Preamble

Inspecting coloured industrial products is an important application area for vision systems. In addition, natural products and raw materials are often inspected, graded or sorted by machine on the basis of colour. For Machine Vision, the prime requirement is almost always *verification*, rather than unassisted *recognition*. The range of variation of each colour associated with an industrial artifact can sometimes be specified quite precisely. This not always possible; in many situations, colours have to be learned from carefully selected production samples. The range of variation of the colours of natural products almost always has to be learned. Even "standard" colours, such as *red*, *yellow*, *green*, etc, also have to be learned. The reason is that every person identifies colours in a different way. Once it has been trained, a colour-recognition program generates images in which each "intensity" value is effectively the name of a colour. Such an image can then be measured using the techniques described earlier. On the way through this chapter, we will also see how colour images can be enhanced and their edges detected.

Colour Vision

Over the centuries, colour vision has been studied extensively by philosophers, psychologists, physiologists, biologists, physicists and chemists. Photographers, printers and video engineers also have a special interest in colour, with a emphasis on trying to reproduce the colours encountered in nature. Research has resulted in a large but still incomplete body of knowledge, termed *Colour Science*. Understandably, this is concerned primarily with understanding and emulating colour vision in humans and animals. *That is not our interest here!*

The preoccupation of industrial Machine Vision systems designers is verifying that coloured features on artifacts are of the correct hue, size, position and shape. We need to be able to segment an image according to the colours that it contains. For example, we might want to distinguish *"red"* and *"blue"* printing; identify *"yellow"* tablets against a '*'green"* surface (to count them); or measure the area covered by red raspberry jam on a cake. Performing tasks such as these does not require that a machine can replicate the subtleties of human colour vision.

Instead, vision engineers are often faced with stark questions, such as

"Is the feature at position [X,Y] green?"

"How big is the "green" blob that covers point [X,Y]?"

But, what is *"green"?* My late wife and I often disagreed (in the nicest possible way, of course) about the names of colours, particularly in the blue-green part of the spectrum. Who is correct? My concept of *"green"* was quite different from hers and, almost certainly, yours as well! I encountered this recently, when a friend saw me taking vitamin D capsules. On this occasion my wife and I agreed that they were *“obviously blue”* but our friend was equally insistent that they were *"green"*. Who was correct? In fact, we all were, according to our own internal concepts of *"green"*. There is no formal physical, or mathematical, definition of colour classes, such as *“blue”*, *“green”*, *“yellow”* etc.. Colour classification is learned in early childhood and is revised through everyday experiences.

In this chapter, we are seemingly going to ignore much of Colour Science. We will concentrate on how *Colour Recognition* can be learned by a machine, or program, under the guidance of a human being. In fact, we are not discarding all of the scientific work that has been devoted to understanding colour but to build on it. The fruits of that research are embodied in the design of the camera. As an engineer, I use what resources are available, in this case low-cost high-quality RGB cameras. They are sufficient to be part of a chain that can reproduce a wide range of natural colours. That is proof enough that the RGB signals convey enough information to be useful within a colour-recognition system.

Typical applications that we will address include:

- Inspecting printing on documents, bank notes, packages, credit cards, etc.
- Examining patterned fabric, carpets,
- Guiding a robot to pick ripe fruit from a tree
- Trimming and grading fruit and vegetables
- Controlling food production (e.g. counting olives and tomato slices on a pizza)
- Monitoring the distribution of cherries in cake slices

- Reading resistor colour codes
- Identifying coloured wires
- Controlling manufacturing systems (assembly, machining, painting, etc)

Other applications of the *coarse* colour recognition that can be performed by machine will be mentioned later.

Image Segmentation by Colour

In previous chapters, we have relied heavily on thresholding intensity as a means of segmenting images. Of course, this relies on there being distinct regions of nearly constant intensity. Within such regions intensity variations are small, compared to the much larger differences that exist between them. In some applications that we will encounter, there is a distinct difference in colour but not intensity. (Figure 6.1) Is there a way in we can "threshold" colour? Indeed, there is and, in this chapter, we shall show how this can be achieved. Once coloured image features have been isolated, they can be measured as we have already explained, for characterising binary objects.

