Chapter 4

4 Enhancing Tracking Using Colour

4.1 Introduction

The colour content of an image is an important attribute, which is often discarded. Common practice in the processing of PDMs and snakes is to merely assess the intensity of pixels, processing as if grey scale i.e. calculating the mean intensity of the red, green and blue colour channels.

This chapter will discuss how colour can be used to enhance the appearance of objects in tracking algorithms. It will also be demonstrated how colour alone can provide a reliable feature for locating and tracking moving objects. Section 4.2 will demonstrate how the simple weighting of colour channels can be used to enhance specific features within an image. Section 4.3 will discuss the use of perceptual colour representations (alternative colour spaces to red-green-blue, RGB). Section 4.4 will discuss the advantage of colour in delineating regions. Section 4.5 shows how more complex colour models can be constructed and used to locate and track a humans. Section 4.6 demonstrates how these ideas can be extended to provide a reliable, computationally inexpensive solution to head and hand tracking, although these techniques extend to any colour object. Finally conclusions are presented.

4.2 Weighted Greyscale Images

In the previous chapter it was shown how high intensity edges could be located locally along a boundary. These high rates of change in pixel intensity were located by assessing the first or second derivative of the intensity along a normal to a boundary. This calculation is normally performed upon the grey scale values of pixels. However, as has already been mentioned, the ready availability of colour provides a far more distinguishable difference between foreground and background objects within an image. By performing processing upon a grey scale representation, calculated from the colour channels (typically the average intensity of the three colour channels) a considerable amount of information about object boundaries is lost.

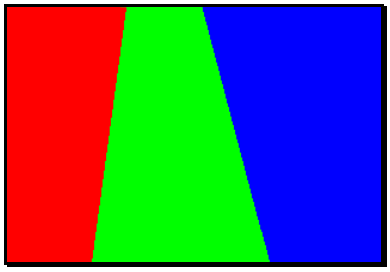


Figure 4.2.1- RGB image of iso-intensity

Figure 4.2.1 shows an image consisting of three colour regions. Each region has the same intensity in its colour channel: the red area has r=255, g=0, and b=0; the green area has r=0, g=255 and b=0; etc. By taking the average of the three colour channels at each pixel, the resulting image would have a constant intensity of 85 and no distinction would be possible between the various areas. However, in the colour image, it is visually apparent that such a distinction does exist and very clear boundaries are defined.

It is clear that reducing the colour information to one channel literally 'throws' information away, information which may be invaluable to the application at hand. One solution to this would be to process each colour channel individually. This can be done by assessing normals for each colour in turn, calculating three second order derivatives, and taking the average, where

$$d^2 I_i = \frac{R_{i+1} - 2R_i + R_{i-1} + G_{i+1} - 2G_i + G_{i-1} + B_{i+1} - 2B_i + B_{i-1}}{3}$$

However, this is still an averaging approach and as such will smooth edges. In addition, the approach effectively requires each normal to be assessed three times and hence results in a significant decrease in speed.

If an object of interest is sufficiently prominent within one of the colour channels, then the intensity of that channel can be used instead of the mean intensity.

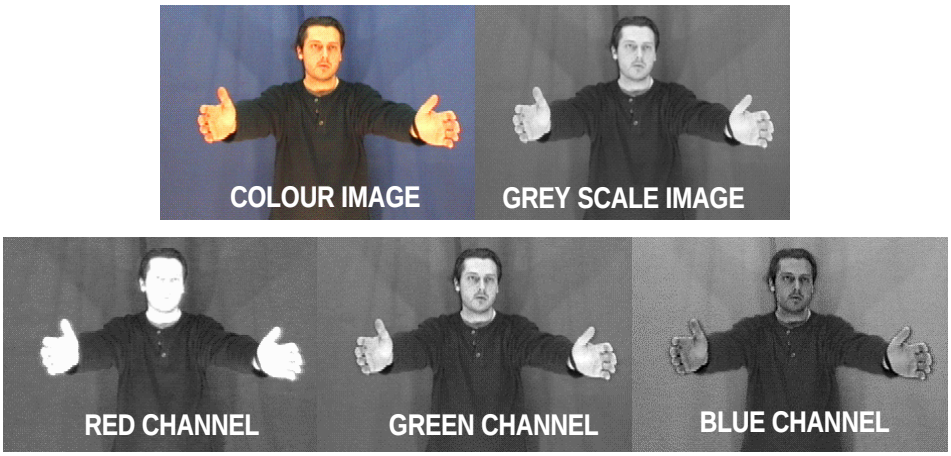


Figure 4.2.2 - The Separate Channels of a Colour Image

Figure 4.2.2 shows a colour image of a person in front of a blue backdrop, along with the grey scale version of the image and the three separate colour channels shown as grey scale intensity images. The grey scale image retains much of the distinctions between regions seen in the colour image due to the small number of highly distinct regions and the uniform background. The individual colour channels, however, each emphasise certain aspects of the image. The blue channel has a lighter background than red or green with a lower contrast figure. This is to be expected, as the blue background will generate high intensities in the blue channel. The red channel emphasises the skin regions of the subject, due to the high red component in skin tones. If the object to be located or tracked within the image were hands or head then using the red channel for image processing would produce far superior results than tracking on the mean intensity

(as the mean intensity effectively smoothes out this distinction). However, simply processing upon the red channel may disregard other important features. In addition, other channels could potentially be used to subdue features that are not desirable, i.e. the background. As it is known that the background is depicted best in the blue channel, subtracting this from the red channel will further increase the distinction between regions.

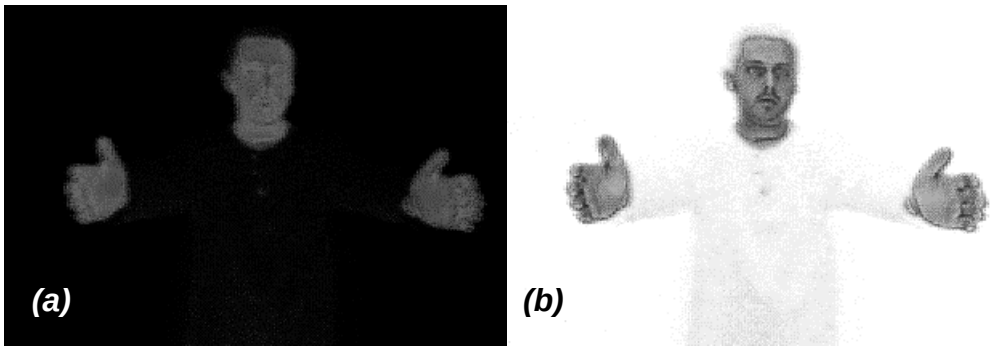


Figure 4.2.3 - Enhancing features Using Colour Channels

(a) Blue channel subtracted from red (b) Inverse of (a)

Figure 4.2.3(a) demonstrates the results of subtracting the blue channel from the red channel. Figure 4.2.3(b) shows the inverse of (a), which improves the visualisation of the distinction between regions. Although the overall contrast between skin and the surrounding area appears less, the segmentation of the skin from the overall image is greatly enhanced. The background and body have almost completely been removed.

