can be fully utilised thus reducing the search space; this in turn simplifies the search process. It is important to note that the object localisation algorithm should have a lower order of complexity than the object recognition algorithm if this two step process is to improve search times.

Assume that a model database of n objects exists and it is required to determine if these models are present in a given image. For argument sake, let it further be assumed that the order of complexity of the object localisation algorithm is $O(X)$ and for the object recognition algorithm $O(Y)$ where $O(X) << O(Y)$. Now, the n models are used by the localisation algorithm giving a complexity of $O(nX)$. However, the localisation algorithm effectively prunes the model database (for example if the localisation algorithm is based on object colour without considering the spatial relationships of object regions, then models whose colours are not in the image are not present) resulting in n_1 ($n_1 < n$) models that might be in the image. The resulting order of complexity of the two step object search process is $O(nX + n_1Y)$, as opposed to $O(nY)$. Just to give an idea of the time savings, assume that the object recognition algorithm is twice as expensive computationally as the object localisation algorithm and that the localisation process prunes the database by half. Therefore, $Y=2X$ and $n_1=n/2$. The resulting orders are the same $O(2nX)$. Therefore, provided that the localisation algorithm prunes the model database

(by at least half) and the order of complexity of the localisation is at most one half of the recognition algorithm, then the two step object search process is faster than the one step approach. In general however $O(X) << O(Y)$.

In this chapter an algorithm for colour object search is described. For each model in an object database, image regions with colours that are similar to the salient model colours are identified. These image regions serve as areas of interest. A match measure is calculated for each cue location and if it exceeds a predefined threshold then an occurrence of the model is assumed to be at that location. The conditions under which this algorithm is invariant are:

1. 2- or 3- dimensional planar object representations.
2. affine object distortions.
3. Changes due to illumination intensity and spectral power distribution (for fluorescent and tungsten filament lighting).
4. absence of colour regions due to moderate occlusion (up to 50% occlusion is allowed).

The organisation of the remaining sections in this Chapter are as follows: Section 3.2 describes the properties of some popular colour spaces and identifies the colour space used by this algorithm. Section 3.3 describes the components of the object localisation algorithm which include the colour image segmentation technique, model creation, cue generation and model match determination. The results of this algorithm applied to real world images using models with different poses, scale, illumination and moderate occlusion are presented in Section 3.4 and finally in Section 3.5 this technique is discussed and compared with other object search algorithms.

3.2 Colour Spaces

The choice of colour space used by researchers is quite often ad hoc. because so many colour spaces exist with similar or drastically different properties. Many researchers simply determine the colour space to use after performing

experiments on many colour spaces or examining the literature. To list exhaustively the known colour spaces and their properties would be time consuming and pointless (to this thesis), rather a selected few will be discussed. These are RGB, opponent colour, IHS (Intensity/Hue/Saturation) and the normalised colour space.

The RGB Colour Space

Most colour cameras are designed to sense three primary wave lengths 410nm (Red), 530nm (Green) and 650nm (Blue) i.e. RGB (which copies the human visual system). However, although the RGB basis is good for image acquisition it is not particularly good for colour perception because it encodes both colour and intensity information [BB82]. For this reason other non-RGB bases are used such as: Opponent process, Normalised colour and Intensity/Hue/Saturation (IHS).

The Opponent Colour Space

The basis transformation from RGB to opponent colour is given by:

$$rg = R - G \tag{3.1}$$
$$by = -R - G + 2B \tag{3.2}$$
$$wb = R + G + B \tag{3.3}$$

The properties of the opponent colour space are:

1. The basis transformation is linear, transforming RGB measurements into two colour (***rg*** and ***by***) and one intensity channel (***wb***).
2. Since the transformation is linear, changes in illumination, changes in viewing geometry or changes in acquisition equipment, which can be modelled by a scaling factor ***k***, result in a shift of the ***wb***, ***rg*** and ***by*** values [Mat96].The colour space is therefore not invariant to these conditions.

The Normalised Colour Space

The basis transformation from RGB to the L^1 normalised colour space is given by:

$$r = \frac{R}{R+G+B} \tag{3.4}$$

$$g = \frac{G}{R+G+B} \tag{3.5}$$

There are some important properties of the normalised colour space worthy of note:

1. It has been suggested that two points on the same surface, where one of the points is in a shadow, will have the same chromaticity co-ordinates. This is only true if the points were illuminated by the same spectral power distribution which is often not the case in shadows.
2. The L^1 normalised colour space is independent of scene geometry.
3. There is a non-removable singularity at zero signal (r = g = b = 0) in sensor space and is highly unstable near this point.

A more complete description of the properties of the normalised colour space can be found in [Hea92] [Hea89] [Ken76].

The IHS Colour Space

The intensity in this basis is computed as:

$$intensity = R + G + B \tag{3.6}$$

The lack of whiteness in a colour is measured by saturation. Colours such as "fire engine" red are saturated while pinks and pale blues are desaturated. Saturation can be computed by:

$$saturation = 1 - \frac{3 \cdot min(R,G,B)}{intensity} \tag{3.7}$$

Hue is approximately proportional to the average wavelength of the colour. Based on the Torrance-Sparrow reflection model (the reflection from composite material is approximated by the sum of the body reflection and surface reflection components) and white illumination, hue is independent of viewpoint, surface orientation, illumination direction and intensity and highlights [GS96]. However, in general scenes the white light assumption is not valid. One of the many basis transformations can be defined by the program fragment:

$$hue = \cos^{-1}\left\{\frac{1/2[(R-G)+(R-B)]}{\sqrt{(R-G)^2+(R-B)(G-B)}}\right\} \qquad (3.8)$$

If $B > G$ then $hue = 2\pi - hue$

where R, G and B are the RGB colour space co-ordinates. This definition of hue has an essential singularity at $R = G = B$. Many authors including Gevers et al. [GS96] utilise hue as a partial illumination invariant; however, in this thesis the normalised colour space is used (because of its independence of scene geometry).

3.3 The Algorithm

Before the object search algorithm can be presented the model building process must be described. During model building, the image of the model is partitioned into regions of constant colour (using a histogram backprojection process called the software colour filter (***SCF***) developed in earlier work [WE95] — this approach was also used by Matas and Kittler [MK95] in their Feature and Spatial Domain Clustering (***FSD***) algorithm) and the colour (colour processing is performed in the normalised colour space) and area of each region determined (c.f. Figure 3.3 and Table 3.1). These parameters are determined for all the database models. To localise a given model in an image, image regions with matching model colours are identified and treated as seed co-ordinates for a growing process. Growing starts at the given seed point and includes neighbouring image pixels with colours which are similar to model colours. A match measure is calculated for different object sizes and the object size with the largest match measure is assumed to be an occurrence of the model if the match measure is above a given threshold. This process is repeated for all of the image regions (which match model colours) identified earlier.

The ***SCF*** uses graph-theoretical clustering (a feature-based clustering algorithm) [KNF76][KB90] to identify clusters in colour space then backprojects image pixels into these clusters and identifies the spatial regions

associated with the clusters. The ***SCF*** process therefore partially segments the image into regions of uniform colour characteristics.

3.3.1 Graph-theoretical Clustering

Graph-theoretical clustering is a feature-based image segmentation algorithm (Algorithm 3.1) which identifies unimodal clusters in the colour histogram of the image. This non-iterative peak-climbing clustering algorithm was first introduced by Koontz et al. [KNF76] and was subsequently extended by Khotanzad et al. [KB90] who determined the optimal histogram size (based on cluster density) for a given image for good segmentation. This segmentation technique was chosen because of its low computational expense and good segmentation results (in the experiments performed); also, no prior cluster distribution model was required to perform the segmentation.

Figure 3.1: A colour test image used to generate the colour histogram illustrated in Figure 3.2.

An example 16x16 chromaticity histogram (created by quantising the r and g channels of the normalised colour space) of the image illustrated in Figure 3.1 is clustered using Algorithm 3.1. The histogram bins with similar shading represent identified unimodal clusters (c.f. Figure 3.2). The cell with the maximum value in each unimodal cluster is a peak cell.

Algorithm 3.1: **Graph-theoretical clustering**

1. Generate a chromaticity histogram of the image maintaining a list of the pixels contributing to each bin.
2. For each bin determine the bin with the maximal count in a given neighbourhood (an 8-neighbourhood was used, however a 4-neighbourhood yields similar results [Mat96]). Store a link to this bin.
3. At the end of the link assignment, peaks are cells with the largest count in the neighbourhood and all other cells connected to this peak form unimodal clusters.

g/r	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*
0																
1													6.78			
2									0.71	1.84			1.49			
3		0.27					0.27		1.45	0.12						
4		1.41	0.39	5.18	0.31	0.51	3.69	0.16			3.02					
5					3.41	2.59	3.88	20.9	24.9							
6			0.39		0.20	0.04	0.12	0.98	8.51							
7																
8			0.63	0.04				2.20								
9			0.35			0.55										
10																
11																
12			2.67													
13																
14																
15																

Figure 3.2: A 16x16 normalised chromaticity histogram (each cell contains the percentage of image pixels with that colour) of the image in Figure 3.1. Histogram bins with the same shading belong to the same unimodal cluster. Eleven unimodal clusters have been found in this histogram (the peak cells are underlined). The horizontal axis is the ***r*** chromaticity co-ordinate while the vertical axis the ***g*** co-ordinate.

It is assumed that Hung's [HE95] colour constancy algorithm is applied to all images prior to processing.

3.3.2 The Software Colour Filter

The software colour filter [WE95][Wal96] was designed to partially segment colour images using colour space-based clustering followed by spatial clustering. The ***SCF*** when applied to model (or test) images identifies regions of constant reflectance. The ***SCF*** is described in Algorithm 3.2.

Algorithm 3.2: **The software colour filter**

1. Generate a chromaticity histogram of the model image (in our earlier work we used opponent colour histograms [Wal96]. The advantages of the normalised color space have been discussed earlier).
2. Perform graph-theoretical clustering and assign unique labels to each unimodal cluster found.
3. Backproject the pixels belonging to each unimodal cluster into the image, associating the label of the unimodal cluster with the pixel.
4. Perform spatial clustering (connected component analysis) of the labelled image grouping pixels with the same label into regions.
5. Extract the parameters (area and colour) from each resulting region.

3.3.3 Model Creation and Object Search

During model creation, the model image is segmented, regions with small area discarded (because these are often noisy regions) and the colour (represented by its mean r and g normalised co-ordinates) and area of each region recorded. Each model is subsequently defined by five parameters for each model colour. These parameters are the r and g normalised co-ordinates, the minimum and maximum region area percentages for the given colour (i.e. the number of pixels in the region divided by the total number of object pixels), and the sum

of the region area percentages for the given colour. The model parameter generation process is described in Algorithm 3.3.

Figure 3.3: A database model.

r	g	Percentage coverage	Percentage coverage of smallest region	Percentage coverage of largest region
0.48	0.49	88.4	16.4	72.0
0.14	0.24	11.6	2.0	9.6

Table 3.1: The parameters for the model in Figure 3.3.

An example of these parameters for the model in Figure 3.3 is presented in Table 3.1. Four representative regions were selected from the model, two of them yellow and two blue. The largest region (which is on the right) is yellow and occupies 72% of the total model area. The smallest yellow region (on the far left) occupies 16% of the total model area. Similarly, the largest and smallest blue regions occupy 10% and 2% of the total model area, respectively. The total percentage coverage for yellow is 88% and blue 12% (c.f. Table 3.1).

Algorithm 3.3: **Model Parameter Generation**

1. Segment the model image and discard regions with small areas.
2. Determine the regions, found in Step 1 with similar colour and calculate the total area of these regions.
3. For each model colour store in the model database: the chromaticity co-ordinates (r,g) of the model colour, the percentage of the total model area (percentage coverage) with the given model colour, and the percentage

coverage of the smallest and largest image regions with the given model colour.

4. Repeat Steps 1-3 for each database model.

The model parameters generated in Algorithm 3.3 are stored in the model database. Given an image which has been pre-processed by Hung's colour constancy algorithm [HE95], object localisation is achieved by applying Algorithm 3.4.

Algorithm 3.4: **Colour Object Search**

1. Segment the image and discard any regions with area greater than 50% of the image area (if this is not a background region the algorithm will still detect the object correctly by identifying other regions of this object and growing).
2. Repeat Steps 3 - 7 for each database model:
3. **Cue generation:** Determine all image regions with colours that are similar to model colours. The locations of these regions serve as cues. Region and model colours match if $C = \sqrt{(\mu_r - \mu_r')^2 + (\mu_g - \mu_g')^2} < cthreshold$ where (μ_r, μ_g) and (μ_r', μ_g') are the chromaticity co-ordinates of the model and image colours respectively.
4. Repeat Steps 5 - 7 for each cue:
5. **Object size determination:** By assuming that the cue region is part of the model and not more than half of it is occluded, a minimum (***min_size***) and maximum (***max_size***) object size bound can be calculated:

$$min\_size = \frac{number\_of\_pixels\_in\_cue\_region}{PCLMR} \cdot 100 \quad (3.9)$$

$$max\_size = 2 \cdot \frac{number\_of\_pixels\_in\_cue\_region}{PCSMR} \cdot 100 \quad (3.10)$$

where ***PCSMR*** is the percentage coverage of the smallest model region with a similar colour as the cue region; and ***PCLMR*** the percentage coverage of the largest model region with a similar colour as the cue region.

6. **Region Growing**: Divide the image into N windows each of size $n \times m$ — so that the model can be localised down to a set of windows rather than a set of regions (which might span much of the image). In each window use the segmentation information from Step 1 and the colour matching measure of Step 3 to determine the number of pixels in each window with model colours. Given that k is the object size increment then:

   for (*object_size* = *min_size*; *object_size* <= *max_size*;
   object_size += *k*) {

   (a) *no_ object_pixels* = 0;

   (b) Determine the window corresponding to the centroid of the cue region.

   (c) Grow into a neighbouring window (8-neighbourhood) if the colours in that window increases $H(I,M)$, equation (3.11).

   (d) if $H(I,M)$ increases then *no_object_pixels* += no of pixels in window with model colours.

   (e) Repeat Step (c) until: *no_object_pixels* >= *object_size*, or there are no more neighbouring windows containing model colours; or $H(I,M)$ is maximised.

   }

7. **Match measure determination**: The maximum $H(I,M)$ for all object sizes is assumed to be the match measure. If this value exceeds *match_threshold* then the model is assumed to exist at this cue location.

$$H(I,M) = \sum_{j=1}^{n} min(I_j, M_j) \qquad (3.11)$$

where $M_j = \dfrac{number\_of\_model\_pixels\_with\_colour\_j}{model\_size}$

and $I_j = \dfrac{number\_of\_object\_pixels\_with\_colour\_j}{object\_size}$.

The object size calculation in Step 5 of Algorithm 3.4 is based on the assumption that the cue region is part of the object and not more than half of it is occluded. For example, consider a model with three regions, two red occupying 25% and 50% of the total object area and one green (occupying the remaining 25%). If a red region containing 20 pixels is found in the image then the minimum object size is $\boldsymbol{min\_size} = \frac{20}{50} \cdot 100 = 40$ pixels. Similarly, the maximum object size $\boldsymbol{max\_size} = 2 \cdot \frac{20}{25} \cdot 100 = 160$ pixels.

As a result of Step 7 of Algorithm 3.4 it is possible that several model candidates may result. It is assumed that the best candidate is the one with the smallest colour error $\boldsymbol{E}$ where $\boldsymbol{E} = \sum \boldsymbol{E_i}$ and $\boldsymbol{E_i} = \sqrt{(\boldsymbol{r} - \boldsymbol{r'}) + (\boldsymbol{g} - \boldsymbol{g'})^2}$ where $\boldsymbol{(r,g)}$ is the chromaticity co-ordinates of the candidate pixel and $\boldsymbol{(r',g')}$ the chromaticity co-ordinates of the model colour closest to $\boldsymbol{(r,g)}$.

3.4 Results

To determine the performance of the object search algorithm under changes in illumination, spatial resolution, affine object distortion, object occlusion and image clutter an image set of six images was selected. These images were captured in an indoor environment and were illuminated by either fluorescent or tungsten filament lighting (or a combination of both). It was believed that these images provided a sufficiently rigorous test of the algorithm. It was expected that the algorithm would find all the models correctly (no false negatives) with a small number of false positives.

The model database (which contains 25 models) used in these experiments is illustrated in Figure 3.4. The database contains books, cereal boxes, playing cards and a Christmas card box. Several of these models have similar geometry, for example models 1, 2, 4, 5, 6, 7 and 15; models 12, 13 and 14; models 20, 21, 22, 23, 24 and 25; and models 9, 10 and 11. The playing card

models (9, 10 and 11) have only two colours, white and red and in the case of models 10 and 11 have similar colour proportions. Models 13 and 14 have practically the same representative colour regions — the text printed on the covers 'Debugger' and 'Assembler' is the only major difference in the models. Models 2 and 6 also have, for the most part, the same representative colours.

Figure 3.4: The model database.

The test images used in these experiments are illustrated in Figure 3.5. Because the model and images were captured under different illuminants, Hung's [HE95] colour constancy algorithm was applied to all images before processing.

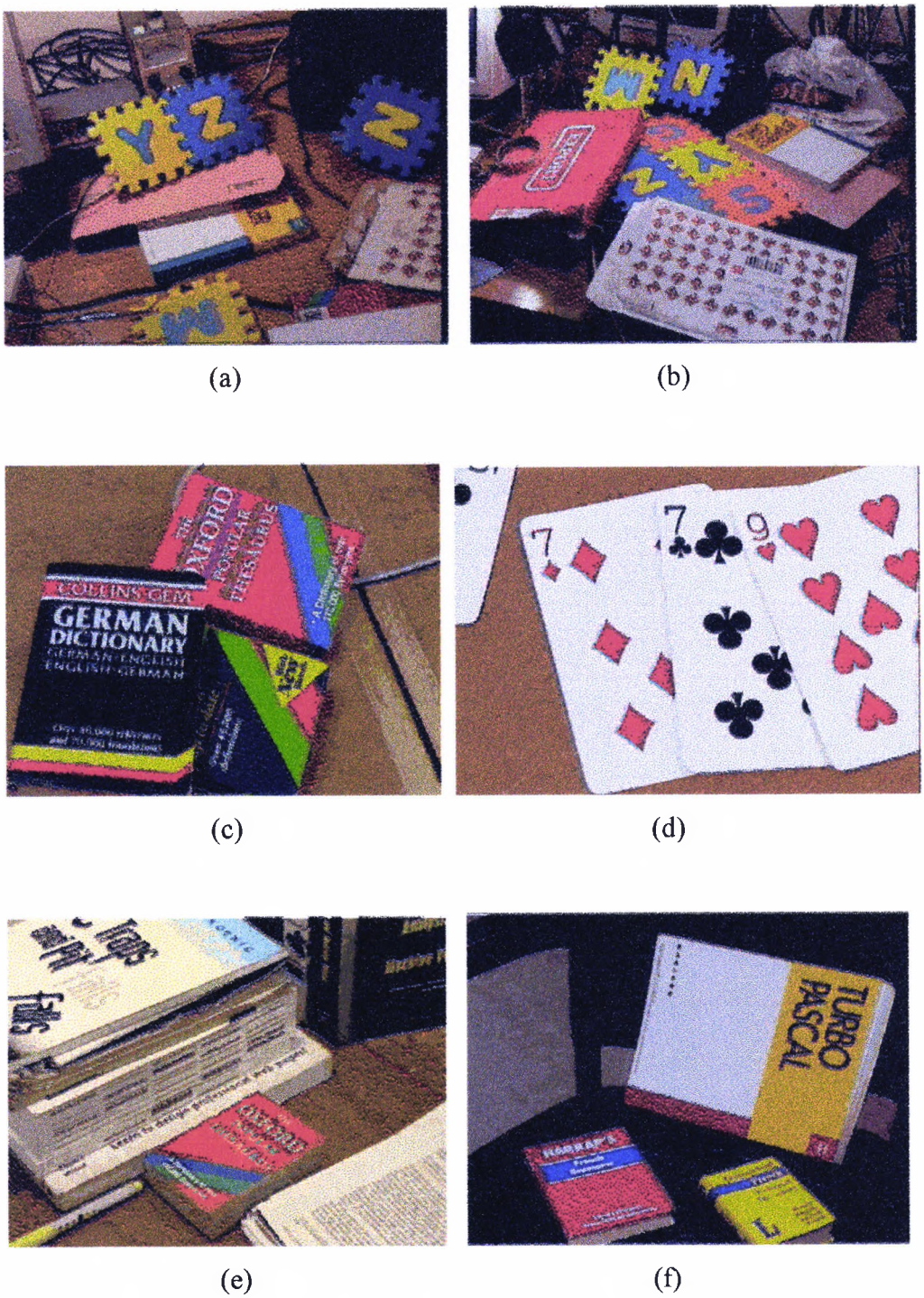


Figure 3.5: The images used in the object search experiments.