Colour Recognition

Our objective is to build a machine, or write a computer program, that is able to learn from samples of coloured objects shown to it. After an initial training phase, the machine should be able to recognise those same colours, whenever they are encountered again. The colour samples for learning are selected by a person, or a committee of experts, called the *teacher*. He/she/they is solely responsible for defining and naming colours for the machine to learn. Suppose, for example that we want to design a machine that is able to inspect dress fabric having just three distinct colours. The teacher may decide to call these *“red”*, *“green”* and *“blue”*. On the other hand, a German speaker would probably use other names: *“rot”*, *“grün”* and *“blau”* instead. (Remember that we saw earlier that colour terms do not always translate accurately from one language to another.) Such differences of nomenclature do not matter at all, because these are merely labels that the teacher uses for his/her convenience. What does matter is the ability of the machine to learn and later to identify colours in the same way as the teacher. We could just as readily use numbers as *colour names*. In fact, this is what an automated *Colour Recognition Filter* (*CRF*) will do. How those numbers are reconciled with familiar colour names in English, German, or any other natural language is totally irrelevant for the design of the CRF. This seemingly trivial point is important because it makes it clear that only the teacher can legitimately judge the effectiveness of the learning exercise. It is unreasonable for anybody, apart from the teacher, to criticise a well-designed CRF; if anyone wishes to disagree with its performance, then they should argue with the teacher instead!

Sample Applications

In a typical application of the type that we will discuss, there may be just a few distinct colour categories to consider. Furthermore, the variation within each class is relatively small. Just what this means in practice will become evident as we progress through this chapter. A picture playing-card serves as a convenient model for a large group of applications, such as plastic/cardboard containers printed using a small number of inks. The card illustrated in Figure

6.1[TL] was printed on white card, using five inks: *red*, *blue*, *yellow*, *“skin”* and *"black"*. [*"Skin"* is merely a convenient abbreviation for the flesh-coloured tone on the playing card] Notice that our machine must be able to identify two additional classes of colour: *“white”* and *“everything else”*. Identifying each of these involves only a *crude* form of colour recognition, that is able to distinguish clearly separable colours. By comparison, human colour vision is sophisticated, subtle and not very well understood.

Assumptions

In what follows, many of the complications and subtleties of human colour vision are avoided by making certain assumptions. We will therefore take it that there are no effects on the recognition of the colour of a single pixel as a result of

- Localised colour contrast
- Motion
- Fluctuating brightness or colour
- Colour adaptation
- Variable ambient lighting
- Observer fatigue
- Insufficient light for photopic (day-light) vision
- Variations in colour perception due to illness, ageing, alcohol or other drugs
- Language differences

Colour Recognition Filter (CRF)

By analogy with optics, we will use the term *Colour Recognition Filter* (*CRF*) to refer to a device/program/

function that accepts a colour image as input and generates an image as its output. The intensity of each pixel in the image generated by a CRF indicates the colour of the corresponding pixel in the input image. The association of output intensity values and classes of colour has absolutely no fundamental significance. For example, output level 67 might represent a group of colours that is very different from those associated with 66 or 68, while the colours represented by level 157 might might look similar. In what follows in this chapter, we will often display colour recognition results using pseudo-colouring. Again, do not place any significance on the choice of pseudo-colours, which is perfectly arbitrary. We will often display the results from several CRFs in a single image, which might, for example, delineate the limits of *"red"*, *"green"*, *"yellow"* regions in a single (natural) image)

The teacher might want to design a CRF that reflects the range of colours appearing in a factory. To do so, he/she/they might collect samples, or swatches, of factory product, packaging samples and assign unusual names to their colours, such as *"tuna can red", "margarine tub blue", 3M red, IBM blue, BP green, etc.* The nomenclature implicit in a formal colour-naming scheme, such as the Munsell Colour Atlas, might be used instead.