If the simple conversion to grey scale is formulated as the average pixel intensity of the three colour channels, this can be expressed as

$$I_{x,y} = \frac{r_{x,y} + g_{x,y} + b_{x,y}}{3}$$

then subtracting the blue channel from red can be expressed as the weighted average of the pixels,

$$I_{x,y} = \frac{\alpha r_{x,y} + \beta g_{x,y} + \chi b_{x,y}}{\max\left(1, \left|\alpha + \beta + \chi\right|\right)}$$

where

$$\alpha = 1, \beta = 0, \chi = -1$$

by tailoring these colour coefficients for specific applications, features can be enhanced or subdued as required. Figure 4.2.4 shows the results of further enhancing the skin regions by applying the coefficients $\alpha = -2, \beta = 0, \chi = 2$.

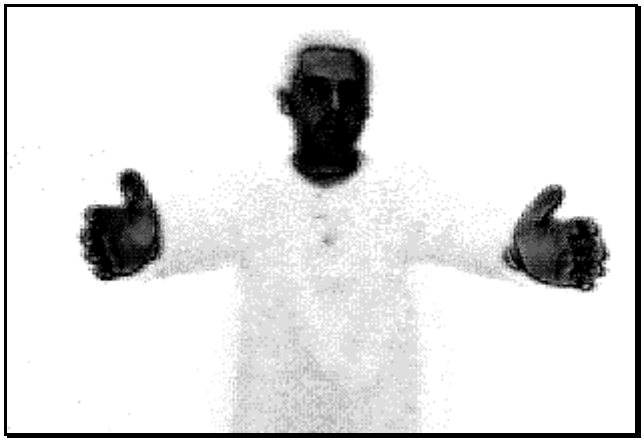


Figure 4.2.4 - Enhancing features Using Colour Channels

4.3 Perceptual Colour Spaces

The RGB-colour space (typically used in computer applications) allows three primary colour channels to be used to specify up to 16.7 million colours by representing the colour space as a 3D-colour cube (each channel having 256 discrete intervals). This provides a simple mechanism for constructing and representing a broad spectrum of colours. However, this is not an intuitive representation in terms of human perception, where similar colours (as judged by the eye) may occupy completely different areas of rgb-space. This is confirmed by the initial observations made from Figure 4.2.1. It has already been noted that the intensity of each colour region has the same value, even through the distinction between the areas is visually apparent. Furthermore, the central green region looks brighter to the human eye than either the red or blue regions. The notion of a perceptual colour space is to model the colour volume so to better correspond with how the human eye perceives colour and relative intensities.

Discussions of colour perception usually involve three quantities, known as *hue, saturation* and *lightness*. *Hue* distinguishes among colours such as red, green and purple. *Saturation* refers to how far colour is from a grey of equal intensity, i.e. red is highly saturated, pink is not, although both have similar hue/red-component. *Lightness* embodies the achromatic notion of the perceived intensity of an object. These perceptual colour spaces include Hue, Saturation, Value (HSV) (or HSB for Brightness); Hue, Lightness and Saturation, (HLS) (or *HSL* for Luminosity); and Hue, Value, Chroma, HVC [Foley 1990].

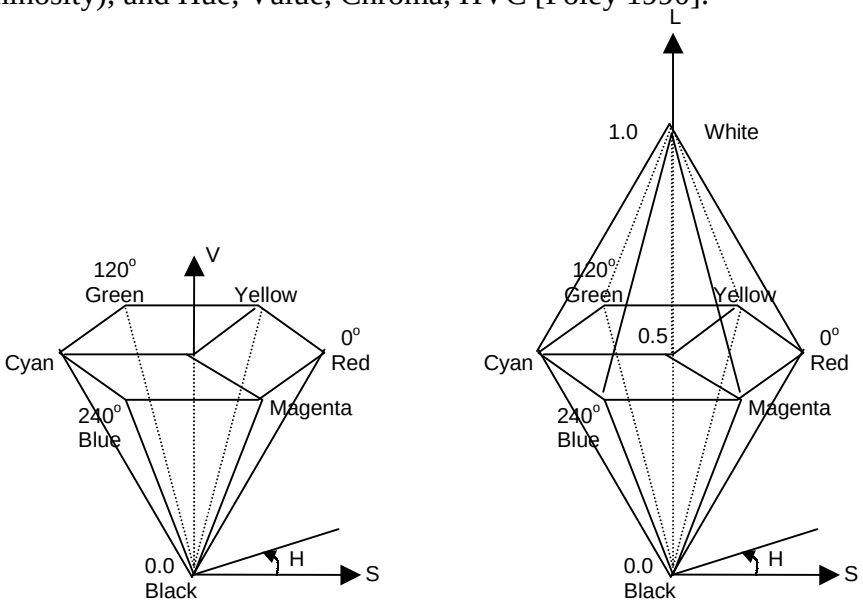


Figure 4.3.1 - HSV and HLS Colour Spaces

Hue **S**aturation **V**alue (HSV or HSB) colour space is a hexcone or six sided pyramid where Hue is the angle around the vertical axis, S is the distance from the central axis and V is the distance along the vertical axis. Colours along the vertical axis have zero saturation and are therefore grey scale values. **H**ue, **L**ightness **S**aturation (HLS or HSL) colour space is a double hexcone and can be thought of as a deformation of the HSV space.

The notion of separating colour from intensity provides a more robust method for colour feature extraction. Where colours change from shading or lighting differences, it would be expected that this would result in changes in intensity but not in colour.

Figure 4.3.2 – Separate Channels of HSL Image

Figure 4.3.2 shows the same colour image from section 4.2 converted in *hue, saturation* and *luminosity* with each channel shown as an intensity image. It can clearly be seen that the difference between the areas of the image is far more distinct in both hue and saturation than in any of the rgb colour channels (Figure 4.2.2). The saturation image provides excellent segmentation between the skin and other areas of the image frame, producing a distinct boundary between the skin and background elements.