Figure 3.5 (a) contains an occluded model 6 and 14; Figure 3.5 (b) also contains model 14. Figure 3.5 (c), (d), (e) and (f) contain model 1 and 2; model 9 and 10 (both occluded); model 2; and model 5, 7, and 12, respectively.

Table 3.2 presents a summary of the results of applying Algorithm 3.4 to the six images of Figure 3.5. The first column of this table contains the image identifier (Figure 3.5 (a) - (f)), the second column contains the number of database models that are present in the given image. The third column contains the placement of each match, that is whether the cue with the best match value (1^{st}) represents the object, or is the second best (2^{nd}), or the third (3^{rd}) best or greater than that. The fourth column contains the total number of false positives that have occurred, and the fifth, the total number of false negatives. Finally column six gives the percentage reduction in the search space (which is defined as the number of image windows containing the localised model over the total number of image windows).

Image	***Number of models in image***	***Correct Match Placement***				***False positives***	***Percentage reduction in search space***
		1st	*2nd*	*3rd*	*>3rd*		
(a)	2		2			9	80.7
(b)	1	1				9	78.7
(c)	2	1	1			11	58.3
(d)	2	2				5	0.0
(e)	1		1			9	60.1
(f)	3	1	1	1		8	41.6

Table 3.2: A summary of the results of applying Algorithm 3.4 to the images of Figure 3.5.

In image (a) (Figure 3.5 (a)) models 6 and 14 had the second best rank at their correct image location, however there were 9 false positives. The percentage reduction in the search space was 80.7% and the percentage reduction of the models present in image (a) is 44% (11/25). The remainder of the table is interpreted in the same way. It is important to note however that there was no appreciable reduction in the search space for Figure 3.2 (d).

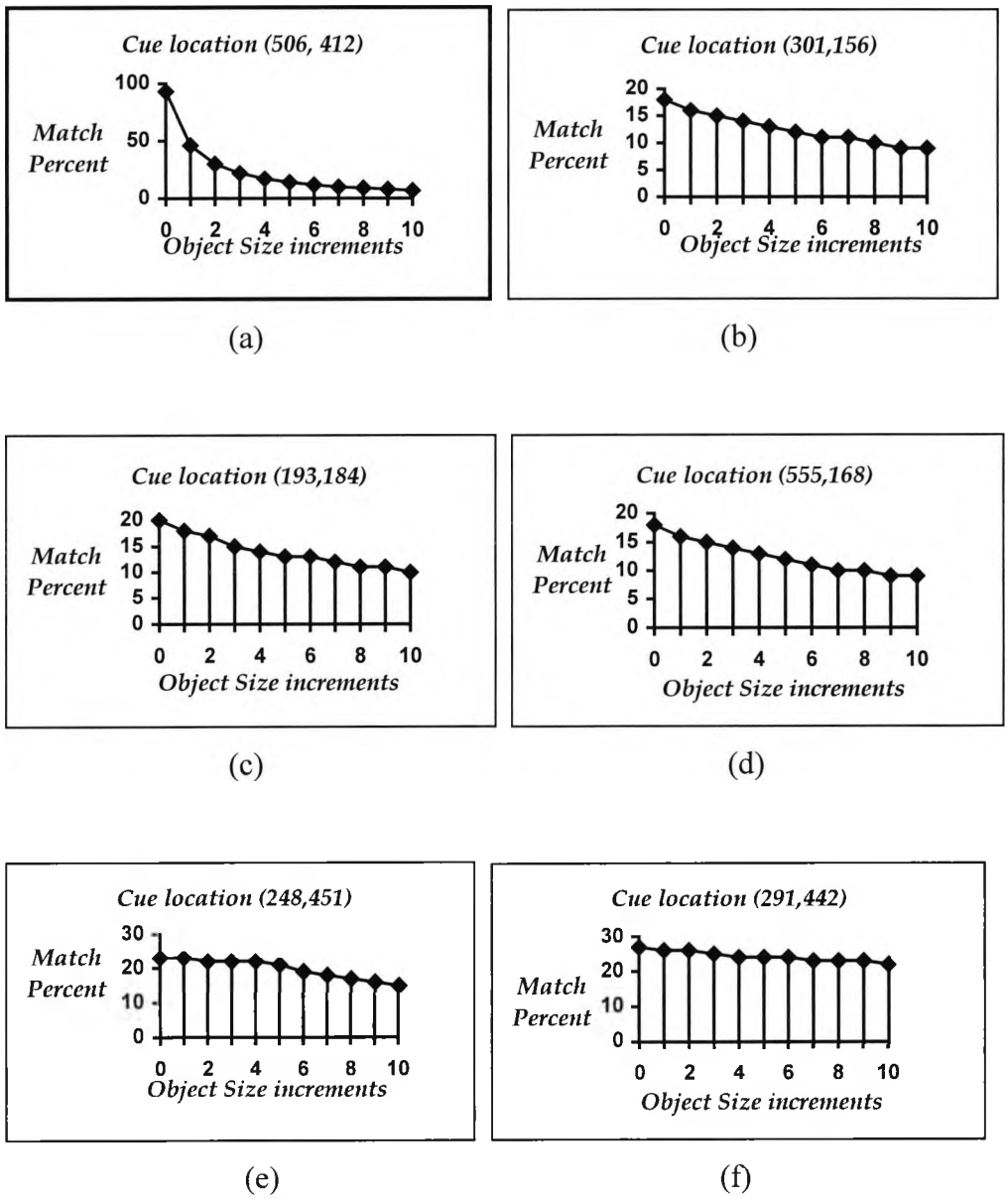


Figure 3.6: The match percentages for increasing object size increments at the cue locations indicated when searching for model6 in image Figure 3.5(a). In this example the location (506,412) which yields a match percentage of 93%, (a) contains model6.

To illustrate the search process, consider the search for model6 in Figure 3.5(a). The parameters used in all the experiments were: window size 10×10 pixels, minimum object match percentage 88.0% and 11 object sizes were selected between ***min_size*** and ***max_size*** (all equally spaced) for each region cue found. In Step 2 of Algorithm 3.4, six cue regions were identified with centroids at the (x,y) co-ordinates: (506,412), (301,156), (193,184), (555,168), (248,451), and (291,442). At each cue, the growing process was performed for each of the 11 object sizes (c.f. Figure 3.7) and the match measure calculated.

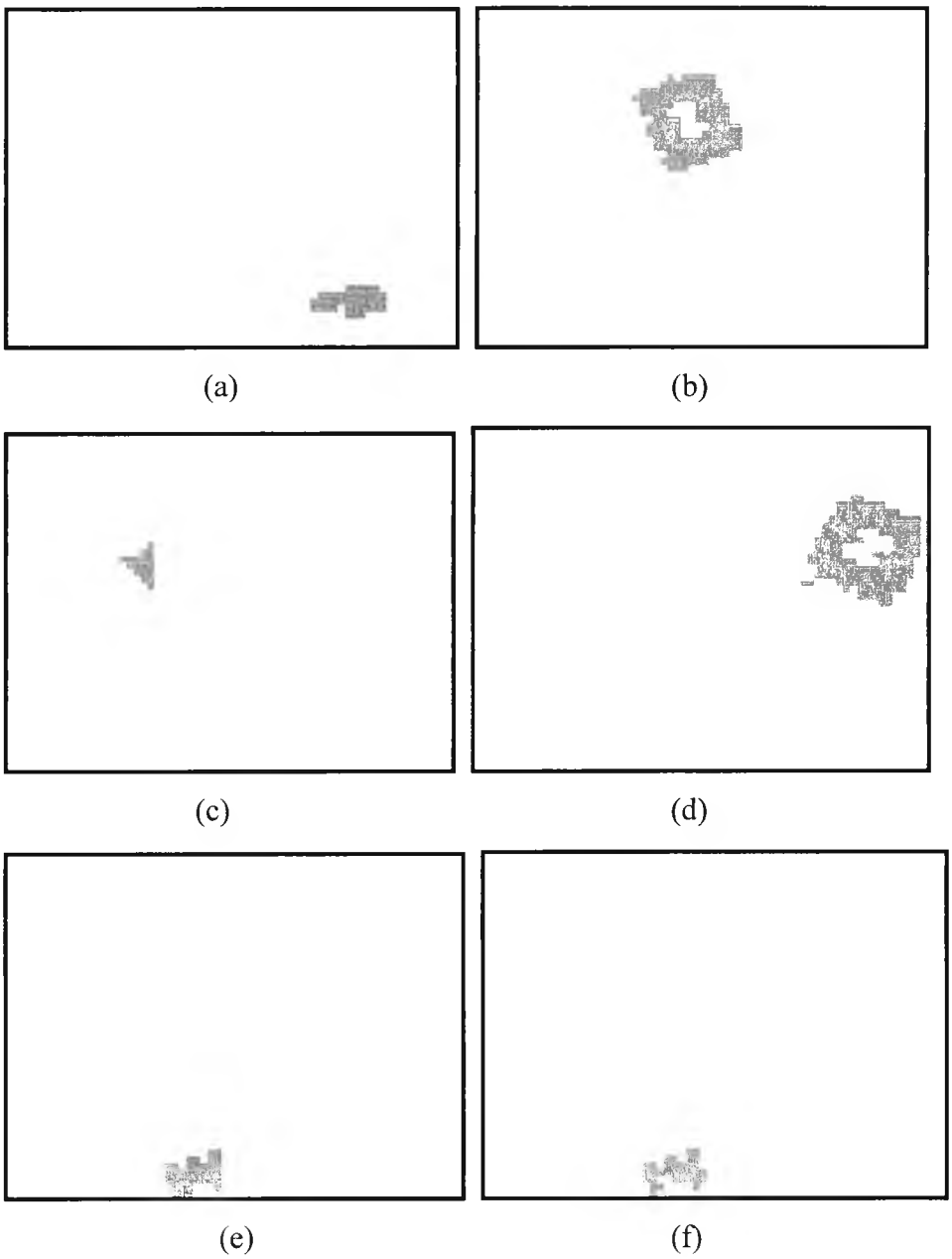


Figure 3.7: The six cue regions.

The only cue which generated a match percentage greater than the 88.0% minimum match percentage is at co-ordinates (506,412) (c.f. Figure 3.6(a) and 3.7(a)) where a match percentage of 93% was calculated for object size increment 0 (i.e. ***min_size***); therefore this is the solution. Note that 93% is the largest match percentage for all object sizes at the cue location.

3.5 Discussion

The algorithm presented in this chapter has several advantages over existing colour histogram and Backprojection-based methods. These advantages include:

1. This algorithm is capable of modelling both 2- and 3-dimensional planar objects. This is not the case with histogram based techniques. Consider two adjacent sides of a cube with two different colours. At different viewing angles different proportions of the colours exist. Representing such an object using a colour histogram based method is non-trivial.
2. The object size is not required a priori. This algorithm makes the assumption that the cue region is at most 50% occluded and calculates an object size range.
3. Although this algorithm uses a colour histogram intersection metric, it is more stable than Swain's metric because colour bins are based on mean region colour rather than histogram bin counts which are unstable under lighting changes [Mat96].

The results, presented in Table 3.2, show that the object search experiments were successful with 45% of the models being found with a rank of 1, 45% with a rank of 2 and the remaining 10% with a rank of 3. No false negatives where recorded but overall a total of 51 false positives, an average of 8 per experiment. The average reduction in the model database was 68% and the average reduction in search space 53%. These are significant savings using colour information alone especially since many of the database objects had similar colour proportions (c.f. Figure 3.4).

The performance of this algorithm is far better than histogram backprojection methods (c.f. Chapter 2); however, this model can not represent as many objects as Matas' [Mat96] Colour Adjacency Graph. To compare these two methods three criteria are used: object representation, search speed and accuracy. Matas' method can represent 3-dimensional, non-rigid objects which are perspectively distorted; this algorithm can only represent 3-dimensional

planar objects which are affine distorted. Matas recorded a 93% (2 false negatives) accuracy while this method gave 100% (no false negatives); however, Matas only had one false positive result while this method had 51. It is dangerous to judge the performance of the algorithms on this data because the datasets were different and the database used in these experiments contained many objects with similar colours; Matas' however used a model database containing objects with dissimilar colours. In terms of search speed this algorithm performs better because its order of complexity is linear (Matas' is approximately $O(N^3)$). One final point should be made, this algorithm would perform better than Matas' if the object has a low spatial resolution and only one region (for example) could be accurately segmented from the background; since the adopted method only requires a single region for object size calculations while Matas' method requires that more than one of the object regions be identified correctly.

The experiments presented in this Chapter show that colour can be used successfully as a cue when the object is at a low spatial resolution. However, the experiments also show that a more effective object recognition algorithm is required to determine the presence of objects at cued locations. The main failure of this method is that the object recognition algorithm is based on colour proportion; and the model database selected for these experiments contains several objects with similar colour proportions, resulting in a high false positive rate. To alleviate this problem an object recognition algorithm which is not based on colour proportions is required.

An alternative object recognition algorithm is one which models the colour and position of the regions in the object. Such an algorithm is presented in Chapter 4. This algorithm first determines corresponding model and image regions using a region area invariant then calculates the transformation needed to bring the model and image into the same viewpoint (where region matching is performed). One of the important properties of this algorithm is the ability to model disjoint regions — Matas' Colour Adjacency graph method is only

capable of modelling objects with adjacent regions. The object search problem will be revisited in Chapter 5 where a powerful search method based on the Syntactic Neural Network [Luc96] is presented.

Chapter 4

Object Recognition Using Region Colour and Area Ratio Indexing

4.1 Introduction

In Chapter 3, the object recognition part of the object search algorithm recognised objects using the proportion of colours on the object's surface. In this Chapter, however a colour object recognition algorithm is described which instead of using colour proportions, exploits colour region geometry. This algorithm provides greater object discrimination because many objects may have similar colour proportions, but different object region geometry; also, this model can describe objects that can only be represented by a set of disjoint regions.

The proposed algorithm is similar to geometric hashing [GG92][LW88] and the Colour Landmark Model (***CLM***) [WE96][Wal96] (our earlier work), in that it identifies three anchor (or landmark) points in the model and image, calculates a geometric transformation and transforms the model and image into the image viewpoint for matching. The ***CLM*** improved upon the geometric hashing search mechanism by introducing colour and shape constraints on landmark region selection. As a result, a smaller number of affine parameter calculations needed to be performed; also, there was a reduction in the number of model candidates. The problem with the ***CLM***, however is that the moment shape descriptor used increased the computational complexity of the algorithm and was not robust to changes in spatial resolution. Therefore, to improve the speed of the search process, while maintaining the characteristic of constrained landmark selection, a new method called area ratio indexing is proposed; this method utilises both region colour and an area ratio invariant, which is

invariant to affine distortions, to constrain the selection of corresponding model/image region triplets.

The objective of these searches is to determine the occurrences of database models in the image; in this algorithm objects appear affine distorted in the image, e.g. in Figure 4.1 a six region model (with centroids m_0-m_5) is transformed by the affine (shear, rotation, translation and scale) transformation $\boldsymbol{T}$ (a 270° clockwise rotation). In order to calculate $\boldsymbol{T}$, three corresponding model/image region points — called model region triplets and image region triplets, respectively (e.g. $\{m_0, m_1, m_2\}$ and $\{I_0, I_1, I_2\}$ in Figure 4.1) — must be identified so that the model can be transformed into the same viewpoint as the image for region matching. To determine the corresponding model/image region triplets affine invariant area ratios are employed.

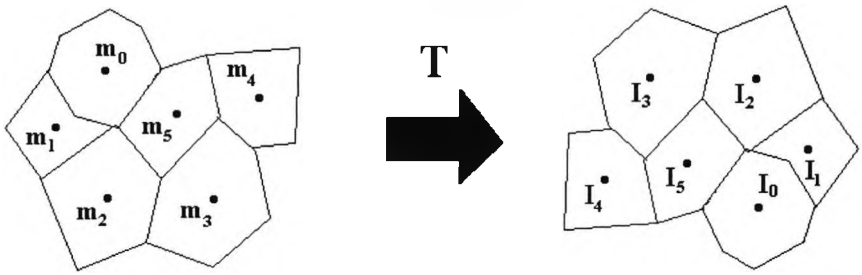


Figure 4.1: A six region model with centroids m_0-m_5 transformed by $\boldsymbol{T}$ (a 270° clockwise rotation) with new region centroids I_0-I_5.

Given an image all possible region triplets are identified and the region areas of each triplet used to calculate two affine invariant indices. These indices are used to access a table (called the area ratio table) entry which contains all model region triplets with similar area invariants. Only those model region triplets with similar colour to the image region triplet are considered. To determine if the model and image region triplets correspond, an affine transformation is calculated to bring the model and object into the same viewpoint where region position and colour determine possible region matches. Transformations are calculated for all the model region triplets (with matching colour) in the table entry. The model which produces the best object match is assumed to be present in the image. This process is repeated for all image region triplets.

The remainder of this Chapter is organised as follows; in Section 4.2 the area ratio table is described, while in Section 4.3 the geometric transformation and model/image region matching are discussed. Section 4.4 presents the algorithm; in Section 4.5 the results and finally in Section 4.6 the discussion.

4.2 The Area Ratio Table

The most difficult part of the recognition problem is determining the corresponding image and model region triplets. In this algorithm, the search is constrained using colour and area ratio indexing. Firstly, an $n \times n$ area ratio table is defined and initialised — this table will contain, in each table entry, model region triplets with similar area ratios (equation 4.1 - 4.4). Before object recognition can take place this table must be filled with the representative model region triplets for each model. These were manually selected. Typically, the user must determine how much occlusion is to be allowed and select the model region triplets appropriately — the fewer chosen for each model the quicker the search process, but robustness to occlusion is decreased.

Model	No. of model regions	Allowed region occlusion	Percentage occlusion of model area	No. of representative triplets
0	5	0	< 10%	4
1	5	1	< 10%	4
2	4	0	< 10%	4
3	10	4	< 30%	12
4	8	3	< 20%	14
5	4	0	< 10%	4
6	4	0	< 10%	4
7	4	0	< 10%	4
8	5	0	< 30%	4
9	5	0	< 40%	4
10	8	3	< 25%	8
11	13	6	< 38%	26
12	8	3	< 25%	8

Table 4.1: The number of regions in each model, the allowed amount of region occlusion, the maximum percentage of the model area that can be occluded and the number of region triplets used to describe each model.

In Table 4.1, the number of model region triplets used to define each database model (illustrated in Figure 4.2) is presented; as well as the number of regions in each model, the maximum number of regions that can be occluded for each model and the maximum percentage of the model area that can be occluded (these are approximated for each model)..

To add a model region triplet (I_1, I_2, I_3) to the area ratio table the region areas are used to calculate the index (i_1, i_2) to the table using equations 4.3 and 4.4 (the regions in the triplet are selected so that region I_1 has the biggest area and I_3 the smallest).

$$r_1 = \left| \frac{area(I_1) - area(I_2)}{area(I_1) + area(I_2)} \right| \tag{4.1}$$

$$r_2 = \left| \frac{area(I_2) - area(I_3)}{area(I_2) + area(I_3)} \right| \tag{4.2}$$

$$i_1 = integer(r_1 * n) \tag{4.3}$$

$$i_2 = integer(r_2 * n) \tag{4.4}$$

In equations 4.3 and 4.4, n is the dimension of the $n \times n$ area ratio table. At table entry (i_1, i_2), a record (containing the attributes: model number, and the model region labels for the regions of the triplet) is added. To allow for possible errors in the area ratio index, records are added to all table entries within a chessboard distance of e (e=2 was selected for these experiments).