Image Enhancement

Although colour recognition is the main theme of this chapter, we begin by discussing simple ideas for modifying an image to enhance details of interest. This will reinforce our claim that the RGB components of an image contain all of the information needed to analyse its colour content. Just a moment's thought will show that this is so: television relies on RGB sensors and displays. Any other, perhaps more sophisticated representation of colour has to be translated into combinations of RGB signals. If this were not the case, you and I would not be happy watching colour television.

We will start with the simplest possible ways of analysing the RGB component images and will gradually build up to more complex ways of combining them. Eventually, we will reach a form of colour recognition filter that is potentially capable of emulating its teacher perfectly, within the limits imposed by using an RGB camera. Designing it is a bit more problematical!

Terminology

The monochrome image representing the R-component of a colour image will be called the *R-image*. The *G-* and *B-images* are defined in a similar way. Also, the monochrome image forming the H-component (hue) of a colour image will be called its *H-image.* The monochrome intensity image studied in earlier chapters will be called the *I-image*. We will explain in detail later how hue and saturation can be calculated from R, G and B. For the moment, let it suffice to say that this involves evaluating straightforward mathematical formulae.

Familiar Ideas

Figures 6.2 - 6.5 show the R-, G-, B-, H- and S-images for four sample colour images. In some limited situations, useful colour separation can be achieved by selecting just one of the R-, G-, B-, H- or S-images. Note the following points:

Figure 6.2 (Picture playing card)

[TR], R-image: “Red”, “yellow” and “skin” features are almost indistinguishable from each other and from white.
[CL], G-image: Not very useful. (There are no “green” features in the original.)
[CR], B-image: The “red” and “yellow” features are almost indistinguishable from "black".
[BL], H-image: Not very useful as hue fluctuates wildly in “white” areas.
[BR], S-image: “Red”, “blue” and “yellow” features are almost indistinguishable from each other. The intensity in the S-image fluctuates wildly in “black” areas. ”Skin” is clearly identifiable, since it is the only colour which maps to mid-grey.

Figure 6.3 (UK bank-note fluorescent security feature)

The pink and white areas are clearly distinguishable from each other and from the background.

Figure 6.4 (Citrus fruit)

The B-image and saturation both provide good contrast; the mesocarp and placenta can readily be distinguished from the endocarp and exocarp.

Figure 6.5 (Bare printed circuit board)

[TR] R-image: The bare copper pads are clearly distinguishable from the coated copper and substrate.
[BL] S-image: The bare copper pads are indistinguishable from the coated copper.

[BR] G-image subtracted from the R-image: The bare copper pads are highlighted.

The last of these gives a hint of what is to come: combining the R-, G-, B-images to obtain a better and more flexible approach to the discrimination of colours.

Adjusting RGB Separately

Many useful and interesting effects can be achieved by adjusting the RGB components of an image separately. Many standard image-editing programs, such as Adobe Photoshop™, provide interactive tools for doing this.

Figure 6.6 provides several examples of what can be achieved with quite simple adjustments of the RGB channels. In [TR] and [CL], the R-, G-, B-images have each been given a slightly different non-linear "tweak" [The R-, G-, B-images have been modified individually by adjusting the gamma parameter. See Chapter 4]

To produce [CR], each of the R-, G-, B-images was separately subjected to histogram equalisation. This can be a very effective tool for enhancing colour-texture images.

The bizarre appearance of [BL] was achieved by negating the R-image. Non-linear [square and square-root] transformations were also applied to the G- and B-image respectively.

In [BR], the G-image alone was adjusted by mixing it with an Intensity wedge.