Some devices provide colour space conversions in hardware. However, for the most part this must be implemented in software. For real-time systems where each pixel must be transformed independently, this overhead can become a significant speed-limiting factor. However, with contour based approaches this conversion does not produce a significant overhead, as only pixels along normals to the contour are assessed and hence need conversion.

A similar coefficient weighted expression to that demonstrated for rgb space can be used in HSL space, where

$$I_{x,y} = \frac{\alpha h_{x,y} + \beta s_{x,y} + \chi l_{x,y}}{\max\left(1, \left|h+s+l\right|\right)}$$

Provided hsl values are normalised to the range $0 \rightarrow 1$.

Further extensions can be made by combining both RGB and HSL weighted techniques. However, coefficient selection becomes a complex task. Instead, a more generic, automated method of enhancing/extracting features is required.

4.4 Colour Thresholding

As was demonstrated in the previous section, areas of skin produce high values in the saturation channel of the HSL colour image (Figure 4.3.2). These high areas can be used to threshold the areas of skin from the image in a similar manner to grey level thresholding. This technique is not dissimilar to chroma/luma keying.

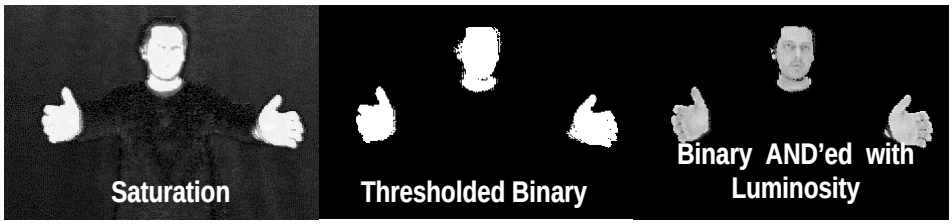


Figure 4.4.1 – Thresholded HSL Image

Figure 4.4.1 shows the saturation channel of the colour image. As the areas of skin produce high values of saturation, these areas can be extracted simply by thresholding the colour saturation channel into a binary image mask. The white-segmented areas correspond to the location of skin within the mask. Figure 4.4.1 shows the results of taking the logical AND of the binary image with the luminosity channel and demonstrates how the head and hands can be extracted using colour saturation instead of intensity to delineate colour regions of the image while retaining the internal features of objects or regions.

It should be noted that although the head and hands consist of various colour changes due to the features such as eyes, nose and the effects of non-diffused lighting, few of these features are apparent (to the eye) in hue or saturation. This is due to the separation of the *colour* information from the *brightness or luminosity*. The *luminosity* contains the information of how bright a pixel is and the hue-saturation *h-s* pair provides the information about colour. Rather than performing thresholding in $\Re^3$ of *rgb*, it can be performed in $\Re^2$ of *h-s* space. This provides a slight computational saving but has the added advantage that with the intensity component removed, much of the lighting/shading differences are absent. This provides a more uniform colour space in which to work.

Discarding the *luminosity* component of the colour effectively compresses the *hsl* colour space down onto a two-dimensional hexagon. In this space, consistent colours of varying *luminance* will produce clusters on the *h-s* hexagon. By discarding the luminosity for HLS and the value component of HSV spaces, both spaces become compressed onto the two-dimensional hexagon and the distinction between the two spaces is lost.

4.5 Gaussian Colour Models

For a number of years, research at the School of Computer Science, Carnegie Mellon University has used normalised rgb colour spaces to probabilistically label and segment regions of skin from image sequences for the location and tracking of the human face [Waibel 94] [Hunke 94] [Yang 98]. They have demonstrated that human skin clusters in a small region of colour space: Human skin colours differ more in intensity than actual colour, and under certain lighting conditions, a skin colour distribution can be characterised by a multivariate normal distribution in a normalised colour space [Yang 95]. Rainer, Stiefelhagen and Yang use this colour labelling to provide a rough estimate of the location of a head within the image frame to initialise a model based gaze tracking system [Stiefelhagen 97] [Stiefelhagen 98]. The normalisation of the colour space removes much of the variability in skin colour between individuals and lighting inconsistencies such as shadows [Yang 98]. Ivins and Porril used a normalised rgb colour space to label and track, in real-time, various colour regions of an industrial robot arm [Ivins 98].

McKenna, Gong and Raja have extended this work on colour labelling into the HSV colour space [McKenna 97]. Using a Gaussian mixture model to represent the colour space, they have shown how multiple models for individuals can be used to probabilistically label an image and determine the most likely person present. Azarbayejani and Pentland have used similar methods in HSV colour space to automatically segment both the hands and head from stereo image pairs, and using this, calculate the position and trajectory in 3D space [Azarbayejani 96].

Work by these authors has shown that human skin naturally clusters in a small region in colour space. Hunke and Waibel show that in a normalised rgb colour space, statistical bounds can be approximated for colour clusters and used to segment the human head from an image [Hunke 94]. Using colour as a feature for tracking has several problems: firstly, the colour representation of a face obtained by a camera is influenced by many factors such as ambient light, object movement, and the effect of diffused and specular reflections of an object moving relative to a light source. Secondly, different cameras produce significantly different intensity responses for the same wavelength of light. Thirdly, video signal encoding standards, such as PAL or NTSC, do not respond to the full colour space and effectively flatten the resulting colour spectrums of objects. Finally, human skin colours differ in rgb space from person to person [Yang 98]. McKenna *et al* demonstrated how these problems could be partially overcome by performing probabilistic classification in HS space, where variations in intensity have been removed [McKenna 97].

Human skin actually occupies a small cluster in HS space regardless of race or skin pigmentation. Differences in skin tone are primarily expressed by variation in the intensity of the colour: once the intensity has been removed the *h-s* colour space that they occupy is remarkably similar.

In order to verify this fact, four subjects were taken from different ethnic origins. For each subject, pixels were sampled in *rgb* from the skin tones on the palm of the hand. The results can be seen in the two graphs shown in Figure 4.5.1 and Figure 4.5.2. These two graphs allow the visualisation of the volume of the rgb colour cube in which the samples lie. It is clear that a fairly distinct single cluster is generated by the samples. However, this sample occupies a relatively large sub-volume of the total colour space. This is due to the difference in intensity of the samples along its major axis i.e. the variation in intensity of the pixels across any one sample.