Given an image containing an occurrence of a database model, the image must first be partitioned into regions of constant reflectance. This is achieved using the software colour filter algorithm described in Chapter 3. The following process must be repeated for all the image region triplets (all combinations of three regions in the image): The index to the area ratio table is determined using the region areas as described earlier. The model region triplets in this table entry with colours that are similar to the image region triplets are considered as possible corresponding regions. The transformation T is calculated from the centroids of corresponding model and image region pairs

and the model transformed into the same viewpoint as the image. Model/image region matches are recorded if there are image regions close to a transformed model region with the same colour. The number of matches is used to calculate a match measure and if above a predefined threshold an occurrence of the model is assumed to exist at the given image location.

4.3 Region Transformations and Match Function

For each image region triplet the co-ordinates of the bin in the area ratio table are calculated. If there is an entry (record) in the bin then each model associated with the entry might be present in the image. To determine whether a model is present or not, it is necessary to compute the affine transformation parameters using the corresponding pairs of model/image regions. All model region centroids are transformed by these parameters and if an image region with a similar colour exists near (within a Euclidean threshold) the co-ordinates of the transformed model region then a model/image region match is recorded. The total number of these matches determines the object match measure.

Given the centroids of the image region triplet (X_i, Y_i) and the corresponding model region triplet (X_i', Y_i'), the affine transformation parameters $\boldsymbol{a}$, $\boldsymbol{b}$ are estimated using (4.5):

$$\boldsymbol{a} = A^{-1}X, \quad \boldsymbol{b} = A^{-1}Y \tag{4.5}$$

where:

$$X^T = [X_1, X_2, X_3], \quad Y^T = [Y_1, Y_2, Y_3],$$

$$a^T = [a_0, a_1, a_2], \quad b^T = [b_0, b_1, b_2] \text{ and}$$

$$A = \begin{bmatrix} 1 & X_1' & Y_1' \\ 1 & X_1' & Y_1' \\ 1 & X_1' & Y_1' \end{bmatrix}$$

Model region centroids ***(x',y')*** are transformed from model space to image space co-ordinates ***(x,y)*** using the equations $x = \mathbf{A}_1\mathbf{a}$ and $y = \mathbf{A}_1\mathbf{b}$, where $\mathbf{A}_1 =$ [1 x' y'].

A model/image match is recorded if the transformed model centroid is close (in a Euclidean sense) to an image region with the same colour. The match function used should reflect that the greater the number of matches the more likely the object is a model occurrence. A linear function is therefore not adequate. Rather, a match function was selected which gave high match values if most of the model/object regions were matched correctly. This match function was given by:

$$P_1(n',n) = \frac{1}{\sqrt{1 + \frac{1}{4 \cdot \left(\frac{n'}{n}\right)^2}}}$$

$$P(n',n) = \frac{P_1(n',n)}{P_1(1,1)} \qquad (4.6)$$

where n is the total number of model regions and n' the total number of model/image matches (plus the three landmark points). Any match function with these properties can be used.

4.4 The Algorithm

Given a model database, model parameters are generated using Algorithm 4.1 and the object recognition algorithm described in Algorithm 4.2.

Algorithm 4.1: **Model Creation**

1. Apply Hung's [HE95] colour constancy algorithm to the model image.
2. Segment the model image into regions of uniform colour (using the software colour filter algorithm described in Chapter 3) and calculate and store for each region, the region label (id.), its centroid, the number of region pixels and colour (the mean r and g chromaticity co-ordinates).

3. Initialise an $n \times n$ area ratio table.
4. Select (manually) a set of representative model region triplets which adequately represent the model under moderate occlusion; for each of these region triplets calculate the index (i,j) in the area ratio table (using the procedure described in Section 4.2) and add a record entry containing the attributes: model number, and the model region labels for the regions of the triplet.
5. Add the same record entry to all indices within a chessboard distance of e (e = 2 was used) from (i,j)— to allow for errors in calculated bin co-ordinates.
6. Repeat Steps 1-5 for each database model.

The process of recognising an occurrence of a model in an image is described in Algorithm 4.2.

Algorithm 4.2: **Object Recognition**

1. Apply Hung's [HE95] colour constancy algorithm to the image.
2. Segment the image into regions of uniform colour (c.f. Chapter 3) and calculate the centroid, the number of region pixels and colour (the mean r and g chromaticity co-ordinates) for each region.
3. For each image region triplet (A,B and C):
   (a) Calculate the index (i,j) in the area ratio table (described in Section 4.2).
   (b) For each entry in the table (i,j), assume that the model region labels are A',B',C'; there are six possible correspondences between A, B, C and A', B', C', these are {ABC: $A'B'C'$, $A'C'B'$, $B'A'C'$, $B'C'A'$, $C'A'B'$, $C'B'A'$}.
   (c) A region triplet pairing is valid if the colours of the model/image region pairs match. For example for $A'C'B'$, if the colour of region A' is similar to A, C' similar to B and B' similar to C then the pairing is valid.

(d) If ***(r, g)*** and ***(r',g')*** are the chromaticity co-ordinates of an image and model region respectively, then a colour match is recorded if the ***Euclidean(r, g, r', g') < colour_threshold***.

(e) For each valid region triplet pair compute affine transformation parameters and transform all other model regions into image space. A region match is recorded if an image region with the expected colour exists close to (in a Euclidean sense) the computed co-ordinates.

(f) Determine the object match measure from the total number of model/image region matches and equation (4.2).

(g) Repeat step 3.

4.5 Results

In this section, results for the images used in the experiments are presented. These results demonstrate the algorithm's performance under a variety of conditions including: affine object deformity, object occlusion of up to 25% of the model area, and image clutter due to other objects in the scene. It is believed that the experiments presented sufficiently test the algorithm and that most of the model occurrences would be identified correctly. It was assumed that only one occurrence of a given database model may exist in a scene; however, multiple database models may have been present.

The model database used in these experiments, illustrated in Figure 4.2, contains books, playing cards, cereal packages, floor mats and a sun-face model. This is a difficult database because many objects have only four regions, therefore only one region (after selecting the region triplet) would be available for match verification. Notice that although model6 and model7 are represented by the same coloured regions and topology, they are actually two different objects. These types of objects require more expensive geometric techniques (such as surface pattern matching [ED94]) in order to differentiate them.

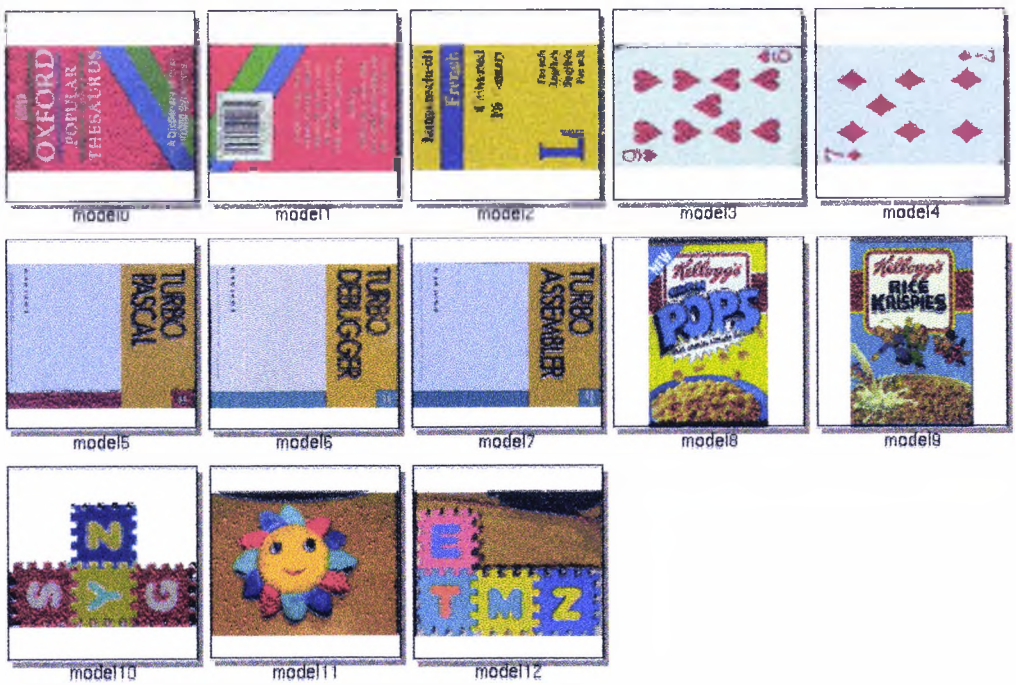


Figure 4.2: The model database.

The 20×20 area ratio table used in these experiments was generated from representative model region triplets (the table could have been generated from all possible model triplets, but this would have made the table much larger, thus increasing object search times). For each model region triplet, record entries were not only added to the calculated bin co-ordinate, but also to all neighbouring bins within a chessboard distance of 2. In so doing, a mismatch of up to 25% was allowed for the image area ratios. Other applications might require a smaller area ratio error which would simply mean a change in table size and the number of bins that region triplets are added. The frequency of the entries in the area ratio table is illustrated in Figure 4.3.

i\j	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	38	38	38	5	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	41	41	41	7	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	47	47	47	10	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	23	23	24	14	4	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
4	20	20	21	14	4	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
5	19	21	22	15	6	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0
6	16	18	19	13	5	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0
7	10	12	13	10	5	4	2	1	1	1	0	0	0	0	0	0	0	0	0	0
8	6	8	8	6	4	3	1	1	2	2	1	1	1	0	0	0	0	0	0	0
9	7	9	9	5	4	3	1	1	3	4	3	3	3	1	0	0	0	0	0	0
10	2	2	2	0	0	1	1	1	3	6	6	6	6	4	1	0	0	0	0	0
11	3	3	3	0	0	1	1	2	4	8	9	9	8	6	2	0	0	0	0	0
12	3	4	4	3	3	3	2	3	4	9	11	11	10	7	2	0	0	0	0	0
13	4	5	6	5	4	4	3	3	5	10	12	12	12	10	5	3	3	2	0	0
14	2	3	4	5	4	4	3	3	4	8	10	10	10	9	5	3	3	2	0	0
15	2	3	4	5	4	4	3	3	5	9	10	11	12	10	6	5	4	2	0	0
16	1	3	5	8	7	7	5	3	4	7	7	9	13	12	9	9	7	3	0	0
17	1	2	4	5	4	4	3	1	3	5	5	7	11	11	9	9	7	3	0	0
18	0	1	2	3	3	3	2	1	1	3	3	5	8	8	6	6	4	1	0	0
19	0	1	2	3	3	3	2	1	1	3	3	5	8	8	6	6	4	1	0	0

Figure 4.3: The frequency of bin entries in the 20x20 area ratio table used in these experiments.

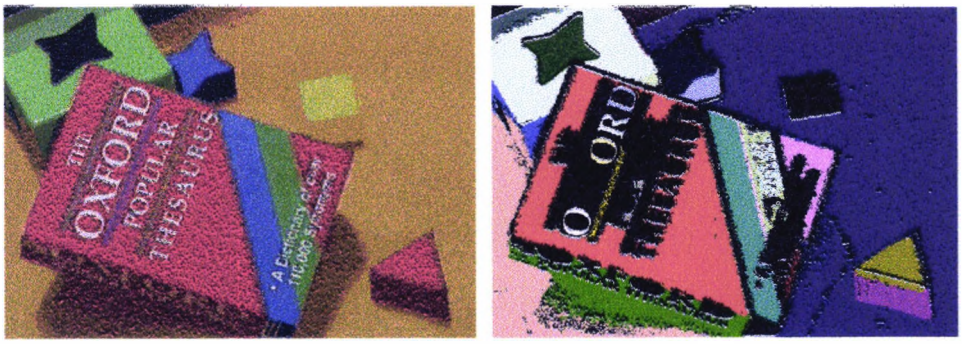
In Table 4.1 the number of regions used to represent each object is presented. In the case of model8 and model9, these regions (c.f. Figure 4.7(e) and 4.7(f)) are selected from the non-textured object regions (which are more resilient to noise and changes in resolution). Region triplets were selected (manually) by occluding different parts of the object and selecting one (or more) triplet from the non-occluded object regions. The number of triplets used for each model is also presented in Table 4.1.

In the case of the models with 4 regions (for example models 2, 5 and 6) no regions were allowed to be absent. This did not mean that object regions could not be partially occluded, rather if regions were occluded then their centroid had to be close to the true region centroid.

The selection and sensitivity of parameters is always an issue in complex image processing algorithms. For this reason a short discussion of these parameters will be included here. There were five important thresholds used, these are:

1. The size of the neighbourhood to use in the area ratio table where model region triplets are added (the parameter ***e*** described in Algorithm 4.4, Step 5).
2. The Euclidean distance threshold between transformed region centroids and image regions.
3. The size of the area ratio table.
4. The threshold used for colour matching.
5. The minimum region size.

The size of the neighbourhood to add model region triplets to is dependent on the application and whether a priori information is available as to the errors in area ratios. In these experiments an error of about 25% was allowed in the area ratios. Selecting the second parameter is difficult because a small value may prevent a valid match from being recorded. This may result from the occlusion of an object region which shifts the region centroid. Conversely, a large value will increase the number of false positives. The size of the area ratio table is also based on the area ratio errors (the discussion of the first parameter is also relevant here). The threshold for colour matching is based on whether models and images are viewed under the same illuminant. If they are then these thresholds can be small. Alternatively, if they are not and a colour constancy algorithm is used then the thresholds selected are based on the quality of match produced by the colour constancy algorithm (which is determined through experimentation). This aspect of the colour constancy algorithm used [HE95] is discussed in Appendix A.2 and Chapter 2. Finally the minimum region size is based on the size of the smallest expected region. Although region areas tend to be fairly robust at reduced resolutions, image noise and sensor errors tend to cause area errors at very low resolutions. For this reason the lowest resolution allowed was restricted.



(a) (b)

Figure 4.4: (a) A test image containing database model0 in the presence of distracters. (b) The labelled image resulting from colour image segmentation using the ***SCF*** (c.f. Chapter 3).

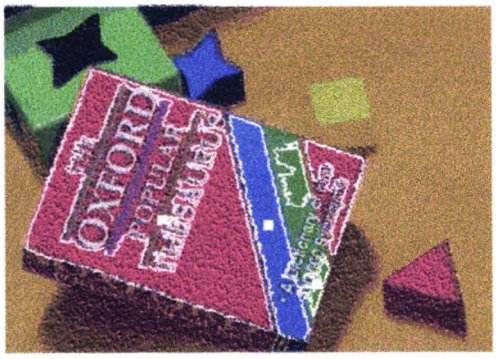


Figure 4.5: The result of applying Algorithm 4.2 to Figure 4.4(a). The regions with a white border are the object regions which have been correctly matched. The solid white squares are the centroids of the anchor (landmark) points used (c.f. Table 4.4) to calculate the affine transformation parameters. Notice that all five object regions have been identified correctly despite segmentation errors.

The first test image, illustrated in Figure 4.4(a), contains database model0 at high resolution and affine deformed in a scene with distracters. Figure 4.4(b) illustrates the results of the ***SCF*** [WE96] partial image segmentation algorithm (using a colour histogram size of 25×25 — which is used for all the images presented) using a minimum region size of 400 pixels (this choice was arbitrary). There were 27 regions resulting from the segmentation.

Region Label	Centroid (x_c,y_c)	# of Region Pixels	*g*	*r*
0	(225,234)	108569	0.13	0.71
1	(444,194)	29487	0.28	0.14
2	(505,160)	17563	0.40	0.36
3	(544,68)	12589	0.13	0.73

Table 4.2: The region parameters for database model0. The parameters ***r*** and ***g*** are the mean chromaticity co-ordinates of the region.

The region parameters for model0 and the segmented image are presented in Table 4.2 and Table 4.3, respectively. These image parameters are passed to Step 3 of Algorithm 4.2. As seen in Table 4.1 there were four representative triplets for this model (model0); these triplets are (region labels): {0,1,2}, {0,1,3}, {0,2,3} and {0,1,4}.

Region Label	(x_c,y_c)	Region Area	*g*	*r*
0	(416,118)	3387	0.41	0.39
1	(38,148)	1549	0.56	0.23
2	(351,222)	1923	0.49	0.19
3	(359,210)	1696	0.55	0.20
4	(196,266)	32353	0.13	0.72
5	(412,215)	3405	0.13	0.72
6	(150,246)	856	0.21	0.45
7	(108,282)	522	0.30	0.36
8	(470,175)	93455	0.38	0.50
9	(172,384)	11773	0.22	0.74
10	(328,273)	9567	0.26	0.14
11	(388,346)	723	0.55	0.20
12	(523,380)	3201	0.22	0.74
13	(525,339)	3617	0.13	0.72
14	(184,256)	421	0.13	0.72
15	(144,189)	511	0.31	0.36
16	(340,233)	634	0.37	0.12
17	(51,327)	16950	0.38	0.53
18	(156,156)	529	0.31	0.36
19	(169,121)	526	0.31	0.36
20	(275,107)	1323	0.45	0.36
21	(206,96)	424	0.56	0.22
22	(93,66)	14441	0.48	0.21
23	(249,77)	3229	0.25	0.14
24	(89,56)	3781	0.45	0.36
25	(33,15)	1473	0.56	0.22
26	(199,12)	1364	0.45	0.36

Table 4.3: The region parameters for the 27 regions segmented from the image of Figure 4.4(a).

A correct match for model0, with a match measure of 1.0 (c.f. Figure 4.5), was recorded when Algorithm 4.2 was applied to Figure 4.4(a).

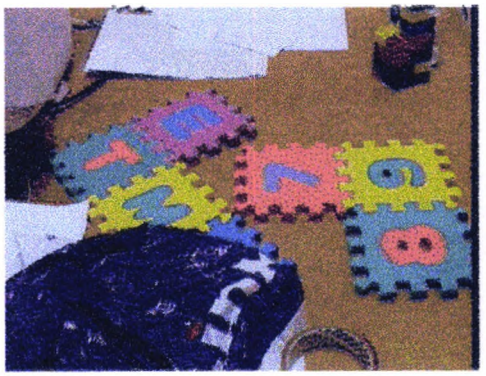
The co-ordinates of the model/image region triplets used to calculate the affine transformation parameters, the transformed model centroids and the distance errors between transformed model and image centroids are presented in Table 4.4. The computed affine parameters are $\boldsymbol{a}$ = {95.74, 0.58, -0.13} and $\boldsymbol{b}$ = {38.78, 0.18, 0.80}.

Model	Image	Transformed	Distance Error	Landmark
(225,234)	(196,266)			Yes
(444,194)	(328,273)			Yes
(458,92)	(359,210)	(349,193)	20.55	No
(544,68)	(412,215)	(402,190)	27.71	No
(562,259)	(388,346)			Yes

Table 4.4: The parameters for the solution of the first recognition experiment. The "transformed" column represents the co-ordinates of the transformed model region centroids (into image space) using the affine parameters calculated from the model/image region triplet match. The "distance error" column is the Euclidean distance between the transformed model centroid and the closest image region.

The computational expense of this search when compared with a method based on exhaustive testing of the model and image region triplets is an important consideration when rating the performance of the method. For the image in Figure 4.4(a), affine transformation parameters were calculated 70 times and region centroids transformed 272 times. This compares with 1,442,350 and 10,000,900 respectively required by the exhaustive method. The only overheads of this method are the image area ratio and colour matching calculations.

The second experiment used the image in Figure 4.6(a) which contains an affine deformed and occluded model12. The model was identified correctly with a match measure of 0.97 as illustrated in Figure 4.7(a). Model12 was also identified correctly in Figure 4.6(b) with a match measure of 1.0 (c.f. Figure 4.7(b) and Table 4.5).