Using these techniques, it is easy to produce pictures in the style of Andy Warhol's famous false-colour portraits of Marilyn Monroe. (Figure 6.7)

Brief Digression

Colour-balance adjustment is one of the primary methods of enhancing images for forensic photography, advertising, fashion and other photography. It is also useful for correcting the appearance of an object that has been photographed in tinted light. Figure 6.8 shows the effects of viewing three different surfaces under three different light sources. The term *"White LEDs"* refers to devices containing LEDs that emit short-wavelengths, illuminating a phosphor with a flat emission spectrum. *"R, G & B LEDs"* refers to tri-colour devices that emit light in three narrow spectral bands; the brightness of each one can be adjusted separately. A standard digital camera allows the adjustment of colour-balance to suit a variety of different lighting conditions. The camera has a built-in hardware/software facility for making separate adjustments to the RGB signals automatically.

The human visual system also compensates for variations in the colour of ambient lighting, without our being aware that it is happening. This is called *Colour Constancy*. Be aware that some cameras have in-built colour correction, which may not be obvious and could confuse a learning system.

Segmenting Coloured Images

However, this has little direct relevance for our primary objective: building vision systems that can inspect, or measure, industrial artifacts. For this purpose, the real need is for increasing the range of methods for segmenting coloured images.

When we consider grey-scale images, there were two approaches:

Edge-based, use an edge-detector function

Region-based. The main tool is thresholding.

We can now extend this choice to include both *colour-edge detection* and *region identification based on colour*.

Detecting Edges

When we are presented with a colour image, an obvious approach is to convert it to a grey-scale version and then apply an edge-detector, such as one of those discussed in Chapter 4. However, this does not always detect edges in a satisfactory way. Figure 6.9 illustrates this. Sometimes, it is better to select just one of the RGB channels before applying the edge detector. However, this can lead to, say, a red-mauve edge being missed in the R-image. [CL]

In [CR], green-cyan, cyan-yellow and yellow-grey edges are almost invisible in the G-image.

The images obtained by applying an edge detector to the R-, G-, B-image individually can be combined to give a better result. In [BR], we see that all of the edges have been detected successfully by doing this.

Figure 6.10 shows colour edges found on an image derived from a piece of dress fabric. This time, combining the results of applying edge detectors to the G- and B-images gives good results, even though the R-image has been ignored.

Colour Clusters

In order to understand how a vision system recognise colours, we need to introduce the concepts of RGB space and Colour Scattergram. These will provide us with a convenient way to visualise and understand the range of colours present in an image and are important tools in helping to design an effective colour recognition filter.

For a given pixel in a colour image, a standard camera produces three outputs, which we will denote by *r, g and b.* Together, they define one point in a 3-dimensional space, called *RGB space*. Now, let us project every pixel in the input image into RGB space. This generates a cloud of points, generating the RGB scattergram. (Figure 6.11) An image containing block colours that are obviously distinct from one another, will generate a cloud with dense local concentrations (clusters) of points in RGB space. Other regions are relatively sparsely populated. (Figure 6.11 and Figure 6.12) Clustering occurs because pixels that have similar colours are mapped into points in RGB space that are close to each other. In order to understand the cluster structure properly, more that one view of the the RGB scattergram may be needed. (Figure 6.13) [The MATLAB software package was used to produce these scattergram plots and enables us to perform a controlled "flight" around the RGB space.] Each point in the RGB scattergrams shown in Figures 6.11 - 6.13 has been rendered with the corresponding pixel colour. This emphasises the fact that different parts of RGB space can be associated with the familiar named colours: "red', "green", "blue", "orange", etc.