Each sample pixel was then converted into HSL space, the luminosity discarded and the results shown in Figure 4.5.3. The Hue-Saturation space shows a far 'tighter' cluster with little variation in either hue or saturation. It is also important

to note that this colour 'fingerprint' of human skin is now 2 dimensional rather than the original 3D-rgb space.

The large number of sampled pixels and similarity in each of the four ethnic skin types makes the comparison of each difficult. To simplify, the mean and standard deviations in each colour channel can be calculated by

$$\bar{r} = \frac{1}{n}\sum_{i=0}^{n} r_i \quad \text{and} \quad \sigma_r = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(r_i - \bar{r})^2}$$

Figure 4.5.4 demonstrates the colours generated for the skin of four subjects with varying racial origin and pigmentation.

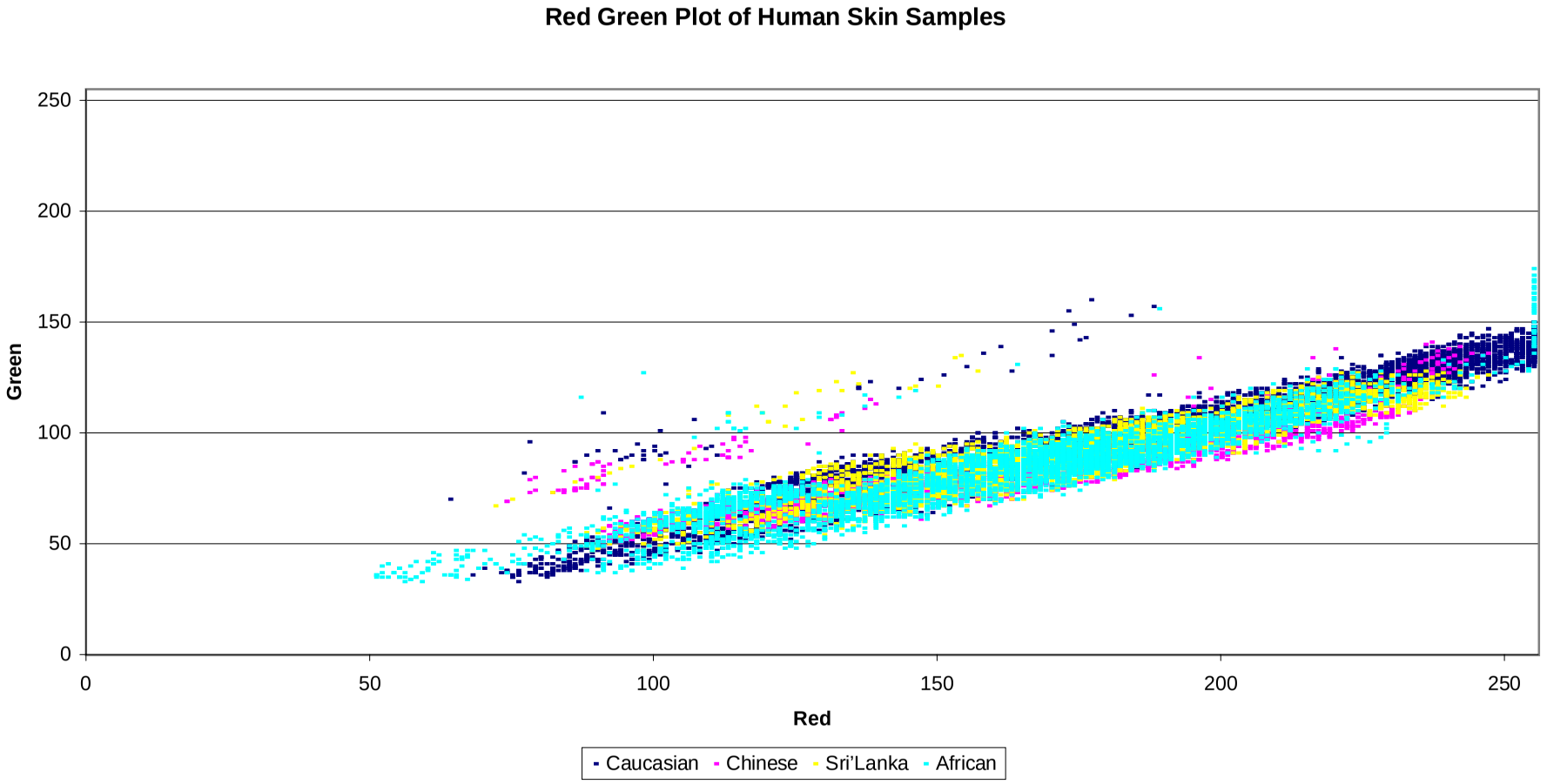


Figure 4.5.1 - Human Skin Samples Plotted in Red Green Space

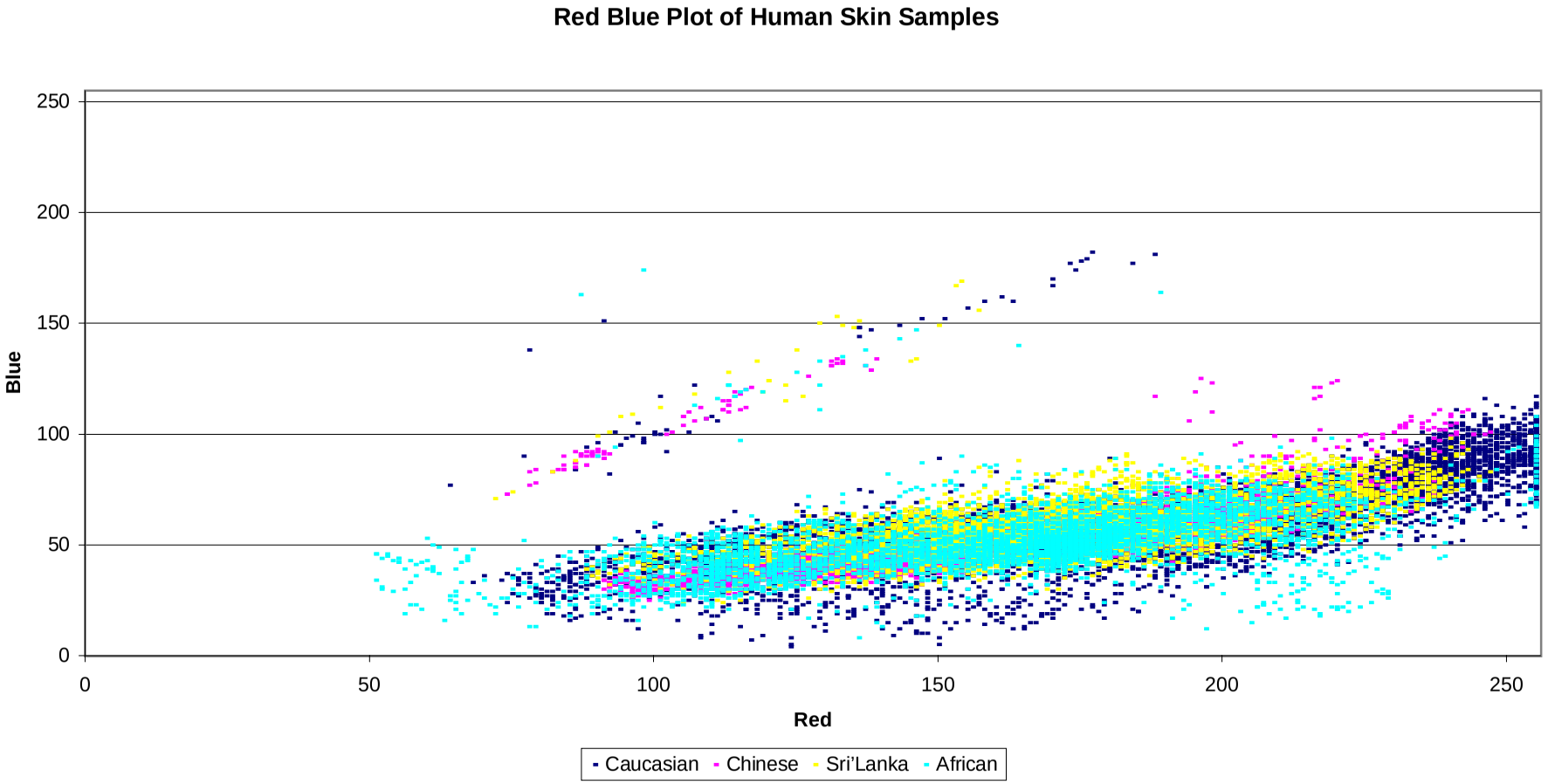


Figure 4.5.2 - Human Skin Samples Plotted in Red Blue Space

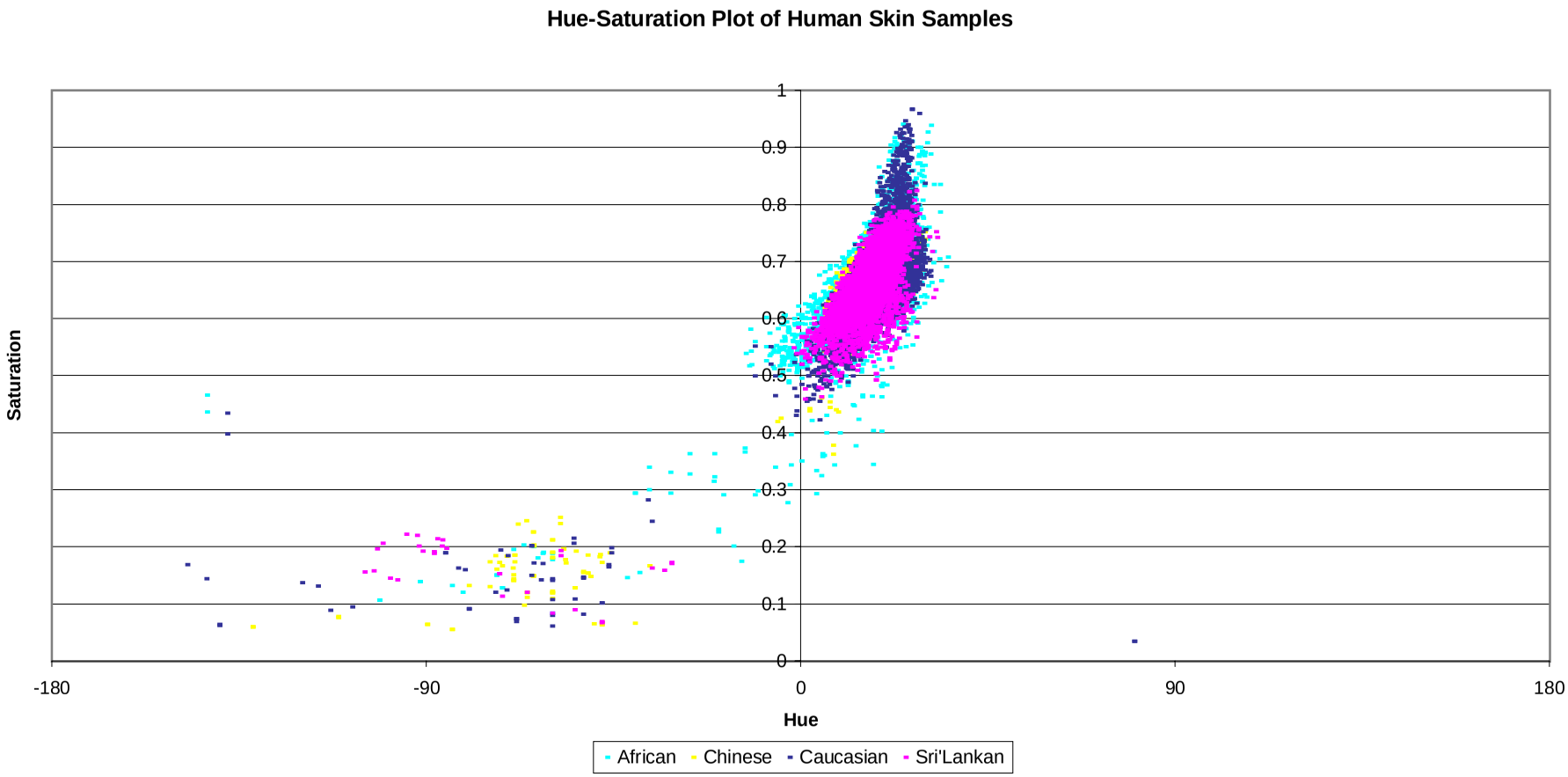
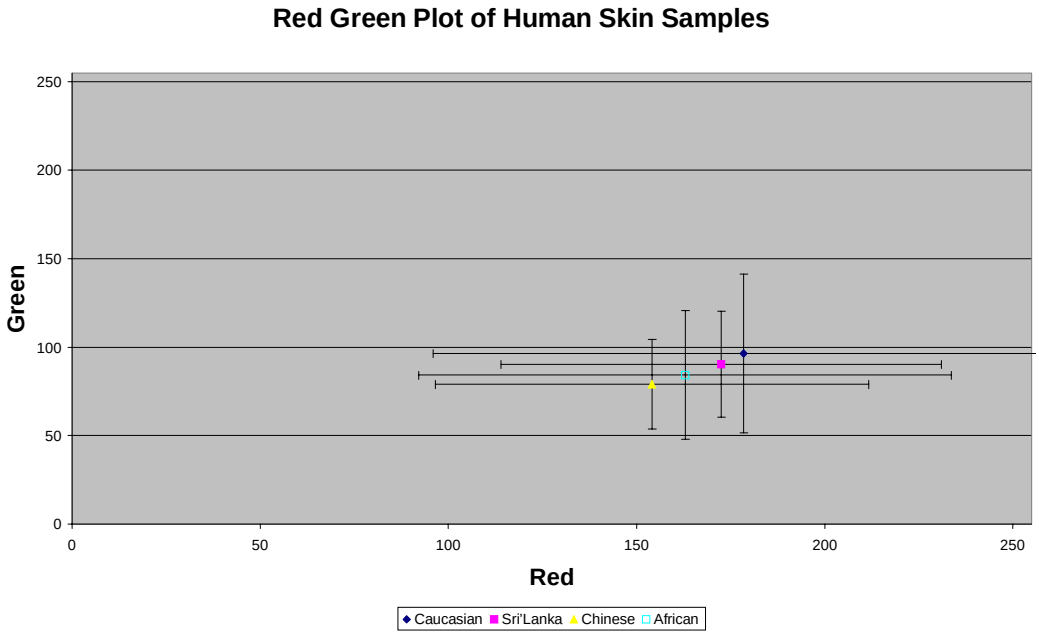