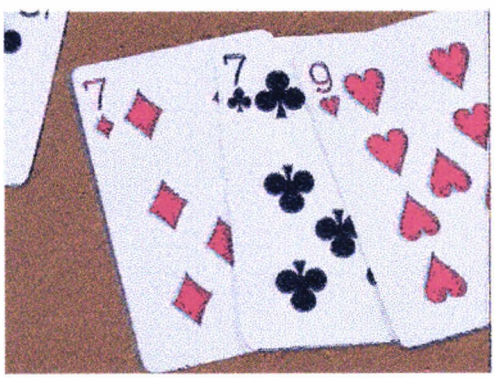


(a)

(b)

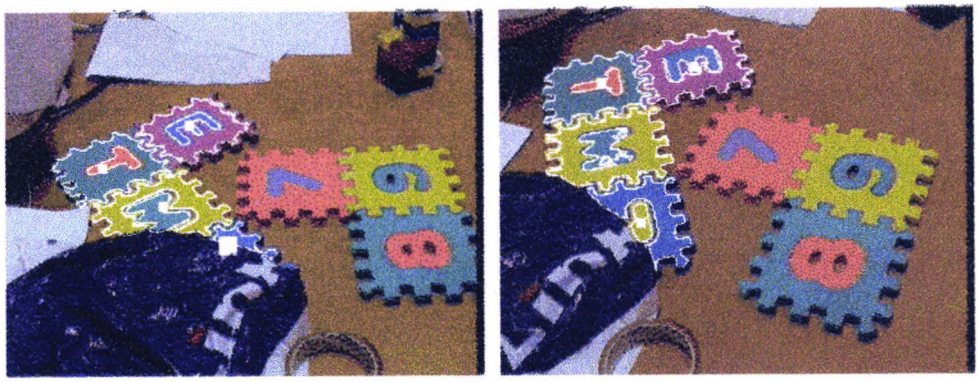
(c)

(d)

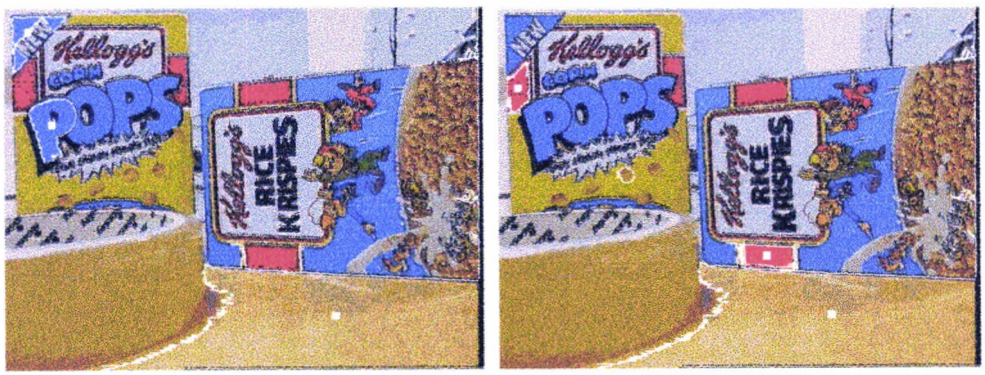


(e)

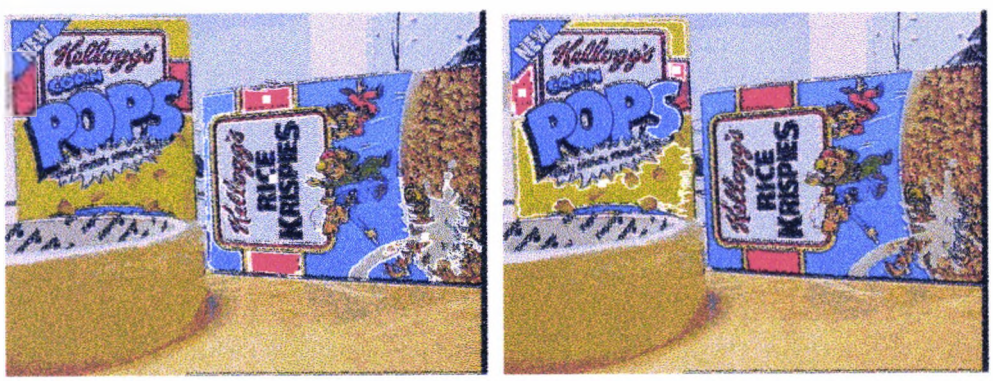
Figure 4.6: The images used in the recognition experiments. (a) Contains an affine deformed and occluded model12 in a cluttered scene. (b) Contains an affine deformed and occluded model12 in a cluttered scene. (c) Contains an occluded model8 and non-occluded model9. (d) Contains a non-occluded model8. (e) Contains an occluded model3 and model4.



(a) (b)



(c) (d)



(e) (f)

(g)

Figure 4.7: The results for the images illustrated in Figure 4.6. (a) the larger white squares indicate the position of the expected region.

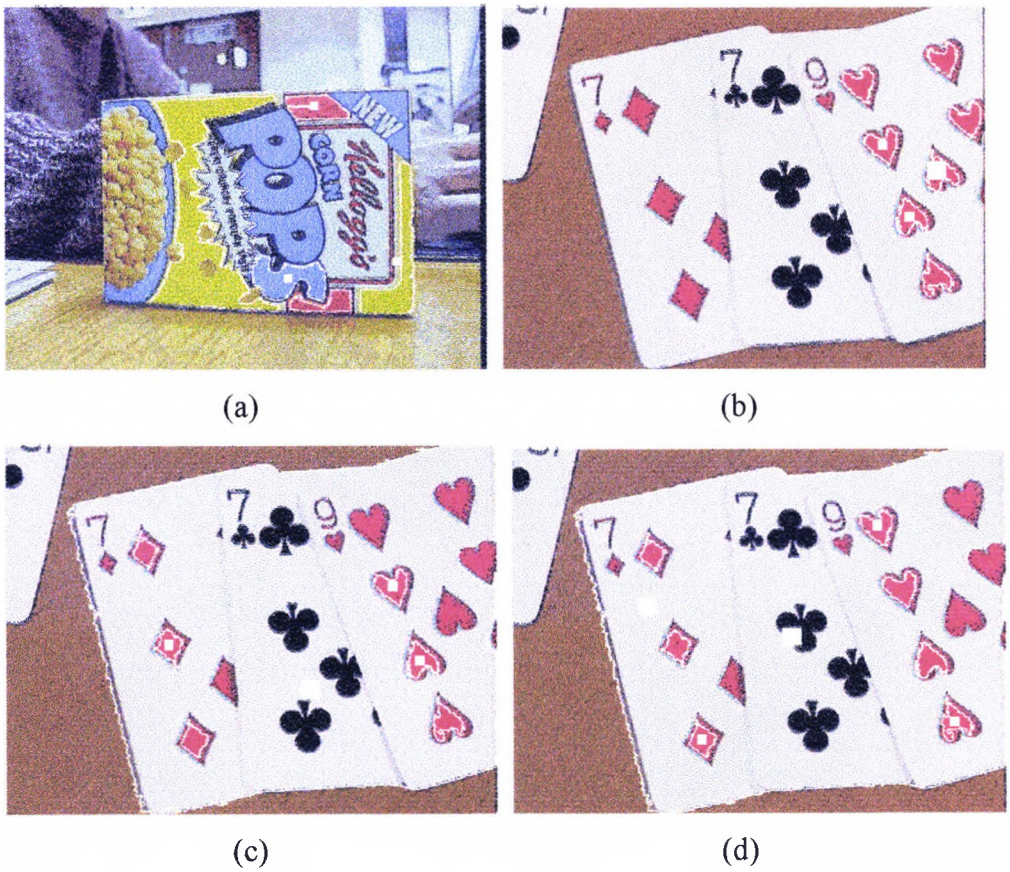


Figure 4.8: The results for the images presented in Figure 4.6.

Model8 and model9 contain both textured and non-textured regions making them more difficult to represent. However, as illustrated in the results (c.f. Figure 4.7 and 4.8) these types of models can be adequately represented by some of their non-textured regions. Figure 4.6(c) contains a non-occluded model9 and an approximately 25% occluded model8. Both model8 (Figure 4.7(f)) and model9 (Figure 4.7(e)) were identified correctly, however there were two mismatches, model5 (Figure 4.7(d)) and model6 (Figure 4.7(c)). Figure 4.6(d) contains a non-occluded model8 which was correctly identified (c.f. Figure 4.8(a)), however there was a mismatch for model2 (c.f. Figure 4.7(g) and Table 4.5).

Quite often there is more than one candidate solution for a given model from which the best candidate must be chosen. This selection process considers both the match measure, as well as the distance errors of each non-landmark point.

If two candidates have different match measures then the candidate with the highest match measure is assumed to be the best. Alternatively, if both match measures are the same then the candidate with the smaller distance errors is assumed to be the best choice.

Finally, matches for model3 and model4 were reported for Figure 4.6(e), however two of these three results are mismatches, one for model3 (Figure 4.8(d)) and one for model4 (Figure 4.8(c)). Model4 was too occluded to be recognised since only a maximum of three regions could be occluded at a time. On the other hand, the best candidate for model3 (Figure 4.8(d)) has a match measure of 0.95, however the correct match (Figure 4.8(b)) only had a match measure of 0.91.

Image	# of image regions	Model found	Match measure	Correct match
Figure 4.6(a)	38	model12	0.97	yes
Figure 4.6(b)	34	model12	1.00	yes
Figure 4.6(c)	61	model6	1.00	no
		model5	1.00	no
		model9	1.00	yes
		model8	1.00	yes
Figure 4.6(d)	45	model2	1.00	no
		model8	1.00	yes
Figure 4.6(e)	19	model3	0.91	yes
		model4	0.97	no
		model3	0.95	no

Table 4.5: The results for the recognition experiments using the images in Figure 4.6.

The seemingly high match values for the incorrect matches of model5 and model6 in Figure 4.6(c) and model2 in Figure 4.6(d) result from the fact that each of these regions are only represented by 4 regions. That means that only one region was used for match verification. This increased the likelihood of false positive matches.

4.6 Discussion

In this chapter a method of representing objects using the colour and area ratios of regions was described. It was shown that by introducing these constraints significant computational savings, over an exhaustive method, can be realised. Only one false negative result was recorded and 5 false positive results; a maximum of approximately 25% occlusion was allowed. Even more significant is that objects with a combination of textured and non-textured regions (that could not be easily represented by models such as the region adjacency graph [Mat96]) can also be represented by this model. This has implications for image retrieval and retrieval using hand drawn picture applications. In the case of image retrieval, complicated (textured etc.) objects can be represented by this model and retrieved quickly. Also, unlike many image retrieval systems, the algorithm works well in the presence of cluttered scenes and varying resolutions. In the case of retrieval using hand drawn pictures or even by natural language descriptions, this method also seems appropriate. Images may be naturally described not only by their colour content, but also by approximate size ratios of regions and distances, e.g. "The image contains four (salient) regions, one green, one blue one yellow and one black. The green region is the same size as the blue and black one, but twice as large as the yellow one." These types of descriptions, although detailed, are quite natural.

When compared to methods such as colour histogram intersection-based, statistical-based and colour region adjacent graph methods (c.f. Chapter 2 for an overview of these methods) several points can be made. Firstly, colour histogram-based methods can represent both textured and non-textured objects easily. The adopted method requires that the objects be partitioned into regions which limit those objects it can represent. However, the adopted algorithm works well in the presence of image clutter (colour histogram-based algorithms do not). Statistical-based methods have a distinct advantage over all these methods in that they represent images with a few floating point numbers. The adopted method requires that five parameters be stored for each

model region — the area ratio histogram must be stored as well. The range of conditions that this algorithm will work in is less than Matas' colour adjacency graph which is capable of representing 3-dimensional deformable objects which are perspectively distorted. The algorithm adopted here is limited to 3-dimensional planar and rigid objects which are affine distorted. However, this algorithm can model objects which can only be represented by a set of disjoint regions, a significant advantage over Matas' colour adjacency graph which is based on region adjacency.

The area ratio invariant described in this chapter is merely one of the invariants that could be used in the creation of an invariant table. Other useful invariants include: the ratio of distances between a region triplet, which is a Euclidean invariant; and the cross ratio which is perspective invariant. The formulation of the table would be the same as the area ratio table where the invariant serves as the index into the table.

Chapter 5

Colour Object Search Using a Modified Syntactic Neural Network

5.1 Introduction

So far in this thesis two colour-based algorithms have been presented, one for object search and the other for object recognition. In the object search algorithm, the topology of the object regions was not really considered (remember however that the search was made for object regions that were spatially close, so in a loose sense adjacent). In the object recognition algorithm, the geometry of the object regions was modelled using a rigid object model which was limited to planar non-deformable models. In none of these methods has the topology of object regions been exploited. Topological relationships such as adjacency, enclosure, in between, and near, share the property of invariance to perspective transformations, which was not achieved with the other models (only affine invariance). Another important point about the algorithms presented so far is that they do not produce solutions in a best-first order therefore search time is increased. This Chapter addresses these issues and presents an object search algorithm which is capable of searching unconstrained indoor environments with good success.

In this Chapter, an object search algorithm is presented which exploits region colour and topology and produces solutions in a best-first order. This algorithm accepts as input an image partitioned into regions of uniform colour characteristics; it then produces an image region relationship table (image ***RRT***) which contains the relationships between the image regions. The region relationships considered are: adjacency, enclosure and disjoint (not enclosed and not adjacent). Each model in the model database is described by the colour

of its regions (chromaticity co-ordinates are used as described in the previous algorithms) and a model region relationship table (model ***RRT***). Several methods could have been used to search for valid region combinations, for example the interpretation tree [Gri90] or hash tables; however, a modified syntactic neural network (***SNN***) [Luc96] was used because results were produced quickly and in a best-first order — although one disadvantage of the ***SNN*** is that it normally requires large amounts of memory.

The remainder of this Chapter is organised as follows: in Section 5.2, Lucas' ***SNN*** is described and in Section 5.3 the modifications required by the adopted object search algorithm are discussed. In Section 5.4 the object search algorithm is described, Section 5.5 the results from the experiments performed are described and finally, in Section 5.6 the method is discussed.

5.2 The Syntactic Neural Network

Lucas [Luc96] described a neural architecture based on context-free grammars called ***syntactic neural networks*** which concatenates symbols (e.g. ***A*** and ***B*** concatenated is ***AB***) to form larger strings. Simple grammar fragments (e.g. **AC** or **BD**) are parsed by ***Local Inference Machines*** (***Lims***) which perform a lazy best-first evaluation of the Cartesian product of two ranked lists. For example given two lists {***A***,***B***} and {***C***,***D***} the Cartesian product is the list {***AC***, ***AD***, ***BC***, ***BD***}. Retrieving these pairs in a best-first ranked order would normally require the probability (or rank) of each of ***AC***, ***AD***, ***BC***, ***BD*** to be determined (the probability of each pair is assumed to be the product of the individual symbol probabilities) and the list sorted in decreasing order of probability. In the ***SNN***, however a lazy evaluation method is used which identifies the next best symbol without computing all the probabilities and sorting. This process is performed by the ***Lims***.

Figure 5.1 illustrates a typical ***SNN*** with four inputs —the number of inputs determines the number of ***Lims*** in the Network. At each input bin a classifier assigns a rank (or probability) to each symbol based on some symbol feature.

The symbols at each input are then sorted, placing the symbol with the highest rank at the top of the list (e.g. ***A*** has the highest rank in the leftmost input bin of Figure 5.1). Each ***Lim*** outputs strings of concatenated symbols in best-first order. This process is effected at each level of the ***SNN*** until the final string is output from the ***Root Lim***. The strings expected to be output from the ***SNN*** in Figure 5.1 are {***ACE***, ***ADE***, ***BCE***, ***BDE***, ***ACF***, ***ADF***, ***BCF***, ***BDF*** }where their order is dictated by the probabilities of the symbols.

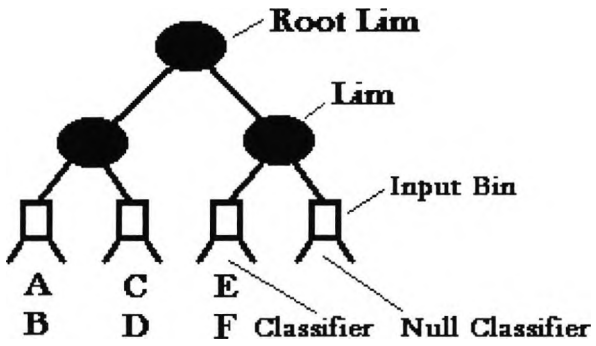


Figure 5.1*:* The Structure of the Syntactic Neural Network

In order to describe the lazy evaluation process used by the ***Lims***, assume that the probabilities assigned to the symbols in Figure 5.1 are **A** (0.9), **B** (0.6), **C** (0.8), **D** (0.7), ***E*** (0.8) and ***F*** (0.5); and consider Figure 5.2.

	C (0.8)	**D** (0.7)
A (0.9)	0.72	0.63
B (0.6)	0.48	0.42

Figure 5.2*:* The Lazy Evaluation Process

The purpose of the lazy evaluation is to return the symbol strings in best-first order without calculating the product of each set of symbols and sorting. Lucas [Luc96] devised a set of rules to accomplish this. Considering the table in Figure 5.2 — where the rows and columns of the table are labelled with the symbols input to the ***Lim*** — the top left-hand table entry and the bottom right-hand table entry contain the symbol strings with the largest (***AC***) and smallest (***BD***) probabilities, respectively. Subsequently, Lucas identified the table

entries from which the next best symbol string must come (see [Luc96] for details). As a result products only need to be calculated for these table entries and the symbol string with the largest probability selected. This process is continued until all symbol strings are retrieved.[1] The order that the symbol strings are retrieved from this ***Lim*** are ***AC*** (0.72), ***AD*** (0.54), ***BC*** (0.48) and ***BD*** (0.42).

There are two output strings from the rightmost ***Lim***, these are ***E*** (and ***NULL***) and ***F*** (and ***NULL***). The evaluation and order of output of the symbol strings from the ***root Lim*** is illustrated in Figure 5.3. Note that initially the table is empty, then when a request is made to output a symbol string from the ***root Lim*** it gets the next best symbol string from the lower left and right ***Lims*** (c.f. Figure 5.3 (a)).

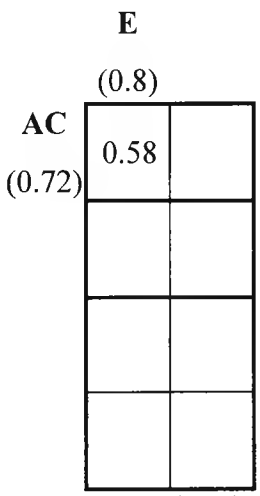


Figure 5.3 (a)*:* The ***root Lim***, illustrating the evaluation process for the retrieval of the first symbol string ***ACE*** (0.58).

Figure 5.3 (a) illustrates an important point, that is all symbols are not needed at higher level ***Lims*** to produce the next valid symbol string. To retrieve the next best symbol string another symbol string must be retrieved from the left and right lower level ***Lims***, these are ***AD*** and ***F***. Using Lucas' rule [Luc96] the next best symbol string must be either ***ADE*** (0.43) or ***ACF*** (0.36). Figure 5.3 (b) illustrates the evaluation and retrieval of the second symbol string ***ADE*** (0.43).

[1] In the experiments performed in this work, the rules used in lazy evaluation [Luc96] were found to be reliable.

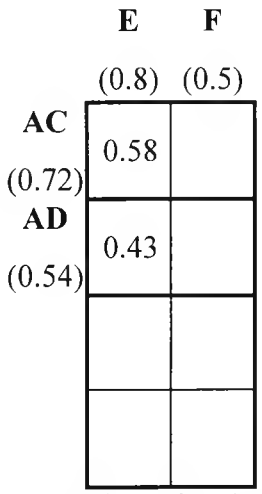


Figure 5.3 (b): The ***root Lim***, illustrating the evaluation process for the retrieval of the second symbol string ***ADE*** (0.43).

The next best symbol string must be either ***BCE*** (0.38) or ***ACF*** (0.36). Two points are worthy of note here, the first being that only the next best symbol string ***BC*** (0.48) from the leftmost lower Lim must be retrieved and the probability for ***ACF*** has already been calculated (c.f. Figure 5.3 (c)).