Consider the cloud of points in RGB space associated with the surface of a ripe ("red") tomato. This scattergram has a single distinct cluster, which we shall denote by T. Suppose we now generate the RGB scattergram for a "red" pepper. This creates another cluster (P). Since "tomato red" and "pepper red" are perceived as being similar shades of "red", we might expect clusters T and P to be very close to one another, perhaps even overlapping. That part of the RGB space associated with "red" (R) is bigger than both T and P; R also includes *"blood red"*, *"strawberry red"* etc. With enough time, we could define a closed surface in this space that would enclose all examples of "red" that *you* recognise and no other colours. It might be conceptually helpful to think of this surface as consisting of several overlapping spheres, each one enclosing a subset, such as "tomato red", "pepper red" etc. Notice however, that you and I would not agree exactly about where those spheres should be placed, nor how big they are. A colour recognition filter using overlapping spheres to make decisions about what is "red" or "not "red" is a real possibility. However such a decision surface is defined in mathematical terms, it must be designed by analysing numerous colour samples that somebody collects. In practice, a Colour Recognition Filter must be designed by learning, not by programming.

As in any other language, there is a limited number of names in English for colours, so it is common practice to invent new names. This gives rise to terms such as *"buttercup", "canary", "sulphur", "mustard",* etc, which are, of course, all examples of *"yellow".* For this reason, we must not limit a colour recognition system to identify the *"seven colours of the rainbow"*. The machine must be able to learn and then identify more esoteric colours, such as *"pen-top green"*, *"candy-box orange"*, *"peanut-packet blue"*, etc. We must content ourselves with the fact that each application requires a separate learning phase for the Colour Recognition Filter.

In Figure 6.14, the RGB scattergram for an image with three similar shades of yellow and no other colours is viewed from four different directions. The outlying points around the main clusters have been removed for clarity and the clusters have been assigned false colours for the same reason. In this example, the three clusters, representing different shades of yellow are clearly separate. This fact gives us hope that a CRF can be designed to perform quite subtle colour recognition.

Application: Printed Card

A "picture" playing card provides a convenient model for demonstrating colour clustering. The playing card shown in Figure 6.15[TL] shows the result of five non-overlapping imprints: "red", "yellow", "skin", "blue" and "black". The RGB scattergram is plotted in [TR]. This image, which contains only block colours, generates one cluster in RGB space for each ink used in printing. Figure 6.15 also makes a valuable point: in some instances, the RGB Scattergram can be replaced by a 2-dimensional plot. [BL] is easier to visualise and it can be processed just like any grey-scale image. In the example illustrated here, almost all of the clusters are separable in the RB plane. (G is ignored.) However, the clusters corresponding to "red" (r) and "yellow" (y) do overlap in the RB plane and are not separable without reference to the G component as well. Notice that pseudo-colour [BR] helps us to visualise sparsely populated regions. The significance of Figure 6.15 is that it shows that, in some situations, useful colour discrimination can be obtained using just two of the RGB components, in this case R and B. Notice however, it is not always possible to separate colours satisfactorily without using all three components.

Application: Monitoring Ripening of Fruit

Before we go on to discuss colour recognition in detail, let us digress for a moment to look at how the RGB space can help us to understand and quantify the colour changes that take place as fruit ripens. Imagine that a camera is focussed on a single tomato, still hanging on the vine. Let us consider a single point on the tomato during the summer and early autumn, as its colour changes from green to yellow to orange to red. During the ripening season, the RGB values of that point effectively trace a path through the colour space. If we now consider the whole tomato as it ripens, the time-varying collection of RGB vectors describing its overall surface colour traces a broader snake-like path.

I did not have access to growing tomatoes, so the point is illustrated using a golden-red apple that shows the effects of varying degrees of exposure to the sun. Some parts of the apple are riper (redder) than others. Figure 6.16 plots the RGB clusters for yellow and red regions of an apple.

Between them is a region where the colour is in transition from yellow, where the fruit has not been exposed much to direct sunlight, to red where it has been so. It is clear that the “transition” region on the apple surface produces a colour cluster that lies between the “yellow cluster” and “red cluster” in RGB space.