Figure 4.5.3 - Human Skin Plotted in Hue Saturation Space



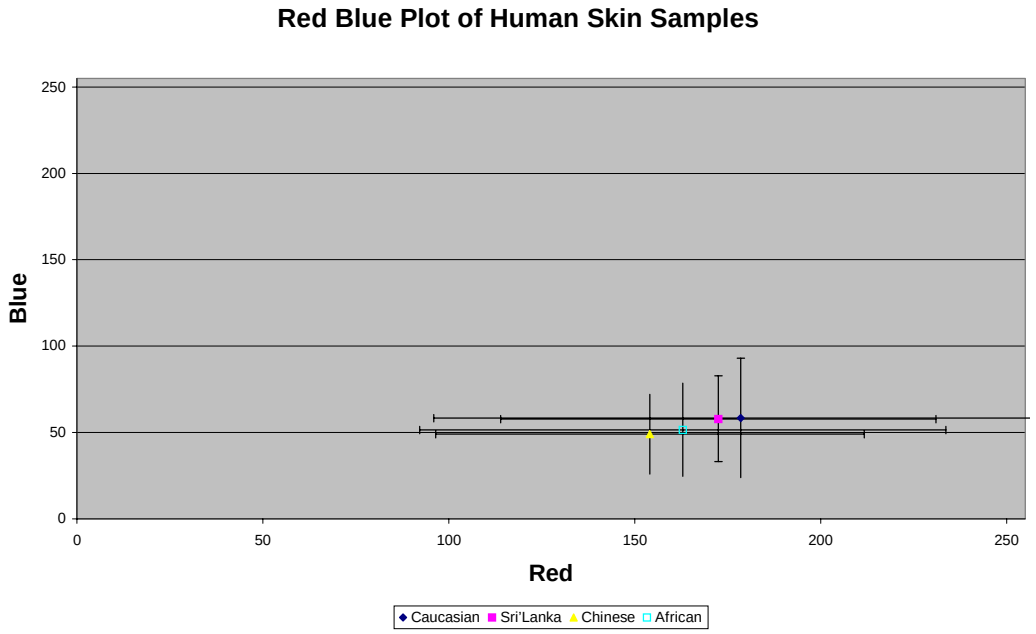


Figure 4.5.4 - Colour distributions of four skin types in r-g and r-b colour spaces

Figure 4.5.4 shows the mean value for each subject plotted with the error bars representing $\pm 2\sigma$. It can be seen in the Red/Green and Red/Blue plots that the various skin tones represent relatively small, overlapping clusters in *RGB* space, with subtle differences between subjects as would be expected. The darkest mean intensities are produced by the Chinese sample which would seem to contradict

any stereotypical observations about skin type. However, this is attributable to the distance of the hand from the camera during sampling. The Chinese sample was taken at a much closer distance than the other skin samples and hence produced darker results. However, this variation in lighting makes little difference to the results of the Hue Saturation plot.

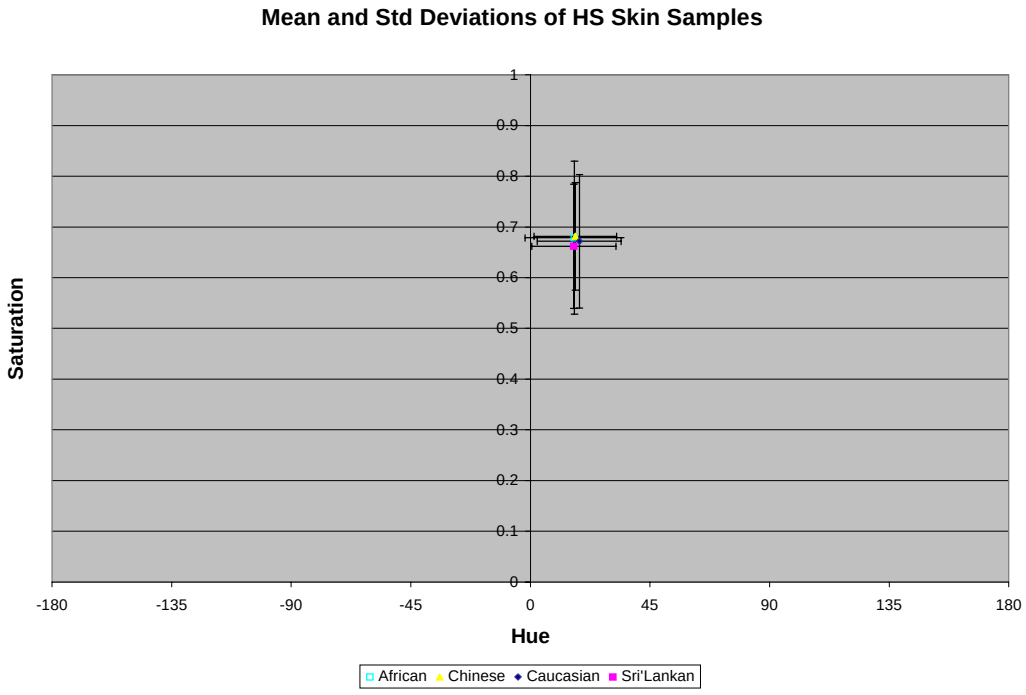


Figure 4.5.5 – Colour distributions of four skin types in HS space

The Hue Saturation plot shows the same statistical representation of the various skin types in *h-s* space. It can clearly be seen that this results in a far tighter colour cluster, which seems to vary little between skin types. Even the Chinese sample that produces dark results due to lighting is indistinguishable in the HS plot.