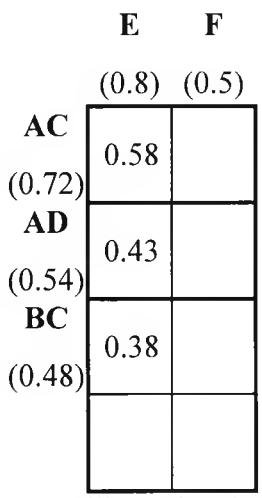


Figure 5.3 (c): The ***root Lim***, illustrating the evaluation process for the retrieval of the third symbol string ***BCE*** (0.38).

The order of retrieval of the remaining symbol strings are ***BDE***, ***ACF***, ***ADF***, ***BCF*** and ***BDF***.

5.3 The Modified Syntactic Neural Network

To adapt Lucas' ***SNN*** to the colour object search problem, a number of modifications had to be made including: the addition of symbol (region)

relationship constraints at the ***Lim*** level, region connectivity constraints at the ***SNN*** output and limiting the number of regions in each input list.

5.3.1 The *SNN* Input Requirements

When a search is being performed for database models in an image, a ***SNN*** is generated for each database model. Each symbol represents an image region and the number of ***Lims*** in the ***SNN*** is dictated by the number of model regions. In an attempt to keep the ***SNN*** balanced, the number of inputs must be equal to 2^L where L, an integer, is the number of levels in the ***SNN*** structure. For each model, L must be selected so that the number of model regions $\leq 2^L$. For example, a model with five regions would use an eight input ***SNN*** and assign ***NULLs*** to the three unused inputs. To each input a model region is assigned. Image regions are then associated with one or more ***SNN*** inputs if their rank is less than a predefined threshold, equation (5.1) where: (r,g) is the chromaticity co-ordinates of the model region assigned to the ***SNN*** input and (r',g') the chromaticity co-ordinates of the image region. This rank also determines the position of the region in the list (the region with the highest rank is placed at the top of the list, and the smallest at the bottom). To limit the size of the resulting ***SNN*** a limit is placed on the number of image regions per list.

$$rank = \frac{Euclidean(r,g,r',g')}{\sqrt{2}} \qquad (5.1)$$

The upper and lower bound for rank are $0 \leq rank \leq 1$. (Note that $\sqrt{2}$ is the maximum Euclidean distance between two points in the chromaticity colour space — at opposite ends of the hypotenuse of the chromaticity triangle.)

Quite often some object regions are absent due to occlusion or image noise, therefore the ***SNN*** design must facilitate object search in these conditions. The method employed is to add to the bottom of each input list a blank label. For example if we add blank labels to the four inputs in Figure 5.1 then outputs such as *A*[BLANK][BLANK][BLANK] (*A*) or *A*[BLANK]*F*[BLANK] (*AF*),

which represent occluded objects, are possible. By adding blanks to the inputs, the string [BLANK][BLANK][BLANK][BLANK] will also be returned by the network and is ignored. The choice of the rank of the blank symbol is arbitrary. A high rank value means that strings representing occluded objects will be output first. A small rank value means that these strings will be output last. In this algorithm, a small rank is chosen for the blank symbol.

At the input level of the ***SNN*** where image regions are being paired it is important, where possible, to assign pairs of model regions with a non-disjoint relationship to each ***Lim***. This is justified because in an image more region pairs are expected to be disjoint than say adjacent (a non-disjoint relationship) and therefore less pairs will proceed to the next level of the ***SNN***. This means that the size of the next ***SNN*** level can be smaller than the size defined by Lucas (this is important when considering the memory required by the ***SNN***).

5.3.2 Lim Restrictions

At the ***Lim*** level of the ***SNN***, regions are combined and returned in best-first order; however, a symbol string validation step is introduced. Consider two symbols ***A*** and ***B*** input from the left and right input list of a given ***Lim***; the symbol pair ***AB*** is only valid if ***AB*** has the same topological relationship as the topological relationship of the model region pair assigned to the ***Lim*** inputs. This reduces the number of symbol combinations and thus the complexity of the search. In addition, all symbol strings output from a given ***Lim*** must contain only one occurrence of each symbol — it is possible for multiple symbol occurrences because the same symbol can be in more than one list.

5.3.3 *SNN* Output Restrictions

There are four restrictions placed on the output of the ***SNN***:

1. A symbol string output from the ***SNN*** is only valid if it contains at least three regions. Three regions were selected because it was observed that in

images there are often multiple occurrences of valid region pairs (valid in terms of colour and topology), but valid region triplets occur less often, thus are more likely the object.

2. The vote assigned to each ***SNN*** output, which is based on the number of valid region relationships found in the object (discussed in detail in Section 5.3.4), must exceed a predefined threshold.
3. A connectivity constraint is placed on output regions. This constraint ensures that the regions in each ***SNN*** output are connected either explicitly or implicitly (e.g. object ***ABC*** is accepted if ***A*** adjacent ***B***, ***B*** adjacent ***C*** and ***A*** disjoint ***C***, but not if ***A*** adjacent ***B***, ***B*** disjoint ***C*** and ***A*** disjoint ***C***).
4. Finally, a limit is placed on the number of outputs considered (which satisfy conditions 1-3). From the set of valid outputs the one with the highest vote is assumed to be an object occurrence. If it is assumed that the object can occur more than once then all outputs with high votes are considered an occurrence of the object.

As a result of point 1 only objects with three regions or more are represented. Note however that by simply removing this restriction, objects with any number of regions can be represented by this method.

5.3.4 Voting Scheme

To determine a rank for the objects output from the ***SNN***, a voting scheme was devised. In this scheme each topological relationship used in the model ***RRT*** is assigned a vote. The value of this vote is dependent on the importance of the relationship. In Table 5.1 the votes assigned to the topological relations used (adjacency, enclosure and disjoint) are presented.

Topological Relationship	Vote
Disjoint	1
Enclosure	20
Adjacency	20

Table 5.1: The votes assigned to each of the topological relationships disjoint, enclosure and adjacency.

As presented in Table 5.1, the disjoint relationship has the smallest vote (i.e. 1), because disjoint regions do not help to discriminate between objects as much as adjacent or enclosed ones. Although a vote of 1 (rather than say 5) seems small, it is significant when objects have several disjoint region pairs. In the experiments performed the votes for adjacency and enclosure were the same, but this does have to be the case in other applications.

The criteria used to determine if a region group output from the SNN was valid was based on the vote exceeding a predefined threshold. Since a minimum of three regions (c.f. Section 5.3.3) were allowed then a minimum vote value of 40 was selected, i.e. at least two non-disjoint region relationships must exist. This vote however does not have to be fixed across the database (although that was done in these experiments), rather it could be dependent on the model. As a result, different amounts of occlusion would be allowed for each model.

5.3.5 Memory Considerations

If some simple calculations are performed it is realised how large the memory requirements are for the ***SNN***. For example, an 8 input ***SNN*** which allows a maximum of 10 symbols at each input will require, for the input level ***Lims*** 100 probabilities (floating point numbers) per ***Lim***, therefore a total of 400 probabilities. At the next level 100*100 = 10000 floating point numbers (per ***Lim***) and the ***root Lim*** 10000 * 10000 = 100 million floating point numbers. Considering that a floating point number is four bytes long, the ***SNN*** will require of the order of 400MB! This memory requirement is further increased when the number of inputs grow.

In the modified ***SNN***, not all symbol pairs (groups of two, four, eight etc.) proceed to the next ***SNN*** level; therefore the size of higher levels need not be as large as described above. For example, if an 8 input ***SNN*** was considered

with a maximum of 40 inputs, then the four input level ***Lims*** will require of the order of 26K. If all the other ***Lims*** are assumed to be the same size, say 1000x1000 then the remaining three ***Lims*** would require of the order of 3MB! This is much less than the 400MB required by Lucas' ***SNN***. It is up to the user to select an appropriate node size for his application.

The memory requirements could be further reduced by replacing the floating point numbers with integers (0-65535) or characters (0-255) and assigning integer ranks to the symbols rather than probabilities. This would reduce the memory requirement by half or a quarter. An alternative to these schemes is a dynamic node size which changes as the need requires. This however would make the ***SNN*** more difficult to program. Considering the price of RAM today, the cost of storing the modified ***SNN*** is quite low.

5.4 The Algorithm

Before object search can be performed, a model database must be created. For each model, the colour of each region must be stored, as well as a model ***RRT***. To generate the set of representative regions the model image is segmented and regions greater than a minimum size selected. The model in Figure 5.4 contains the French Universal Dictionary model which is described by four regions, two yellow and two blue.

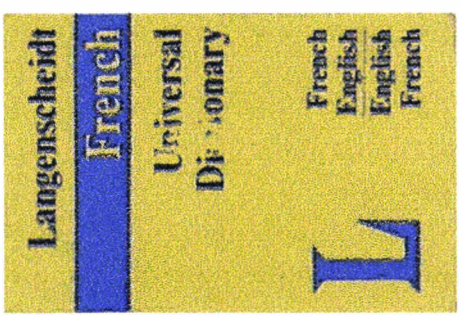


Figure 5.4: A database model.

Table 5.2 presents the region labels and colours for each of these regions. The yellow region on the far left has a label 2; the blue region adjacent to it a label

3; the large yellow region, a label 0 and the enclosed region (the blue 'L') a label 1. It is useful to compare this model representation with the one in Chapter 3, which stores parameters (five floating point numbers, therefore 20 bytes) for each model colour rather than model region. For the same model (Figure 5.4) the representation in Chapter 3 requires 40 bytes (since there are two colours), while this representation requires only 38 bytes (eight floating point numbers for the region colours (32 bytes) and 6 bytes for the ***RRT***).

Label	r	g
0	0.48	0.46
1	0.14	0.24
2	0.47	0.46
3	0.14	0.24

Table 5.2: The parameters for the model in Figure 5.4.

The ***RRT*** for this model is illustrated in Figure 5.5. The contents of the table is read "region <table row> is related to region <table column>"; for example, region 3 is enclosed by region 0. A total of six bytes is used to store this table since only the lower triangular part is required and each of the four relationships is encoded into the bits of a byte (Adjacent: bit 1; Disjoint: bit 2; Encloses: bit 3; and Enclosedby: bit 4).

	0	**1**	**2**	**3**
0		disj	adj	encl
1	disj		adj	disj
2	adj	adj		disj
3	encby	disj	disj	

Figure 5.5: The region relationship table for the model illustrated in Figure 5.4. The symbols "adj", "encl", "encby" and disjoint represent the adjacent, encloses, enclosedby and disjoint relationships, respectively.

Although, the model in Figure 5.4 is 2-dimensional the same method is used to represent three dimensional objects, that is extract the object regions, determine their colour, then create a model ***RRT***. An example of a search for two 3-dimensional models is presented in Experiment 3.

The object search algorithm which utilises the modified ***SNN*** is described in Algorithm 5.1.

Algorithm 5.1: **Colour object search using the modified *SNN***

1. Apply Hung's [HE95] colour constancy algorithm to the input image.
2. Partition the image into regions of constant reflectance (using the ***SCF*** in Chapter 3).
3. For each model create an ***SNN***. Assign each image region to the ***SNN*** input (or inputs) where the colour rank (equation (5.1)) is greater than ***colour_threshold***. The colour rank also determines the position of the region in the list.
4. Retrieve the next combination of regions from the ***SNN*** and calculate its vote.
5. If this combination of symbols satisfies the output requirements (Section 5.3.3) then add to the results set and repeat Step 4 until the required number of solutions are found.
6. Repeat Steps 3-5 for all database models.

5.5 Results

Three object search experiments were performed using the search mechanism described in this Chapter. The first experiment demonstrated the performance of the method over a six image test set. These images varied in model occurrences, viewpoint, illumination, amount of occlusion and spatial resolution. In this experiment it was expected that most of the objects would be located correctly despite the complexity of the scenes.

The second experiment used a single image to assess the robustness of the method to changes in the colour threshold used at the ***SNN*** inputs, the maximum number of ***SNN*** inputs and the minimum region area. These

parameters were modified and the number of false positives and correct matches recorded. It was expected that the number of false positives and matches would remain constant throughout these parameter variations.

The final experiment was used as an "acid" test, that is the search for two 3-dimensional objects in a real (and natural) environment. It is important to note here that although all the images used in these experiments were from real environments, the placement and selection of objects was not necessarily natural. In this experiment, both the placement and selection of the objects being sought in the room were natural. Also, these objects occupied low spatial resolutions. This was therefore considered the definitive test which if passed would satisfy most of the requirements for the object search algorithm presented in Chapter 1.

Each of these experiments were performed using images of size 634x478 and 766x574 captured with an NTSC camcorder and PAL colour camera, respectively. The algorithm was run on a 100MHz Pentium with only 16MB RAM running LINUX. It is important to note that about 12MB of RAM was used by the operating system and other software that was running on the machine, therefore only about 4MB of RAM was being used. A 16MB swap was being used throughout the running of the algorithm. The approximate speeds recorded in the results are based on this machine, therefore much faster performance can be expected from a better machine with more RAM. Today, Pentiums running at speeds of over 200MHz with at least 32MB of RAM are commonplace.

5.5.1 Parameter Selection and Sensitivity

Before discussing the three object search experiments it would be useful to discuss the parameters used in the algorithm, how they were selected and their sensitivity. A select list of important parameters, their values and a description

of their sensitivity are presented in Table 5.3. In Experiment 2 the effects of varying three of these parameters was illustrated.

Parameter	Value	Sensitivity Discussion
Min. model region area	Model based	Larger regions tend to be more resilient to changes in spatial resolution.
Min. image region area	100-400 pixels	Most object regions must be greater than this threshold otherwise the search will fail.
Max. no. of *SNN* inputs	10-40	Smaller values are better provided that the object regions are present in the appropriate input lists. Larger values mean more memory is required for the *SNN*. With poor colour matching, longer input list will help to ensure that object regions are in the list, resulting in successful object searches.
Size of *SNN Lims*	650	Small values can lead to the SNN running out of memory; can occur especially when the image contains a large number of false positives or when colour matching is poor. Larger values means that more computer memory is required.
Min. output vote	40	Increasing this value reduces the amount of allowed region occlusion; decreasing it results in more false positives.
Max. no of *SNN* outputs (the rules in Section 5.3.3 are applied to these outputs)	200	Setting this value too small may cause the correct result to be missed; especially when correct regions are far down in the input list. Larger values mean that the termination point when the object is not present is delayed.
Colour rank threshold (Eqn. 5.1)	0.16 - 0.48	If too small correct regions may not be included in the input lists and the search may fail. If too large then the number of regions per list increases and the number of false positives may be higher.

Table 5.3: A select list of parameters for the object search algorithm, their values and a description of their sensitivity.

5.5.2 Experiment 1

In this experiment a search was made for 13 database models in five images. The model database used contained 2-dimensional objects such as books, parts of a floor mat and a Christmas card box as illustrated in Figure 5.6.

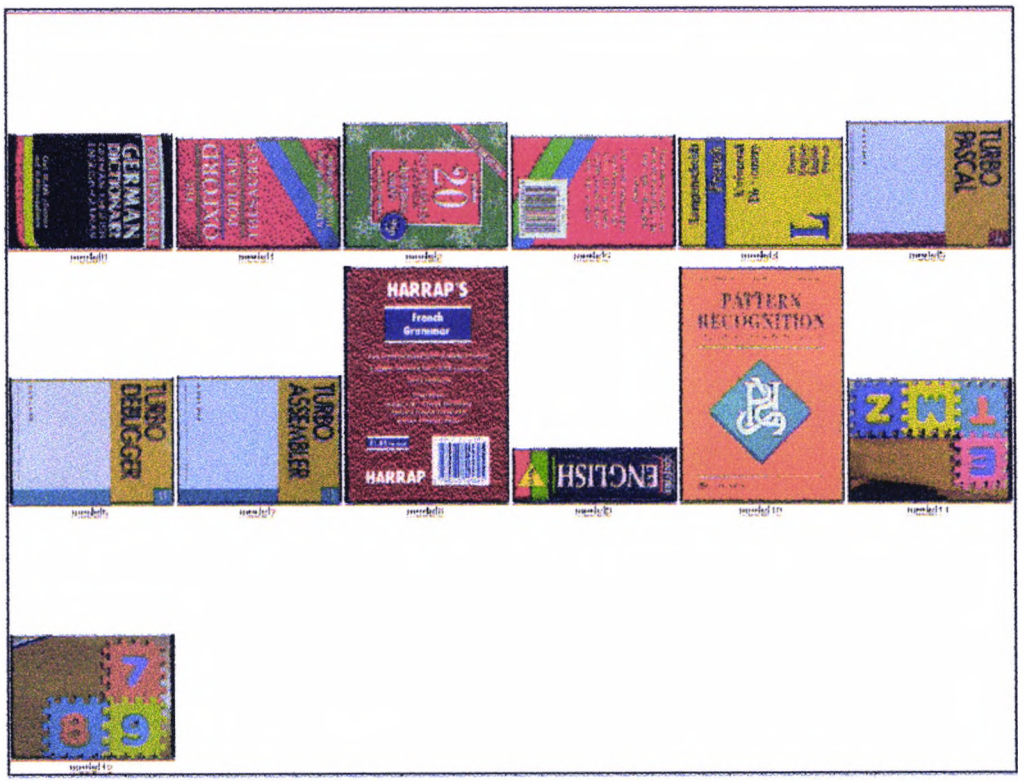


Figure 5.6: The 13 model database used in the object search experiments (the top left-hand model is model0 — the model number increases from left to right, top to bottom — and the last model is model12).

The first search was performed on Figure 5.7 which contains an occluded model11 and a non-occluded model12 in the presence of image clutter. The first step in the search process was to partition Figure 5.7 into regions of constant reflectance (the ***SCF*** with a colour histogram size of 24×24 was used) and to maintain a list of regions with areas greater than 100 pixels. A total of 58 regions were found, 48 of which are displayed in Figure 5.8.

Figure 5.7: An image containing an occluded model11 and non-occluded model12 in the presence of image clutter.

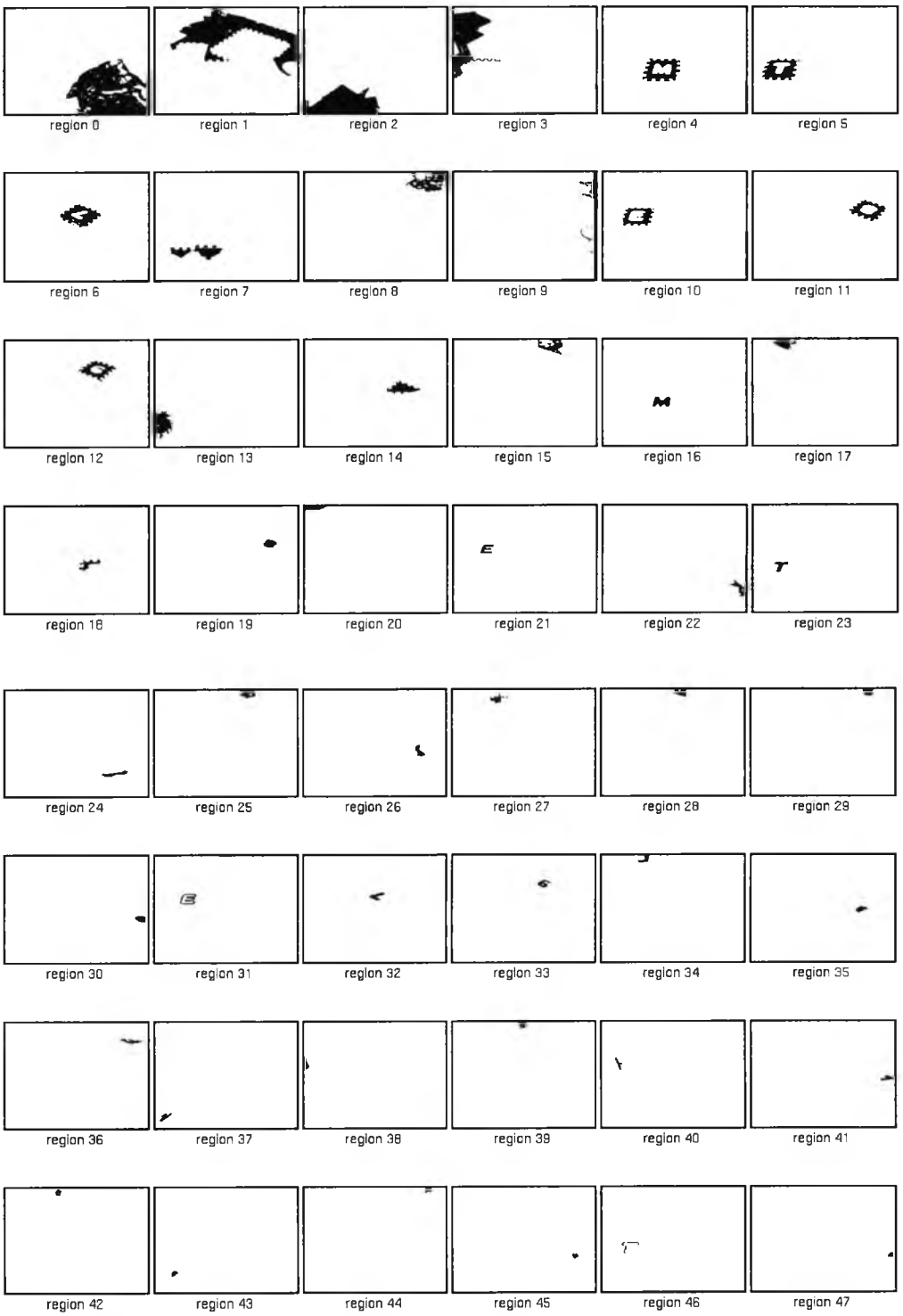


Figure 5.8: The results of partitioning the image in Figure 5.7.