In Figure 6.17, an image derived from seven bananas at various stages of ripening is analysed using only R and G. The R-image shows an increase in intensity as ripening progresses. (not shown) On the other hand, the G-image shows a decrease. [CL and CR] The difference between the G-image and R-image, [BL], shows more significant changes of colour. Ripening traces a "folded" path, resembling the letters "N" or "Z", in RGB space. This can be seen using an interactive display but is not easy to illustrate, in a single static 2D image. [TR]

Towards Colour Recognition by Machine

We have established that there is a relationship between human colour perception and position within RGB space. However, we do not know exactly which part of that space should be associated with a given colour as named by an individual person. Let us restrict our attention for the moment to distinguishing between categories such as "yellow" and "not yellow". This should be possible if we can enclose those RGB scattergram points corresponding to "yellow" within a closed surface, such as a set of overlapping spheres. (Figure 6.18) This method of making decisions is called a *Compound Classifier*. If the RGB colour vector defines a point inside any one, or more, of the spheres, the decision of the classifier is *"yellow"*, otherwise it is *"not yellow"*. To optimise a compound classifier, we must decide

How many spheres are needed
Where each sphere should be placed
How big each sphere is.

Designing a colour recognition filter (CRF) based on a compound classifier manually is tedious. Since it involves simultaneously adjusting many variables, this will almost certainly not lead to an effective filter. A better solution is to use an iterative, self-adaptive learning procedure. Several procedures have been devised for this but require a lot of detailed explanation involving mathematical notation. For this reason, it will not be described here. Details can be found at [B G Batchelor, *Practical Approach to Pattern Classification*, Plenum, London, 0-30630796-0, 1974, ISBN]

An alternative approach is to use a random access memory (RAM) chip is to remember all of the colours assigned to all of the points in RGB space. (Figure 6.19) A commonly used standard for representing images in digital form ("24 bit colour") uses 8 bits to represent each of the RGB components. This allows 16,777,216 different colours to be defined and is adequate for almost all video applications. Recall that a standard memory card for a digital camera has a thousand times this capacity and costs about €20. A CRF requires only a very small proportion of the storage available in a standard desk-top computer

Hue Saturation and Intensity

So far, we have assumed that we need to take intensity into account when recognising colour. In practice we can often discard intensity; over a wide range of brightness levels, human beings are able discriminate colours on a surface independently of brightness. This will result in a simplification that will allow us to gain an important intuitive understanding of the range of colours in an image through the use of a 2-dimensional *Colour Map*.

Hue, saturation and intensity (HSI) were introduced in Chapter 3 with no detailed explanation about how they can be calculated from the RGB components. Video cameras generate RGB signals naturally because there are three different light sensors, each with a different colour filter in its optical path. The HSI method of characterising colour was devised as a superior means of modelling the human visual system.

Figure 6.20 shows the RGB space with the so-called *Colour Triangle* superimposed on it. [This is sometimes called the *Maxwell Triangle* in honour of James Clerk Maxwell.] At the point O in [TL], the values of R, G and B are all zero. All points along the line OPQ have the same hue (H) and saturation (S) values; only intensity (I) varies along this line. P is the point of intersection of this line with the colour triangle. The HS values for all of the points along this line can be specified by the position of P within the colour triangle. H is measured by the angular position of P relative to some arbitrarily chosen reference axis. In Figure 6.20 this is the line from the centre of the colour triangle (C) to the corner corresponding to red. Notice that, H is measured by an angle, and is therefore cyclic. Saturation is measured by the distance from C to P. Intensity is measured by the distance along the line OQ. These are conceptual descriptions of HSI; the precise mathematical details do not matter for our present discussion. Let it suffice to say that

HSI can all be calculated by evaluating standard formulae based on RGB.

Colour Recognition Filter (CRF)

A colour scattergram in the HS plane gives us the information we need to design an effective colour recognition filter. Designing the CRF will benefit from the fact that the 2-dimensional HS scattergram indicates the range of colours in an image just as well as the 3-dimensional RGB scattergram but is easier to understand and manipulate. We will see several examples of HS scattergrams later.