By using this single extracted cluster in HS space and fitting a multivariate Gaussian to it, a probabilistic measure that any pixel is human skin can be determined. A more accurate Gaussian PDF can be constructed by performing PCA on the colour cluster, and approximating its primary axis in addition to its bounds, or using the sum of Gaussians as used in chapter 5. If a sample pixel

from a new image is within the Hue-Saturation bounds of the Gaussian cluster then that pixel is marked as a probable location. Selecting a threshold for which probabilities of lower values are set to FALSE, and higher TRUE produces a binary image. By performing erosion then dilation, noisy points are removed and clusters of probable skin location consolidated into blobs. A simple blobbing algorithm can then be used to calculate approximate locations of skin artefacts within the image.

Figure 4.5.6 – Extracting Blobs of Skin

Figure 4.5.6 shows a sample image frame after processing. The results from the blobbing algorithm are used to calculate the centre of objects by finding the mean pixel of the blob and the approximate size by assuming circular blobs and calculating the radius of a blob from the area (i.e. the number of points in the blob). This is used to place a cross over the segmented features for demonstration purposes. In this instance the three largest blobs found within the image are deemed to constitute the head and the hands. The largest connected blob extracted from the colour labelled image can be used as a rough initial estimate for the position of the head.

4.6 Tracking Colour Features

Using a single Gaussian cluster to probabilistically segment skin tones from an image leads to noisy segmentations for two reasons:

1. The assumption that a single bivariate Gaussian is a good representation of the colour cluster is not completely valid.
2. Background clutter can be misclassified.

Specular reflections are particularly vulnerable to misclassification. Another draw back with the technique is that all the pixels of the image must be transformed into HSL space and colour classification applied. This process quickly becomes a computational overhead and when real-time applications are considered (25Hz or more) the approach becomes unfeasible.

One alternative is to locally search for skin using a Region of Interest (ROI) or window. Only pixels that fall within the ROI need to be converted and classified which significantly speeds up the procedure. In addition, background clutter, outside the ROI, cannot be misclassified. This produces a much cleaner segmentation without the need for erosion/dilation as previously described.

In order to limit processing to within the window (ROI), a mechanism for moving the window must be devised. This is itself a colour tracker, as the window must track the object in order to successfully segment the skin tones.

If the assumption is made that the binary-segmented object has a central white mass surrounded by black background, then the centre of gravity of the blob should be at the centre of the window.

Using a binary image window of size s_x, s_y where, $I_{x,y}$ is zero for the background and one for segmented skin, the centre of gravity for the segmented feature can be calculated by

$$CG_x = \frac{1}{\sum_x \sum_y I_{x,y}} \sum_{x=-s_x/2}^{s_x/2} \sum_{y=-s_y/2}^{s_y/2} x I_{x,y} \text{ , and } CG_y = \frac{1}{\sum_x \sum_y I_{x,y}} \sum_{x=-s_x/2}^{s_x/2} \sum_{y=-s_y/2}^{s_y/2} y I_{x,y}$$

A simple translation can then be calculated to position the centre of the window at the centre of gravity for the next iteration of the algorithm.

This assumption about the shape of an object within the window can also be used to calculate a new window size for the next iteration. Figure 4.6.1 shows a window of size 45x77 pixels with a binary segmentation of a hand achieved using the Gaussian probabilistic threshold described earlier. The figure also shows the horizontal and vertical histograms of the image. If the earlier assumption about the location of an object within the window holds true, then it can be assumed that these histograms will be approximately Gaussian, with their peaks at the centre of gravity previously calculated. By making this Gaussian assumption, the standard deviation in both x and y can be calculated and the bounds of the window for the next iteration estimated. Figure 4.6.1 also shows this fitted Gaussian curve superimposed upon both the x and y histograms. The Gaussian curve is estimated by calculating the standard deviation of the histogram in both x and y. Once done it is known that one standard deviation from the mean (σ) represents 34.1% of the information, 2σ represent 47.7% of the information and 3σ represents 49.9% (See Chebyshev's theorem, Section 3.2). It is therefore known that $\pm 2\sigma$ from the mean encompasses 95.4%. This simple calculation can be used to resize the window ensuring that over 95% of the information is encompassed by the ROI. In the Figure 4.6.1 the window is resized to $\pm 2.2\sigma$ where,

$$new\ size_x' = 4.4\sigma = 4.4 \sqrt{\frac{1}{\sum_x \sum_y I_{x,y}} \sum_{x=-s_x/2}^{s_x/2} \sum_{y=-s_y/2}^{s_y/2} I_{x,y} (x - CG_x)^2}$$