The next step in the algorithm was to generate an ***SNN*** for each model and retrieve the set of valid outputs. For example, since model11 was represented by 8 regions it required an 8-input ***SNN***. The model regions were then assigned to the ***SNN*** inputs.

As described in Section 5.3.1 special care was required when selecting the model regions to assign to each input. To limit the number of pairs passing to the next level of the ***SNN***, model regions which have non-disjoint relationships should be paired at the input (when possible). The region pairs (Table 5.4) selected for model11 illustrate this point — notice that region 1 is assigned to the ***SNN*** input on the far left and region 8 to the input on the far right. Regions 1 and 2 are inputs to the same input ***Lim*** therefore their relationship should be non-disjoint, which it is since it is an enclosure relationship. Only a few region pairs that enter this list will have enclosure relationships so not many pairs will pass to the next level.

Region No./*SNN* input	**Region**
1	Z
2	Z
3	M
4	M
5	T
6	T
7	E
8	E

Table 5.4: The region number assignments for model11. Regions 1, 3, 5 and 7 are the larger regions which enclose 2, 4, 6 and 8, respectively.

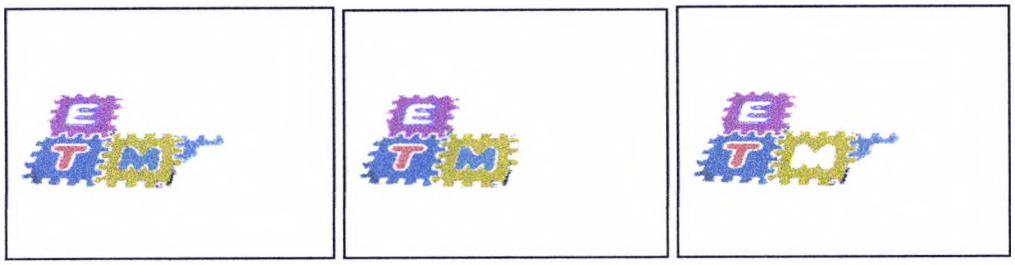
The next part of the algorithm requires that image regions be assigned to the inputs of the ***SNN***. The image regions (and their ranks) assigned to the ***SNN*** inputs for model11 are presented in Table 5.5. The maximum number of ***SNN*** inputs allowed here were 11 (including the blank label which is represented by "Δ").

1	2	3	4	5	6	7	8
21 (0.61)	**51** (0.76)	**51** (0.76)	**16** (0.92)	**33** (0.91)	**23** (0.69)	**23** (0.70)	**32** (0.71)
18 (0.60)	**4** (0.65)	**4** (0.65)	**33** (0.90)	**16** (0.90)	**19** (0.68)	**19** (0.70)	**31** (0.70)
32 (0.60)	**12** (0.65)	**12** (0.64)	**11** (0.89)	**11** (0.90)	**6** (0.68)	**6** (0.70)	**25** (0.64)
5 (0.59)	**41** (0.57)	**41** (0.58)	**5** (0.85)	**5** (0.86)	**50** (0.66)	**34** (0.66)	**8** (0.64)
31 (0.57)	**56** (0.57)	**47** (0.58)	**18** (0.83)	**18** (0.85)	**52** (0.66)	**36** (0.66)	**37** (0.63)
11 (0.57)	**47** (0.57)	**43** (0.57)	**21** (0.81)	**21** (0.83)	**53** (0.66)	**53** (0.66)	**49** (0.62)
33 (0.57)	**43** (0.57)	**20** (0.57)	**32** (0.51)	**32** (0.52)	**36** (0.66)	**52** (0.66)	**22** (0.60)
16 (0.54)	**20** (0.57)	**56** (0.57)	**54** (0.51)	**54** (0.52)	**34** (0.66)	**50** (0.66)	**45** (0.60)
25 (0.54)	**17** (0.57)	**17** (0.57)	**2** (0.51)	**45** (0.52)	Δ	**10** (0.58)	**46** (0.59)
8 (0.54)	**7** (0.56)	**7** (0.57)	**45** (0.51)	**2** (0.52)		Δ	**54** (0.58)
Δ	Δ	Δ	Δ	Δ			Δ

Table 5.5: The symbols (image regions) assigned to the inputs (1-8) of the ***SNN*** for model11. The regions with the best colour rank are at the top of each list. These ranks are displayed in brackets.

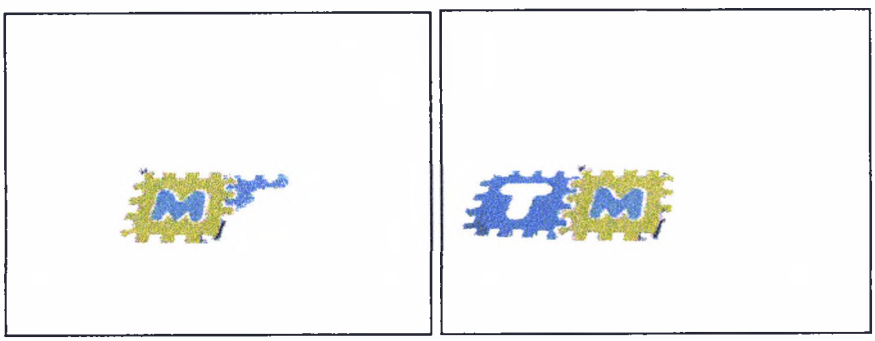
The first three outputs of this ***SNN*** are illustrated in Figure 5.9. Notice that the first output correctly finds the occurrence of model11 in the image. Seven of the eight regions were found — one of the regions was completely occluded. The image regions that matched each of the model regions 1-8 are: 18, occluded, 4, 16, 5, 23, 10 and 31. Notice the position of these regions in the ranked input list of Table 5.5.

Associated with each output is a vote (c.f. Section 5.3.4); the votes for the three outputs are: 135, 110 and 110. This is out of a maximum score of 161 for model11. Model12 and model4 also produced results for this image; model12 was a correct match while model4 was a false positive result. The first three outputs for each of these models are illustrated in Figure 5.10. For model4 all three outputs had votes of 41 (out of a maximum vote of 63), while for model12 the votes were: 110, 86 and 86 (out of a maximum of 110).

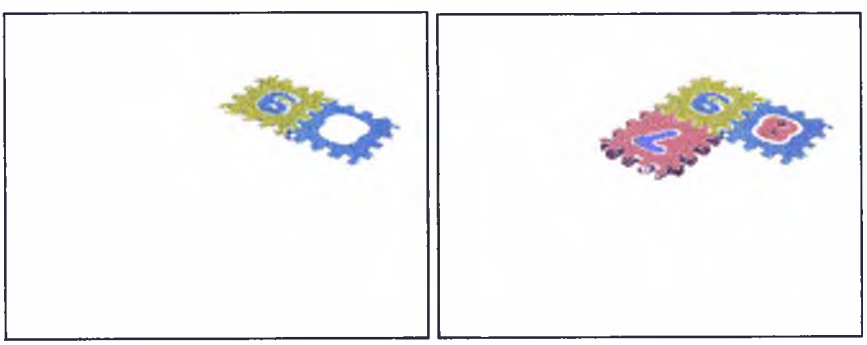


(a) (b) (c)

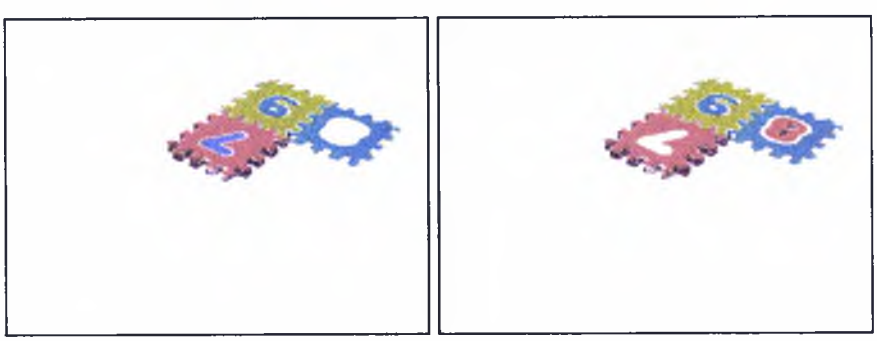
Figure 5.9: The first three *SNN* outputs for model11. (a) Output 1. (b) Output 2. (c) Output 3.



(a) (b)



(c) (d)

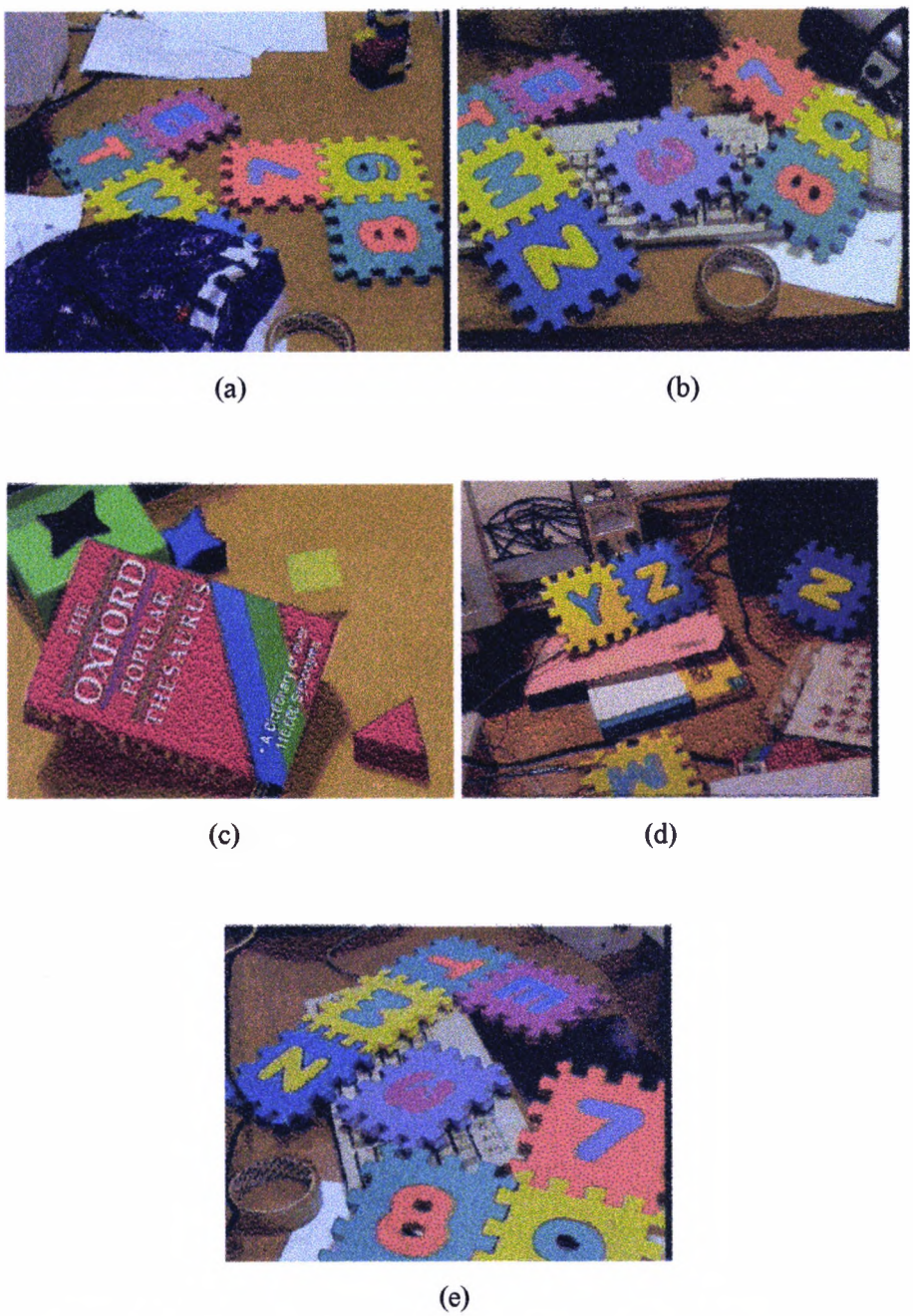


(e) (f)

Figure 5.10: (a)-(c) The first three outputs of the *SNN* for model4 when applied to Figure 5.7. (d)-(f) The first three outputs of the *SNN* for model12 when applied to Figure 5.7.

In this image the search for the 13 models in the database was performed in approximately 8 seconds.

The results for five other images (illustrated in Figure 5.11) are summarised in Table 5.6 and Figure 5.12.



(a) (b)

(c) (d)

(e)

Figure 5.11: The five remaining images used in this experiment.

In Figure 5.12 (a) and (b) models 11 and 12 were found accurately (for the search in Figure 5.11(a)) both with a vote of 110; this is out of a maximum of 161 and 110, respectively. A perfect match was therefore recorded for model 12. The other results are interpreted in a similar way using Table 5.6 and Figure 5.12.[ii]

Image	False Positive	Model	Vote	Max. Vote	Object Found
5.11(a)	y	model4	63	63	
	n	model11	110	161	5.12(a)
	n	model12	110	110	5.12(b)
5.11(b)	n	model11	135	161	5.12(c)
	n	model12	86	110	5.12(d)
5.11(c)	n	model1	63	63	5.12(e)
	y	model3	63	67	
	y	model5	41	41	
	y	model9	60	82	
5.11(d)	y	model1	41	63	
	n	model3	82	67	5.12(f)
	y	model4	41	63	
	y	model6	41	41	
	n	model7	41	41	5.12(g)
	y	model9	41	82	
5.11(e)	n	model11	135	161	5.12(h)
	n	model12	86	110	5.12(I)

Table 5.6: A summary of the results of search for model objects in the five images in Figure 5.11.

The overall recognition rate for the six images in this experiment is 100% with no false negatives and 8 false positives. The false positive results with high votes, for example model4 in Figure 5.11(a) result from image regions with the same colour and region topology as database models. In these cases colour alone is unable to distinguish between these objects, therefore geometric features would have to be used.

[ii] A 100MHz Pentium with 16MB RAM, running the LINUX operating system, was used in these experiments.

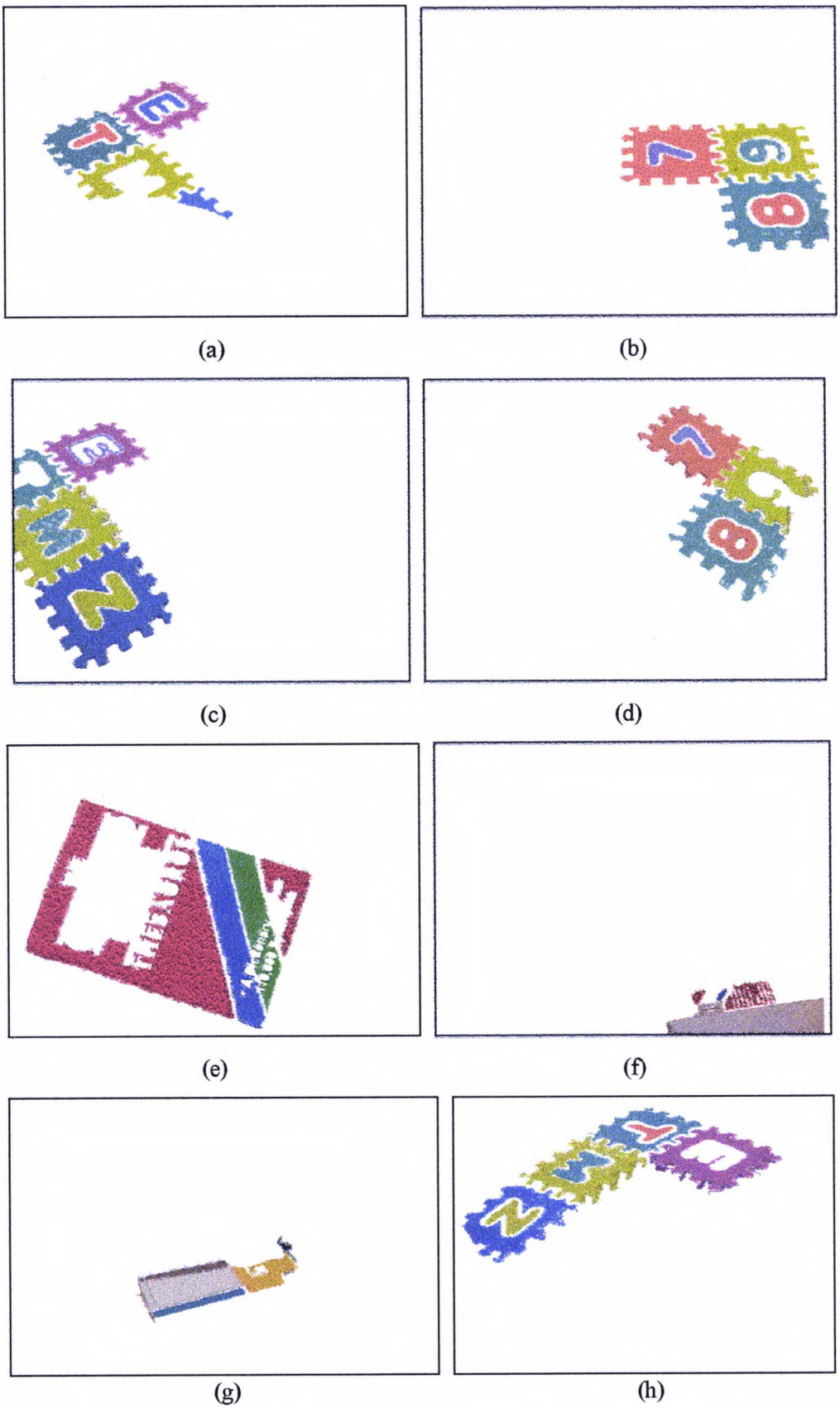


Figure 5.12: The objects that were accurately found in the object search experiment performed on the images in Figure 5.11.

5.5.3 Experiment 2

To determine the effects of varying the number of *SNN* inputs, the minimum image region resolution and the *SNN* input colour threshold, eight different variations of these parameters were used to search the image in Figure 5.11(e). The results of this experiment are presented in Table 5.7.

No.	Min. Area	Colour Threshold	Max. no. of inputs	Matches	*SNN* Output No.	Vote	False Positives
1	400	0.16	30	model11 model12	1 2 1	161 86 63	 model4
2	400	0.16	20	model11 model12	1 2	161 86 63	 model4
3	400	0.16	10	model11 model12	1 2 1	161 86 41	 model4
4	400	0.32	10	model11 model12	1 2 1 1 1	161 86 41 41 41	 model3 model4 model9
5	400	0.48	10	model11 model12	1 2 1 1 1	161 86 41 41 41	 model3 model4 model9
6	100	0.16	10	model11 model12	1 1	135 86	
7	200	0.16	10	model11 model12	1 1	161 86	
8	300	0.16	10	model11 model12	1 2 1	161 86 41	

Table 5.7: The results of varying three parameters in the object search using Figure 5.11(e).