The colour recognition filter employs a so-called *Colour Map*. This can be derived in a number of different ways, often by applying familiar image processing operators to the HS scattergram. Sometimes, a more effective colour map is obtained by employing human intelligence: manually drawing around clusters in the HS scattergram. The colour map consists of a grey-scale (or pseudo-colour) image. Each entry in the colour map is a number, which we may associate with one of the familiar colour terms, such as *"blue", "yellow", "cyan", "magenta"*, etc.

A CRF has two distinct phases of operation.

Training: Plot the HS scattergram. Process/ annotate it to derive the colour map

Naming colours: For each pixel. consult the colour map to find the appropriate colour “name”. (Figure 6.21) This may be a single number (grey-scale intensity value), or three numbers (pseudo-colour component values).

Notice that a colour recognition filter can be implemented using a single look-up table, although Figure 6.21 suggests that two are needed. In other words, Figure 6.21[BL] is therefore an implementation of the idea implicit in Figure 6.19

There remains one important and as yet unspecified task: deriving the colour map. This may be done by applying image processing functions to the scattergram in the *HS plane*, manually drawing on it, or a combination of the two

It is important to note the following points:

- Programming a colour recognition filter is based on a collection of colour samples chosen by a person (or a committee of experts) who we call the *Teacher*.
- There is no universal design procedure for a machine to recognise colour; it involves experimentation and interaction between the teacher and the device/program implementing the CRF. (Figure 6.22)

After collecting the colour samples, the next step is to generate the HS scattergram. (Figure 6.23[TR]) This is then processed and interpreted to create the colour map. (Figure 6.23[CL]) A lot of experimentation is sometimes needed to generate a really useful colour map. This typically involves

- Blurring, to “join up” fragmented spots in the HS plane
- Thresholding
- Noise reduction by binary filtering
- Blob-size adjustment. Making blobs in the colour map bigger allows the filter to accept a wider range of colours. (This is called *Colour Generalisation*.)

- Pseudo-colouring (This is really only useful as an aid to understanding, as it does not provide a good basis for further image processing).

We emphasise that this is just the beginning of what may be a long experimental process to design a CRF..

Figure 6.23 illustrates a typical CRF design. Notice that a CRF can recognise single colours, as shown in Figure 6.23[BL] (“red’) and [BR] (“skin”, short for "white skin"), or multiple colours in [CR]. We have used pseudo-colours for illustrative purposes only. Figure 6.23[CL] & [CR] could be presented in monochrome, which would not be so clear - or pretty!

Colour Generalisation

Figure 6.24 shows that a CRF is potentially able to make fine colour discriminations. The HS scattergram for these three similar shades of yellow has three fuzzy but separate clusters. [TR] Thresholding the HS scattergram produces three separate blobs, which have been pseudo-colour coded for clarity of illustration. [CL] This image was used as the colour map for the CRF, which was then reapplied to the input image. The three yellow areas are separated quite well, although there are some black areas indicating pixels whose colours were not recognised. [CR] The blobs in the colour map were then enlarged. [BL] The CRF then generated an image has far fewer black pixels. [BR] Initially the CRF recognised quite specific colours because the blobs in the colour map were quite small. By increasing their size, it accepts a wider variety of similar colours. Hence, we use the term *Colour Generalisation*. Figures 6.24 - 6.32

present results from images derived from a range of different scenes.

Representations of Colour

Colour is a concept; it exists only in the human mind and does not have a well-defined physical reality Two people may disagree about the colour name that they should associate with a particular object surface. More people may agree with you about the naming of a set of colour samples than they do with me. You may be able to distinguish more colours than I can. (I am not colour blind.) But there is no question about who is correct about naming a colour, neither of us is “right” and the other “wrong”. However, in the context of thof our present discussion, we must take one person’s opinion as defining “the truth”. To obtain a consensus, we might prefer to rely on a vote in a committee. We refer to the person/committee who will define the colours for our experimental work as the “teacher”. He/she/they will typically collect a set of samples of the particular product that we wish to inspect. The experimental work is then based on this collection alone. In this way, the teacher defines “the truth”.