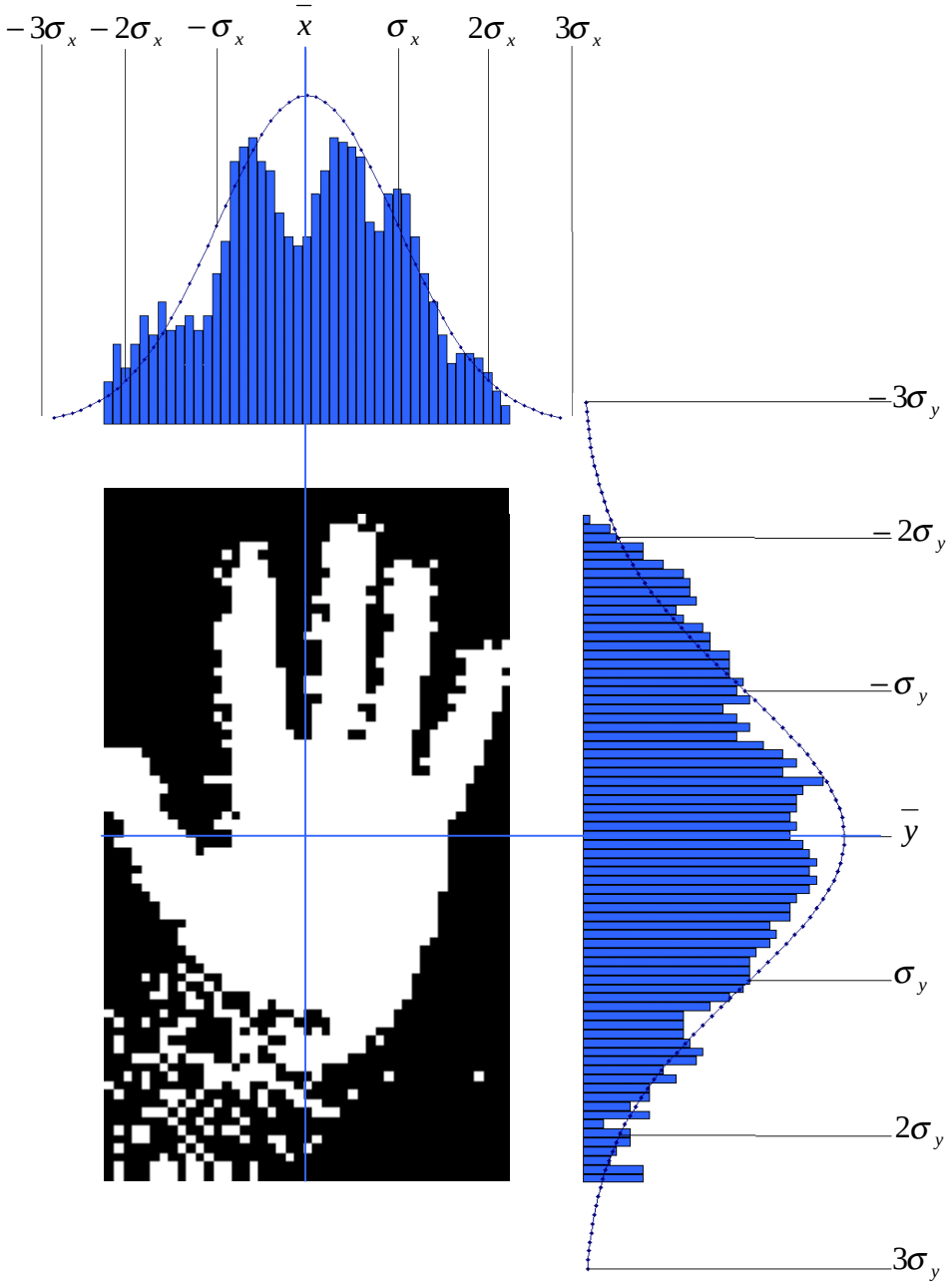


Figure 4.6.1 – Approximating the bounds on an object using a Gaussian Assumption

This simple procedure is iterated for each new image frame of a real-time image sequence. It relies upon a good initial location of the window. However, this can be achieved by performing the full image segmentation as described in section 4.5.

An Algorithmic overview is:

1. Construct PDF for colour thresholding model
2. Assign probability to each colour pixel from PDF
3. If probability is greater than some threshold mark pixel as TRUE else FALSE
4. Search image for largest blob
5. Calculate centre of blob and initialise window to this position
6. Calculate the approximate size of the blob and use to initialise window size

   $s_x = s_y = 2\sqrt{\frac{blob_{area}}{\pi}}$
7. While window size is greater than some threshold,
   8. Capture new image
   9. Segment window using PDF and threshold
   10. Calculate mean white pixel in x and y
   11. Move window to x,y
   12. Calculate the standard deviation in x and y, σ_x, σ_y
   13. Resize window to $2.2\sigma_x, 2.2\sigma_y$
14. Return to 1

If the object is much larger than the window, then the Gaussian that is fitted will be far larger and hence the window will grow in size until equilibrium is achieved. Conversely, if the window is too large, the resulting Gaussian will be far smaller than the window and hence the window will reduce in size until equilibrium has been achieved. This approach allows colour objects to be segmented and tracked quickly as the minimum amount of processing is necessary on each frame.

Figure 4.6.2 (a) and (b) shows the progress of applying this active sampling window to a live image sequence. As the hand is moved and rotated in the image frame, the window dynamically recalculates its parameters to retain the hand within its ROI. Figure 4.6.2 (c) shows the same procedure applied to the head with no change in parameters. Although the model is trained upon a single human, it has proved a generic skin tracker for all subjects regardless of skin type and without the need for relearning the colour space of skin. If however, the

lighting is changed, this requires that a new skin model be learnt due to the large variations in frequency for different sources of light (i.e. fluorescent tube or daylight). This provides a generic tracking approach for applications with consistent illumination.

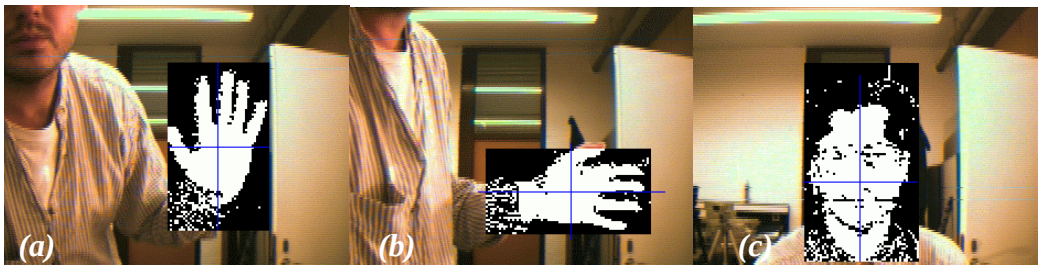


Figure 4.6.2 – Tracking head and hand in the image frame using colour

4.7 Conclusion

This chapter has demonstrated how colour can be used without high computational cost to enhance vision algorithms. Several colour spaces have been discussed and the benefits of 'perceptual' colour spaces demonstrated. It has been shown that object colour is a powerful feature capable of facilitating the robust tracking of objects in its own right. It has also been shown that with simple techniques, colour features can provide a fast, robust approach to tracking any generic colour object.

Throughout the remainder of this work, many of the simple techniques presented here will be used to enhance techniques in general. Chapter 10 will actively use the colour tracker approach presented in Section 4.6 but throughout the remainder of this work the use of colour in PDM tracking and boundary segmentation is implicit.