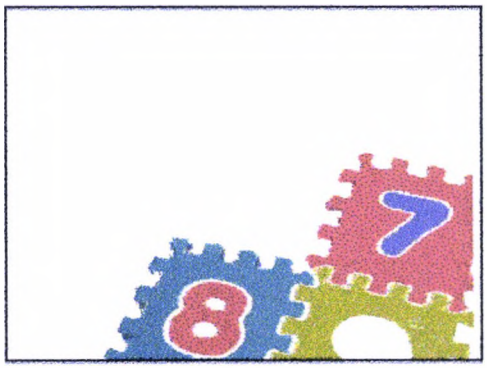
Three observations were made in these experiments. Firstly, the number of matches (and false positives) remained constant when the maximum number of inputs was varied from 10 through 30. This is seen in experiments number 1-3 where model11 and model12 were correct matches and model4 a false

positive. The second observation was that changing the minimum image region area did not affect the results significantly. This is highlighted in experiments 3, 6, 7 and 8 where the minimum area was changed from 400 pixels down to 100 pixels. Interestingly enough, in experiments 7 and 8 there were no false positive results. And finally, the colour threshold parameter was varied from 0.16 to 0.48 in experiments 3, 4 and 5. As expected the results got worse with an increase in false positives. This point which was highlighted in Table 5.3 is a result of introducing region groups with the correct topology but incorrect colour matches.

This experiment demonstrates that the only parameter which significantly affects the outcome of the experiments is the colour threshold. With good colour constancy this threshold can be kept small thus producing less false positive results.

5.5.4 Experiment 3

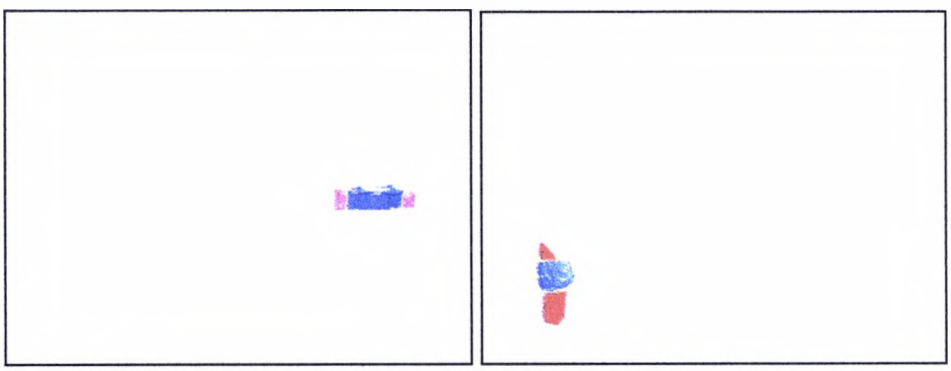
In this experiment a search was performed for two simple 3-dimensional models in the Computer Vision Lab; these objects were a garbage can (represented by two orange regions and one black) and a bag (represented by two pink regions and one blue). The lighting in that part of the Lab is provided by four sets of roof mounted fluorescent lights at different locations — no other/special lighting was used. As illustrated in Figure 5.13 , both objects were at a low spatial resolution and 50% occluded. The maximum number of ***SNN*** inputs allowed was 31 (including the blank symbol), and a ***Lim*** size of 650. After image segmentation only 43 regions had an area greater than 100 pixels. Both objects were described by three regions. The algorithm was able to successfully locate both objects in less than one second, both returning the vote of 41 which was the maximum obtainable for both objects. The results of this experiment are presented in Figure 5.13.



(i)

Figure 5.12 Contd.: The objects that were accurately found in the object search experiments performed on the images in Figure 5.11.

(a)



(b) (c)

Figure 5.13: (a) original image (b) The 50% occluded bag was found accurately (c) The 50% occluded garbage can was found accurately.

5.6 Discussion

In this chapter a model-based object search algorithm was presented which used a modified syntactic neural network to combine image regions and output possible model occurrences. The system was used to represent both 2- and 3-dimensional objects that were perspectively distorted in the scene and at most 50% of their total area was occluded; these objects could be at high or low spatial resolutions. The recognition rate recorded in the experiments was 100% with 8 false positives. These false positives were due to region groups existing in the image with the same region topologies as models in the database. In these circumstances colour alone is unable to differentiate between these objects. The speed of search was less than one second per model.

When compared to existing object search/recognition algorithms this algorithm compares favourably or outperforms them in all areas of comparison. The criteria selected for comparison are: the size of the model representation, the type of objects that can be represented, the amount of allowed occlusion, the recognition rate and the speed of search. The algorithms used for comparison are: colour-histogram based, colour backprojection-based methods, statistical-based methods and the colour adjacency graph (c.f. Chapter 2).

Statistical-based methods are probably the most efficient in terms of the size of model representations since they can describe images in as few as nine floating point numbers. In the modified ***SNN*** method, however the colours of each region (2 floating point numbers) had to be stored for each region, as well as the region relationship table for all the regions. For models with a small number of regions the sizes of both of these representations are similar. For example in Section 5.4 a model was represented by 38 bytes compared with 36 bytes required by the statistical model which uses 9 floating point numbers.

The modified ***SNN*** method can represent both 2- and 3-dimensional models which are perspectively deformed which Matas' colour adjacency graph is also

capable of representing. Colour histogram intersection methods are capable of performing correct recognition with as much as 4/9ths of the original object area occluded. The experiments performed in this Chapter have shown that the modified ***SNN*** is capable of representing objects that are 50% occluded (the allowed amount of occlusion depends on the minimum number of region relationships).

The recognition rate of Matas' colour adjacency graph (an object search algorithm which is capable of searching for similar objects) was 93% with one false negative and two false positives. This method had a recognition rate of 100% with 8 false positives and no false negatives. These algorithms therefore compare favourably in terms of recognition rate. Finally, Swain was able to search for objects in real time; however, his objects were 2-dimensional and not robust to false positives. On the other hand, Matas performed a search for a single object in just over 5 seconds (see Matas [Mat96]). The modified ***SNN*** method was able to find all 13 database object in approximately 8 seconds. This is a remarkable improvement in search speed.

Possible improvements to the modified ***SNN*** method include the inclusion of other region relations such as near (which better defines those regions which do not quite satisfy the adjacency criteria in terms of common border length) and partially enclosed. It is also worth examining how model region relations vary with reduced resolution and modelling these changes. A given pair of regions might then have several relationships instead of one, for example "near" at one resolution and "adjacent" at the next. This information could then be encoded into the model ***RRT*** relationship byte (by setting multiple bits).

Chapter 6

Conclusions

The object search process requires two capabilities: the ability to recognise an object when it comes into view (which is performed by object recognition algorithms); and a mechanism that brings the object into view. If a searcher has only the first capability then all possible regions in the search space would have to be examined at a high spatial resolution. This process, known as linear search, is extremely slow. The need for the second capability should therefore be clear. A feature which seems quite appropriate for this task (and is used by the human visual system) is object colour. Colour is both salient and resistant to changes in spatial resolution — quite unlike geometric features which are more difficult to detect at lower spatial resolutions. Using colour, however does mean that a colour constancy algorithm (which allows the colour of a surface to be seen as the same under different illuminants) must be available if models and images are to be captured under different illuminants. Also, there must be a relatively consistent way of partitioning images into regions of constant reflectance.

The goal of this thesis was to develop a machine vision system which was capable of performing colour object search. It was assumed that a camera system was available allowing scenes to be imaged at both high and low spatial resolutions (through zoom and wide-angled lenses). The requirements of the object search algorithm (described in Chapter 1) were that: the computation complexity of the search would not appreciably increase with an increase in the number of models; a maximum of 50% of the total object area could be occluded; objects may appear with perspective distortion in the scene; only indoor lighting (tungsten and fluorescent) was allowed; and objects could be 2- or 3-dimensional.

This system has been successfully developed. The methodology adopted accepts images of scenes containing objects at high or low spatial resolution. If the object is at a low spatial resolution, the system returns a set of ranked cues (possible object locations) which can be imaged at a higher spatial resolution for more accurate recognition. If the object was at a high spatial resolution then the identity of the object was determined. Colour was used at both high and low spatial resolutions for object localisation and recognition. (It should be clear that geometric techniques could be applied afterwards for further object discrimination.)

The modified syntactic neural network algorithm (c.f. Chapter 5) satisfied all of the requirements of the required search system; while the object search algorithm described in Chapter 3 and the colour and area ratio indexing algorithm (c.f. Chapter 4) only partially fulfilled the requirements. The properties, performance and possible improvements of these methods will be discussed and compared with each other and other colour object search/recognition algorithms in the literature.

In the thesis three algorithms were presented, two performed colour object search and the third colour object recognition. The first object search algorithm determined the salient colours of a model and identified locations in the scene with these colours. Each of these locations served as a cue. A match measure (based on colour proportions) was used to determine if the object was present at a given cue location. The second object search algorithm used a syntactic neural network (***SNN***) to combine image regions with the same colour and topological relationships as the model. A ***SNN*** was generated for each model and a model region assigned to each ***SNN*** input. Image regions with similar colour to the model regions at the inputs were added to the ***SNN*** input list. These regions were then combined by the ***SNN*** and groups of regions output in best-first order. These groups represented possible object occurrences. The object recognition algorithm presented, which was based on

colour and area ratio indexing, determined for each model, image regions with similar area ratios and colour. If three model/image region correspondences were found then a transformation was determined which transformed the model into image space for region matching (based on region colour and position). The quality of match was determined from the number of matching transformed model and image regions.

The prerequisite to each of these algorithm was a colour constancy and image segmentation algorithm. Since model and images were viewed under different indoor illuminants, colour constancy was required. The colour constancy algorithm used (because it was developed in-house) only provided a marginal improvement in colour, however any improvement would increase the robustness of the adopted methods to changes in illumination. If a better colour constancy algorithm were available in-house (such as Finlayson's colour in perspective, c.f. Chapter 2) then potentially further improvement in the results would be achieved. A simple, yet effective colour image segmentation was adopted. This algorithm was based on clustering a colour histogram of the image and backprojecting the pixel in each identified cluster. However, as with most image processing algorithms an improvement in the image segmentation algorithm would have also improved the results.

The first object search algorithm (c.f. Chapter 3) were capable of locating both 2- and 3- dimensional planar objects with, at most, 50% of their surface areas occluded. As presented in the results, the affine distorted objects being identified were illuminated by a combination of tungsten filament and fluorescent lighting in an indoor environment. There was a 100% recognition rate with 51 false positives. Overall, 45% of the objects being searched for were found with rank 1, 45% with rank 2 and 10% with rank 3 (therefore all objects being searched for were found within the three cues with the best ranks).

The modified ***SNN*** algorithm (c.f. Chapter 5) was also capable of representing both 2- and 3-dimensional objects (but they were not restricted to be planar or rigid) with up to 50% of their areas occluded. The lighting used was the same as the first algorithm, tungsten and fluorescent. There was a recognition accuracy of 100% with only five false positive results.

The object recognition algorithm was able to recognise both 2- and 3-dimensional planar (and rigid) objects containing at least four regions and affine distorted. The lighting conditions were the same as the other algorithms described in this thesis. There was one false negative result and 5 false positives. An important feature of the algorithm was its ability to represent objects with disjoint regions. The order of complexity of this algorithm is lower than the exhaustive search method, the ***CLM*** model (because it has a shape computation overhead) and geometric hashing (which uses more point triplets). In terms of the size of the model representation it is greater than statistical and colour histogram-based methods since for each model region five parameters must be stored, as well as the area ratio table. The two main advantages of the algorithm are that it can represent 3-dimensional planar objects (statistical and colour histogram-based methods can not), and it can recognise objects in complex, cluttered environments (c.f. Chapter 4); the other two methods are simply not able to do that. Finally, this recognition algorithm has one advantage over Matas' colour adjacency graph method, because it is able to represent objects which are described by a set of disjoint regions (e.g. a textured object). However, Matas' algorithm is more general since it can describe 3-dimensional objects which are perspectively distorted.

When comparing the two object search algorithms presented in this thesis, it is important to realise that the modified ***SNN*** algorithm operates under a wider range of conditions; notably perspective distortion and non-planar/non-rigid object representations. Also since the modified ***SNN*** method uses region topology in its descriptions it can more accurately represent objects with similar colour proportions. Three criteria will be used to compare these

algorithms: available computer memory, the size of the model database, and the reliability of the image segmentation algorithm. Firstly, if the size of the computer's memory that the algorithm is running on (say a mobile robot) is limited then the first algorithm (not the ***SNN*** based algorithm) would be more appropriate since it uses significantly less memory (c.f. Chapter 3 and 5). Secondly, the computation complexity of the ***SNN*** based algorithm only increases by a small amount with the addition of new models to the database; this is not the case with the other algorithm. Finally, the ***SNN*** method relies heavily on the accuracy of image segmentation to define topological relationships; however the other search algorithm only requires one object region to be identified accurately in order to locate it.

The performance of the colour-based object search algorithm described in Chapter 3 is better than Swain's histogram backprojection technique because it not only identifies object cues but ranks them as well. Histogram backprojection methods in general are restricted to representing 2-dimensional objects; while this algorithm can represent 2- or 3-dimensional planar objects. Also, this algorithm performs well in complex, cluttered environments and requires no a priori knowledge of the size of the object. All the colour histogram-based methods require the colour histogram of the object to be stored, which might require 64, or 256 floating point numbers for 8x8 and 16x16 colour histograms, respectively. The adopted method, however only requires that 5 floating point numbers be stored for each model colour. This algorithm, however does not perform as well as Matas' [Mat96] colour adjacency graph method in many areas. To compare these two methods three criteria are used: object representation, search speed and accuracy. Matas' method can represent 3-dimensional, non-rigid objects which are perspectively distorted; this algorithm can only represent 3-dimensional planar objects which are affine distorted. Matas recorded a 93% (2 false negatives) accuracy while this method 100% (no false negatives), however Matas only had one false positive result while this method had 51. This high false positive rate was expected because several of the database models used in these experiments had

similar colours; this was not the case with Matas' database. Finally, in terms of search speed this algorithm performs better because its order of complexity is linear (Matas' is approximately $O(N^3)$). One finally point should be made, this algorithm would perform better than Matas' if the object has a low spatial resolution and could not be accurately segmented from the background — thus returning invalid region adjacency relationships.

When compared to existing object search/recognition algorithms the modified ***SNN*** algorithm compares favourably or outperforms them in all areas of comparison. The criteria selected for comparison are: the size of the model representation, the type of objects that can be represented, the amount of allowed occlusion, the recognition rate and the speed of search. Statistical-based methods represent images in as little as nine floating point numbers, however the modified ***SNN*** method, must store the colours of each region (2 floating point numbers) and the region relationship table. For models with a small number of regions the sizes of both of these representations are similar (c.f. Chapter 5). Both the modified ***SNN*** method and Matas' colour adjacency graph can represent 2- or 3-dimensional objects which are perspectively deformed. Colour histogram intersection methods are capable of performing correct recognition when as much as 4/9ths of the original object area is occluded. The experiments performed in Chapter 5 show that the modified ***SNN*** is finding objects that are 50% occluded, however the allowed amount of occlusion depends on the minimum number of region relationships that are required. The recognition rate of Matas' colour adjacency graph was 93% with two false negatives and one false positives. This method had a recognition rate of 100% with 8 false positives. Finally, Swain was able to search for objects in real time, however these objects were 2-dimensional and not robust to false positives. On the other hand, Matas [Mat96] performed a search for a single object in just over 5 seconds. The modified ***SNN*** method was able to search through the entire database of 13 objects in appropriately 8 seconds.

There are several improvements that could be made to the three algorithms presented in this thesis. In terms of the colour and area indexing recognition algorithm different invariants could be used to extend the type of objects that can be represented. One example is to use the cross ratio invariant [MZ92] which would allow the algorithm to be invariant to perspective distortions. For the object search algorithm presented in Chapter 3 the object size determination method could be improved by assuming that one object region is not occluded in the scene; that way the quality of match could be determined more accurately, thus improving recognition. Finally, additional relationships could be included in the modified ***SNN*** method. These include "near" and "partially enclosed." With the help of these (fuzzy-type) relationships the effects of reducing spatial resolutions or occlusion can be more accurately modelled. For example, a region may normally enclose another region but is occluded in the scene; by allowing fuzzy relationships, enclosure can deteriorate to partial enclosure, and partial enclosure to adjacency. Also, at one spatial resolution two objects might be adjacent, but at another near. These modifications would improve the overall robustness of the algorithm to object occlusion and changes in spatial resolution.

In conclusion, colour represents a valuable but under-used property in Computer Vision, especially in the areas of object search and recognition. This thesis has demonstrated that colour can be used successfully both as a cueing mechanism and to discriminate between objects of dissimilar colour. Colour is especially useful when the object has a low spatial resolution because it is region-based, thus resistant to changes in spatial resolution (unlike geometric features which are less well defined at lower spatial resolutions). In images with a large field of view colour can be used to generate a set of cue locations which can be examined at a high spatial resolution. This improves the speed of search because the entire search space does not need to be examined at this high spatial resolution.

Glossary

CCA	Colour constancy algorithm
CFG	Context-free grammar
CLM	The colour landmark model which utilises object region colour, shape and region topology to describe an object [Wal96].
CRAG	The colour region adjacency graph.
FSD	Feature and Spatial Domain clustering: an image segmentation algorithm [Mat96].
HVC	Hue/Value/Chroma colour space. ***Hue***: describes the type of colour; ***Value***: describes the total amount of light; ***Chroma***: describes the purity of the colour, i.e. the amount of white light mixed with the colour.
RRT	The region relationship table (stores the relationships between regions)..
SCF	The Software colour filter which backprojects salient colours onto the image and identifies spatially close colour groups. This technique is used in an object search algorithm but is also a partial image segmentation algorithm [WE96].
SNN	The Syntactic Neural Network [Luc96] is a neural network which combines symbols using a set of rules and provides fast retrieval of these symbols.
SPD	The spectral power distribution of a given light source.

References

[ABS90] Andreadis, I., Browne M., A., Swift, J., A., "Image Pixel Classification by Chromaticity Analysis", PRL, **11**, 1990, pp 51-58.

[ACD96] Ardizzone, E., La Cascia, M., Molinelli, D., "Motion and Color-Based Video Indexing and Retrieval", Proc. of ICPR'96, 1996, pp 135-139.

[Bal81] Ballard, D., "Generalising the Hough Transform to Detect Arbitrary Shapes", PR, **13**, 2, 1981, pp 111-122.

[BB82] Ballard, D., Brown, C., Computer Vision, Prentice Hall, 1982.

[BE92] Brock-Gunn, S., Ellis, T., "Using Colour Templates for Target Identification and Tracking", Proc. BMVC92, Ed. Hogg and Boyle, Leeds, UK, Sept., 1992, pp 207-216.

[Ber87] Berry, D., T., "Colour Recognition Using Spectral Signatures", Pattern Recognition Letters, **6**, 1, June, 1987, pp 69-75.

[Bri90] Brill, M., H., "Image Segmentation by Object Color: a Unifying Framework and Connection to Color Constancy", J. Opt. Soc. Am. A, **7**, 10, October, 1990, pp 2041-2047.

[BW86] Brainard, D., H., Wandell, B., A., "Analysis of the Retinex Theory of Color Vision", Journal of Opt. Soc. Am. A, Vol. 3, No. 10, October, 1986, pp 1651-1661.

[Cel90] Celenk, M., "A Color Clustering Technique for Image Segmentation", Computer Vision, Graphics, and Image Processing, **52**, 1990, pp 145-170.

[CG84] Chassery, J., M., Garbay, C., "An Iterative Segmentation Method Based on a Contextual Color and Shape Criterion", IEEE Transactions on Pattern Analysis and Machine Intelligence, **PAMI-6**, 6, November, 1984, pp 794-800.