When designing a Colour Recognition Filter, we try to emulate the teacher’s ability, for example, to inspect bananas, we need to distinguish *"yellow"* from *"not yellow"*. As far as we are concerned the latter is a valid colour category. How could we convey the result of such a study to another person, perhaps in another country, where the natural light and working language are different? One possible approach is to match the colour at one point on each banana to a set of standard colours. Such a collection forms what is termed as called a *Colour Atlas*. This would be extremely laborious and unsatisfactory, because we are interested in the whole banana surface, not just one point. The CRF provides another approach and is based on the following assumptions:

- The teacher cannot properly define a colour class, such as *"yellow"* but can select numerous samples of what he/she/they recognise(s) as that colour.
- We can buy good-quality RGB cameras cheaply. The high image quality achieved by standard off-the shdigital cameras provides ample evidence of this.
- The design and deployment of a CRF are performed in fixed lighting conditions.
-

Figure 6.33 shows how information flows during the various stages of designing a CRF. It is evident that "colour" is represented at several different levels:

- *Mental* The concept of a colour, such as *"yellow"*, originates in the human mind but it cannot be defined precisely.
- *Physical 1* A plentiful collection of colour samples is gathered by the teacher
- *Physical 2* Light emitted by a lamp of constant colour falls on a surface. The spectrum of the light scattered from that surface has been modified by its chemical and physical make-up.
- *Video* The camera converts the light it receives from that surface into a pattern of electric charges. This is, in turn transformed, into a digital format, with three components: R-image, G-image and B-image.

- *Geometric 1* A scattergram is created in the HS plane. This is a grey-scale picture, typically containing a few fuzzy spots and ill-defined cloud-like features.
- *Geometric 2* The HS scattergram image can be processed, for example using techniques such as those described in Chapter 4. Typically, it is filtered and thresholded. Deciding what processing is appropriate here is a matter of human judgement,
- *Geometric 3* The thresholded image consists of a number of blobs and some snow-like noise. The latter is usually removed by judiciously chosen binary morphology operations, similar to those discussed in Chapter 5.
- *Geometric 4* The remaining blobs are enlarged to achieve some generalisation during the colour recognition process. Again, this relies on human judgement.
- *Iconic* The enlarged blobs are approximated by a set of overlapping discs. Each of these is defined by its radius and two position coordinates.
- *Parametric* The list of disc-radii and position coordinates allows the blob structure to be represented approximately by a relatively small number of numeric values.
- *Symbolic* Finally, we can associate this parameter list with the teacher's name for the colour.

So, a colour class, such as *"yellow"* can be described by a sequence of numbers in the following format:

{[R1,X1,Y1], [R2,X2,Y2], ..., [Rn, Xn,Yb]}

where

[[X1,Y1], [X2,Y2],…,[Xn,Yn]] define the disc positions

This allows a CRF to be reproduced exactly, any number of times. Of course, the duplicate CRFs have exactly the same deficiencies, compared to the teacher, as the original.

Addendum: Lighting

The accuracy of a CRF is critically dependent on the stability of the lighting, in terms of both colour and the direction of illumination. While the human eye is able to compensate for subtle changes in the colour of lighting, a camera does not naturally do so. For this reason, subtle changes in lighting might cause a major malfunction in a vision system may but go totally unnoticed by personnel nearby. The sensitivity of camera-based systems to the colour of lighting is demonstrated in Figure 6.8. In these example, the chemical composition of the surface material determines its response to lighting variations. In Figures 6.34 & 6.35 periodic micro-structures are responsible for major alterations in the appearance of a surface. This effect is not uncommon and, if these materials are to inspected by machine, they must be viewed under carefully controlled lighting conditions. Both the position and colour of the light source(s) must be constant. Many types of lamps change colour as they age, so this must also be taken into account when designing a CRF.

.