[CR93] Caelli, T., Reye, D., "On the Classification of Image Regions by Colour, Texture and Shape", PR, **26**, 4, 1993, pp 461-470.

[CV95] Cyganski, D., Vaz, R., "A Linear Signal Decomposition Approach to Affine Invariant Contour Identification", Pattern Recognition, **28**, 12, 1995, pp 1845-1853.

[CW96] Chang, C., Wang, L., "Color Texture Segmentation for Clothing in a Computer-Aided Fashion Design System", Image and Vision Computing, **14**, 1996, 685-702.

[ED94] Ellis, T., J., Dommes, T., "Object Identification Employing Surface Markings", SPIE94, Boston, USA, November, 1994.

[EM95] Ennesser, F., Medioni, G., "Finding Waldo, or Focus of Attention Using Local Color Information", IEEE Transactions on Pattern Analysis and Machine Intelligence, **17**, 8, August, 1995, pp 805-809.

[FCF96] Finlayson, G., D., Chatterjee, S., S., Funt, B., V., "Color Angular Indexing", in Proc. of the 4th European Conf. On Comp. Vis., Cambridge, UK, April, 1996, pp16-27.

[FDF93] Finlayson, G., D., Drew, M., S., Funt, B., V., "Diagonal Transforms Suffice For Color Constancy", Proc. of ICCV, 1993, pp 164-171.

[FDH91] Funt, B., V., Drew, S., Ho, J., "Colour constancy from mutual reflection", Intl. J. Computer Vision, **6**, 1, April, 1991, pp 5-24.

[FF95] Funt, B., V., Finlayson, G., D., "Color Constant Color Indexing", IEEE Trans. on Pat. Anal. And Mach. Intelli., **17**, 5, May, 1995, pp 522-529.

[FH97] Finlayson, G., D., Hordley, S., "Selection and Gamut Mapping Colour Constancy", Proc. Of the 8th BMVC Conf., University of Essex, UK, September, 1997.

[Fin92] Finlayson, G., D., "Colour Object Recognition", Master's thesis, Simon Fraser University, 1992.

[Fin95] Finlayson, G., D., "Coefficient Color Constancy", Ph.D. thesis, Simon Fraser University, April, 1995.

[For88] Forsyth, D., A., "A Novel Approach to Colour Constancy", 2nd Int. Conf. Comput. Vision, Tarpon Springs, FL, USA, December, 1988, pp 9-18.

[FS93] Flusser, J., Suk, T., "Pattern Recognition By Affine Moment Invariants", Pattern Recognition, **26**, 1, 1993, pp 167-174.

[FV92] Ferri, F., Vidal, E., "Colour Image Segmentation and Labelling through Multiedit-Condensing", PRL, **13**, 1992, pp 561-568.

[GG92] Gavrila, D. M., Groen, F., C., A., "3D Object Recognition from 2D Images Using Geometric Hashing", PRL, **13**, 1992, pp 263-278.

[Gri86] Grimson, W., E., L., "The Combinatorics of Local Constraints in Model-based Recognition and Localization from Sparse Data", Journal of the Association for Computing Machinery, **33**, 4, October, 1986, pp 658-686.

[Gri90] Grimson, W., E., L., "Object Recognition by Computer: The Role of Geometric Constraints", MIT Press, Cambridge, Massachusetts, London, England, 1990.

[GS95] Gong, Y., Sakauchi, M., "Detection of Regions Matching Specified Chromatic Features", Computer Vision and Image Understanding", **61**, 2, March, 1995, pp 263-269.

[GS96] Gevers, T., Smeulders, A., W., M., "Color-Metric Pattern-Card Matching for Viewpoint Invariant Image Retrieval", Proc. of ICPR'96, 1996, pp 3-7.

[Hac96] Hachimura, K., "Retrieval of Paintings Using Principal Color Information", Proc. of ICPR'96, 1996, pp 130-134.

[HB87] Healey, G., E., Binford, T., O., "The Role and Use of Colour in a General Vision System", proc. DARPA IV Workshop, USC, CA, USA, 1987, pp 599-613.

[HB88] Healey, G., E., Binford, T., O., "A Color Metric for Computer Vision", Proc. IEEE Conf. Comput. Vision & Pattern Recogn., Ann Arbor, MI, USA, 1988.

[HDMN96] Huang, Q., Dom, B., Megiddo, N., Niblack, W., "Segmenting and Representing Background in Color Images", Proc. of ICPR'96, 1996, pp 13-17.

[HE95] Hung, T, W, R, Ellis, T, "Spectral Adaptation with Uncertainty using Matching", IEE Proc. of the Fifth Int. Conf. on Image Processing and its applications, Scotland, 1995, pp 786-790.

[Hea89] Healey, G., E., "Using Color for Geometry-Insensitive Segmentation", J. Opt. Soc. Am. A, **6**, June, 1989, pp 920-937.

[Hea89b] Healey, G., E., "Color Discrimination by Computer", Man. and Cybernetics, **19**, 6, November/December, 1989, pp 1613-1617.

[Hea92] Healey, G., E., "Segmenting Images Using Normalised Color", IEEE Transaction on Systems Man. And Cybernetics, **22**, 1, January/February, 1992, pp 64-73.

[Hil86] Hilbert, D., R., "Color and Color Perception: A Study in Anthropocentric Realism", Center for the Study of Language and Information (CSLI), 1986.

[Hou62] Hough, P., V., C., "Methods and Means for Recognizing Complex Patterns", U.S. Patent 3069654, 1962.

[HS94] Healey, G., Slater, D., "Using Illumination Invariant Color Histogram Descriptors for Recognition", CVPR, 1994, pp 355-360.

[KB90] Khotanzad, A., Bouarfa, A., "Image Segmentation By a Parallel, Non-Parametric Histogram Based Clustering Algorithm", Pattern Recognition, **23**, 9, 1990, pp. 961-973.

[KCUU86] Keller, J., M., Covavisaruch, N., Unklesbay, K., Unklesbay, N., "Color Image Analysis of Food", IEE Computer Society Conf. On Comp. Vis. and Pattern Recognition Proc., Miami Beach Florida, June, 1986, pp 619-621.

[Ken76] Kender, J., "Saturation, hue, and normalized color: Calculation, digitization effects, and use", CMU Comput. Sci. Dept., Nov., 1976.

[KMW96] Kankanhalli, M., S., Mehtre, B., M., Wu, J., K., "Cluster-Based Color Matching for Image Retrieval", Pattern Recognition, **29**, 4, 1996, pp 701-708.

[KNF76] Koontz, W., Narendra, P., M., Fukunaga, K., " A graph theoretic approach to non-parametric cluster analysis", IEEE Trans. Comput. **C-25**, 1976, pp 936-944.

[Kri78] Kries, J., V., "Beitrag Zur Physiologie der Gesichtsempfindung", Archives of Anatomy and Physiology, 2, 1878, pp 505-524.

[Lan77] Land, E., "The Retinex Theory of Color Vision", Sci. Am., **237**, 6, 1977, pp 108-128.

[Lan86] Land, E. H., "Recent Advances in Retinex Theory", Vision Res., **26**, 1, 1986, pp 7-21.

[LM71] Land, E., H., McCann, J., J., "Lightness and Retinex Theory", J. Opt. Soc. Am., **61**, 1971, pp 1-11.

[Luc96] Lucas, S., M., "Rapid best-first retrieval from massive dictionaries", PRL, **17**, 1996, pp 1507-1512.

[LW88] Lamdan, Y., Wolfson, H., J., "Geometric Hashing: a general and efficient model-based recognition scheme", proc. Second Int. Conf. On Computer Vision, Tampa, FL, 1988, pp 238-249.

[Mat96] Matas, J., "Colour-based Object Recognition", Ph.D. thesis, University of Surrey, May, 1996.

[MK95] Matas, J., Kittler, J., "Spatial and Feature Space Clustering: Applications in Image Analysis", Proc. of the 6th Int. Conf. On Computer Analysis and Patterns, Praque, Czech Republic, September, 1995.

[MKNM95] Mehtre, B., M., Kankanhalli, M., S., Narasimhalu, A., D., Man, G., C., "Color Matching for Image Retrieval", Pattern Recognition Letters, **16**, March, 1995, pp 325-331.

[MMK93] Matas, J, Marik, R, Kittler, J, "Generation, Verification and Localisation of Object hypotheses based on colour", in J. Illingworth, editor, BMVC, BMVA Press, 1993, pp 539-548.

[MMK94] Matas, J, Marik, R, Kittler, J, "Illumination Invariant Colour Recognition", in E. Hancock, editor, BMVC, BMVA Press, 1994, pp 469-478.

[MMK95] Matas, J, Marik, R, Kittler, J, "On Representation and Matching of Multi-coloured Objects", Proc. of ICCV, Boston, 1995, pp 726-732.

[MW86] Maloney, L., T., Wandell, B., A., "Color Constancy: A Method for Recovering Surface Spectral Reflectance", J. Opt. Soc. Am. A, **3**, 1, January 1986, pp 29-33.

[MZ92] Mundy, J., L., Zisserman, A., "Geometric Invariance in Computer Vision", MIT Press, Cambridge, Ma, London, 1992.

[NB93] Niblack, W., Barber, R., et al., "The QIBC project: Querying Images by Content Using Color, Texture and Shape", in Storage and Retrieval for Image and Video Databases *I*, volume 1908 of SPIE Proceedings Series, Feb. 1993.

[ODE96] Olatunbosun, S, Dowling G, Ellis, T, "Topological Representation For Matching Coloured Surfaces", ICIP 96, Switzerland, September, 1996.

[OKS80] Ohta, Y., Kanade, T., Sakai, T., "Color information for Region Segmentation", Computer Graphics and Image Processing **13**, 1980, pp 222-241.

[PTVF88] Press, W., H., Teukolsky, S., A., Vetterling, W., T., Flannery, B., P., "Numeric Recipes in C: The Art of Scientific Computing", Cambridge University Press, 1988.

[SA81] Sarabi, A., Aggarwal, J., K., "Segmentation of Chromatic Images", PR, **13**, 6, 1981, pp 417-427.

[SB90] Swain, M., J., Ballard, D., H., "Indexing via Color Histograms", Proc. of ICCV, 1990, pp 390-393.

[SC96] Syeda-Mahmood, T., F., Cheng, Y., "Indexing Colored Surfaces in Images", Proc. of ICPR'96, 1996, pp 8-12.

[Sch94] Schettini, R., "Multicolored Object Recognition and Location", Pattern Recognition Letters, **15**, 1994, pp 1089-1097.

[SHB93] Sonka, M., Hlavac, V., Boyle, R., "Image Processing, Analysis and Machine Vision", Chapman & Hall, 1993.

[SO95] Stricker, M., Orengo, M., "Similarity of Color Images", In Storage and Retrieval for Image and Video Databases *III*, volume 2402 of SPIE proceedings Series, Feb. 1995, pp 381-392.

[SS94] Stricker, M., Swain, M., The Capacity Of Color Histogram Indexing, Proc. of CVPR94, 1994, pp 704-708.

[Str93] Strachan, N., J., C., "Recognition of Fish Species By Colour and Shape", Image and Vision Computing, 1993, pp 2-10.

[Swa90] Swain, M, "Colour Indexing", Ph.D. thesis, University of Rochester, 1990.

[Sye92] Syeda-Mahmood, T., "Data and Model-Driven Selection Using Color Regions", Image and Understanding Workshop 92, Morgan Kaufmann, 1992, pp 705-716.

[TC92] Tsai, D., Chen, Y., "A fast histogram-clustering approach for multi-level thresholding", PRL, 13, 1992, pp 245-252.

[Tho92] Thomas, A., D., H., "Compressing the Parameter Space of the generalised Hough Transform", PRL, 13, 1992, pp 107-112.

[Tom90] Tominaga, S., "Color Classification of Color Images Based on Uniform Color Spaces", Human Vision and Electronic Imaging: Models, Methods, and Applications, **SPIE 1249**, 1990, pp 356-365.

[Ull86] Ullman, S., "An Approach to Object Recognition: Aligning Pictorial Descriptions", TR931, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1986.

[VM96a] Vinod, V., V., Murase, H., "Object Location Using Complementary Color Features: Histogram and DCT", in Proc. of ICPR 96, 1996, pp 554-559.

[VM96b] Vinod, V., V., Murase, H., "Focussed Color Intersection for Object Extraction from Cluttered Scenes", in Proc. of Vision Interface, May, 1996.

[Wal96] Walcott, P, "Object Recognition Using Colour, Shape and Affine Invariant Ratios", proc. of the BMVC Conf., Edinburgh, Scotland, September, 1996, pp 273-282.

[WB89] Wixson, L, Ballard, D, "Real-time Detection of Multi-coloured Objects", SPIE Sensor Fusion II: Human and Machine Strategies, 1198, November, 1989, pp 435-446.

[WE95] Walcott, P, Ellis, T, "The Localisation of Objects in Real World Scenes Using Colour", proc. of the 2nd ACCV Conf., Singapore, December, 1995, pp 243-247.

[WE96] Walcott, P, Ellis, T, "Modelling Colour Surfaces Using Colour Landmarks", proc. of the ninth IMDSP Conf., Belize, March 1996, pp 100-101.

[WE98a] Walcott, P, Ellis, T, "A Colour Object Search Algorithm", to appear in the 9th BMVC Conf., September, 1998.

[WE98b] Walcott, P, Ellis, T, "Object Recognition Using Colour and Area Ratio Indexing", Technical Report, TR-Wal-02, City University, May, 1998.

[Wix94] Wixson, L., "Gaze Selection for Visual Search", Ph.D. thesis, University of Rochester, 1994.

[WN90] Weill, R., Yair, N., "Use of Colour Vision for Citrus Plucking", Robotic Systems and AMT, Ed. G. Halevi, Yudilevich, I., Weill, I., Elsevier Science Publishers B.V., North Holland IFIP, 1990, pp 161-171.

Appendices

A.1 A Statistical Shape Descriptor

Recently, several affine invariant shape descriptors have been described in the literature [FS93][CV95] most of which could have been used here, however in the ***CLM*** [WE96][Wal96] a statistical (moments based) shape descriptor was used. This shape descriptor was calculated for filled object regions rather than for object region boundaries. It is expected that a region-based shape descriptor would be more expensive computationally than one based on the boundary alone. However, a region-based descriptor should be more robust than a boundary descriptor to occlusion and boundary distortions.

Given the discrete form of the ***(p+q)*** order moment of a *binary* image function ***f(x,y)***, the general moment m_{pq} and the central moment μ_{pq} can be defined as:

$$m_{pq} = \sum_x \sum_y x^p y^q f(x,y) \tag{A1.1}$$

$$\mu_{pq} = \sum_x \sum_y (x - x_c)^p (y - y_c)^q f(x,y) \tag{A1.2}$$

where $x_c = \frac{m_{10}}{m_{00}}$ and $y_c = \frac{m_{01}}{m_{00}}$.

Flusser et al. [FS93] defined a second order affine invariant moment I_1:

$$I_1 = \frac{(\mu_{20}\mu_{02} - \mu_{11}^2)}{\mu_{00}^4} \tag{A1.3}$$

The computational expensive of I_1 can be reduced through the use of the following formulae:

$$\mu_{00} = m_{00} \tag{A1.4}$$

$$\mu_{02} = m_{02} - y_c m_{01} \tag{A1.5}$$

$$\mu_{11} = m_{11} - y_c m_{10} \tag{A1.6}$$

$$\mu_{20} = m_{20} - x_c m_{10} \tag{A1.7}$$

The moment I_1 is less sensitive to digitalisation errors, minor shape deformations, camera non-linearity and non-ideal camera positions and less expensive computationally than higher order moments.

A.2 Performance of Hung's Colour Constancy Algorithm

To recognise the colour of a surface as the same when viewed under different illuminants requires a colour constancy algorithm. Most of these algorithms make strong assumptions about the nature of the world, for example many assume a Mondrian world with constant illumination, no inter-reflection and a single light source. For these reasons it is important to determine whether a given colour constancy algorithm will help to correct image colours before using it. Hung's colour constancy algorithm [HE95] makes the above assumptions and therefore theoretically is constrained to a Mondrian world; however what is of interest here is its performance under indoor lighting (fluorescent and tungsten lighting).

An experiment was formulated to determine how well Hung's algorithm brings into correspondence the colours of two images of the same scene — one viewed under tungsten lighting and the other under fluorescent — compared with not using colour constancy at all. To determine this the colour difference (the Euclidean distance between the chromaticity co-ordinates of the two regions) was calculated using the colours of the same region viewed under tungsten lighting and fluorescent lighting. The colour difference was calculated before and after colour constancy and the values compared. This was repeated for all image regions and a mean colour difference calculated. If the colour constancy algorithm worked then the mean colour difference after constancy should be better than before constancy. This experiment is detailed in Experiment A2.1.

Experiment A2.1: **Colour Constancy Experiment**

1. Capture two images of the same scene, one under tungsten and the other under fluorescent lighting. Label these images I_T and I_F, respectively.

2. Apply Hung's colour constancy algorithm to both images and label the colour constant images I_{TC} (colour constancy applied to I_T) and I_{FC}.
3. Partition and of the images into regions of constant reflectance and determine for each image the mean chromaticity co-ordinates (r,g) of each image region.
4. For the same region in each image compute the colour distance E_i (the Euclidean distance between the mean chromaticity co-ordinates) for image pairs I_{TC} and I_{FC} and I_T and I_F. Repeat this for all regions.
5. Sum the colour differences for all the regions for each image pair.

In Figure A2.3 a graph of the region colour differences versus image regions, with and without colour constancy, for the scene in Figure A2.1 is illustrated — Figure A2.1 was partitioned into 55 regions (c.f. Figure A2.2) which were used in the calculation of the colour differences. The 55 regions are labelled 0-54 on the horizontal axis; the areas of these regions are in ascending order, therefore region 0 is larger than region 1 and so on.

The mean colour difference calculated in the experiment was 0.069 with colour constancy applied to the images and 0.075 without. This indicates a marginal improvement of the colour when colour constancy is used. If Figure A2.3 is examined more closely then it is realised that some of the smaller regions have large colour errors; this is expected because as described in Chapter 2 the prominent colours are used to calculate the transformation.

Although the improvement in colour in this experiment is only marginal it is an improvement, therefore Hung's algorithm is used throughout this work. Also, it is expected that this algorithm will perform better in more diverse lighting conditions.

Figure A2.1: The scene used in the colour constancy experiment which contains a multicoloured toy, parts of a floor mat and a computer keyboard.

Figure A2.2: The results of segmenting Figure A2.1 — 55 regions result.

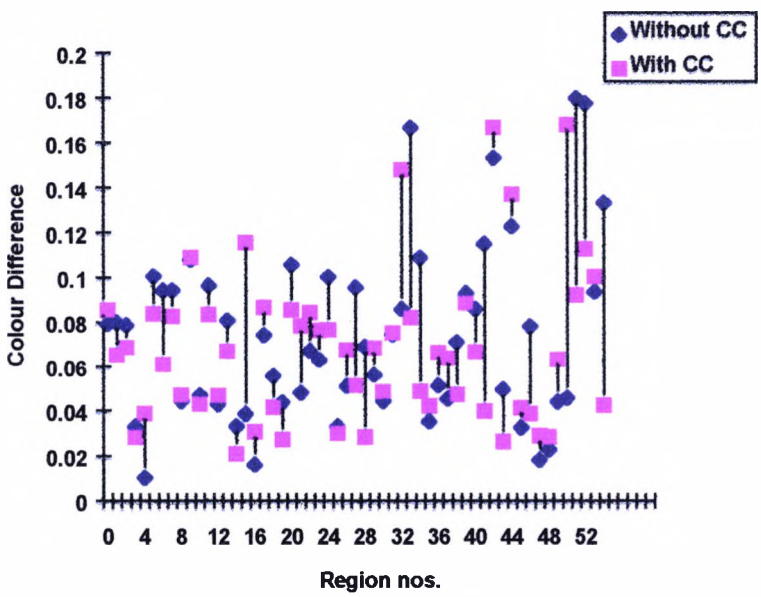


Figure A2.3: The region colour differences with and without colour constancy.