1 Colour Image Processing

1.1 Colour Fundamentals

Colour image processing is divided into two main areas: full colour and pseudo-colour processing. In the former group, the images are normally acquired with a full colour sensor such as a CCTV camera. In the second group, a colour is assigned to a specific monochrome intensity or combination of intensities.

People perceive colours that actually correspond to the nature of the light reflected from the object. The electromagnetic spectrum of the chromatic light falls in the range of 400–700 nm. There are three quantities that are used to describe the quality of a chromatic light source: radiance, luminance and brightness.

- Radiance: The total amount of energy that flows from the light source (units: watts);
- Luminance: The amount of energy an observer can perceive from the light source (lumens);
- Brightness: The achromatic notion of image intensity.

To distinguish between two different colours, there are three essential parameters, i.e. brightness, hue and saturation. Hue represents the dominant colour and is mainly associated with the dominant wavelength in a range of light waves. Saturation indicates the degree of white light mixed with a hue. For example, pink and lavender are relatively less saturated than the pure colours e.g. red and green.

A colour can be divided into brightness and chromaticity, where the latter consists of hue and saturation. One of the methods to specify the colours is to use the CIE chromaticity diagram. This diagram shows colour composition that is the function of x (red) and y (green). **Figure 1** shows the diagram, where the boundary of the chromaticity diagram is fully saturated, while the points away from the boundary become less saturated. **Figure 1** illustrates the colour gamut.

The chromaticity diagram is used to demonstrate the mixed colours where a straight line segment connecting two points in the chart defines different colour variations. If there is more blue light than red light, the point indicating the new colour will be on the line segment but closer to the blue side than the green side. Another representation of colours is to use the colour gamut, where the triangle outlines a range of commonly used colours in TV sets and the irregular region inside the triangle reflects the results of the other devices.

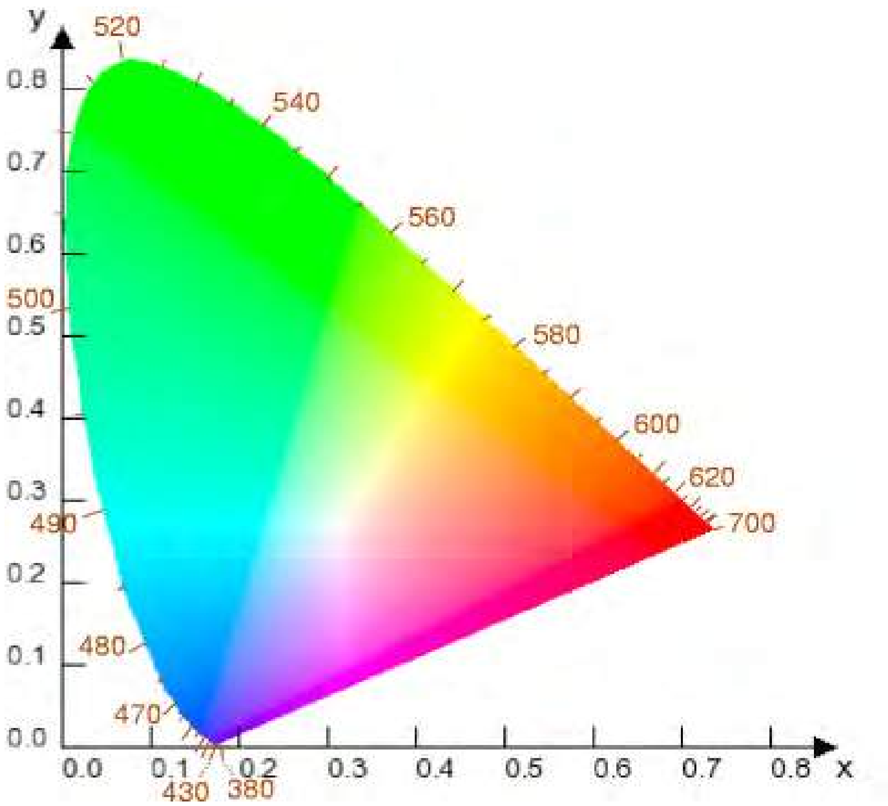


Figure 1 Illustration of the CIE chromaticity diagram ([8]).

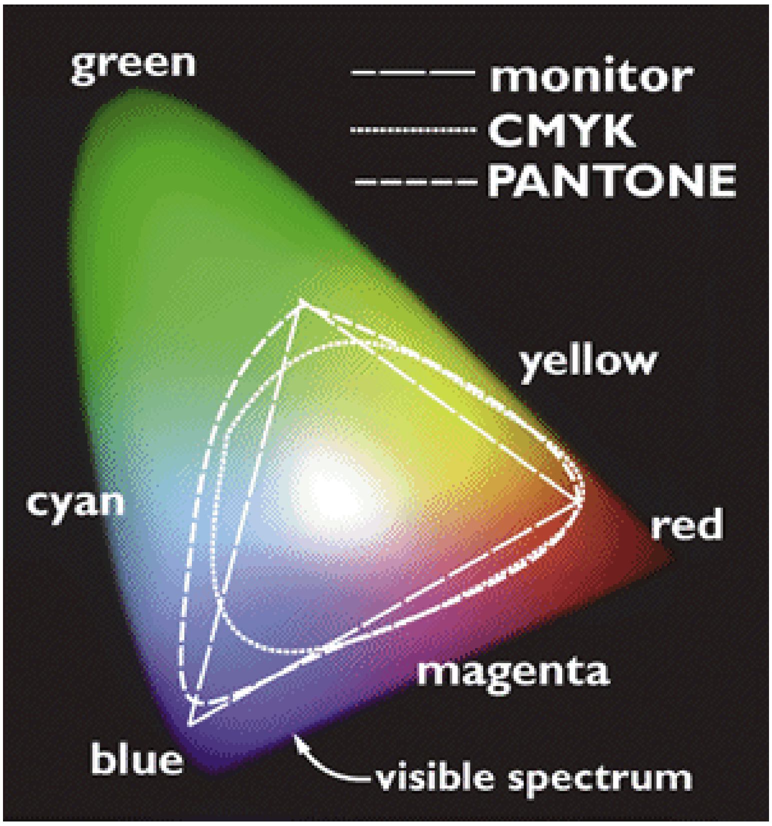


Figure 2 Illustration of the colour gamut ([9]).

1.2 Colour Space

Colour space or coulour model refers to a coordinate system where each colour stands for a point. The often used colour models consist of the RGB (red, green abd blue) model, CMY (cyan, magentia and yellow) model, CMYK (cyan, magenta, yellow and black) model and HIS (hue, saturation and intensity) model.

RGB model: Images consist of three components. These three components are combined together to produce composite colourful images. Each image pixel is formed by a number of bits. The number of these bits is namely pixel depth. A full colour image is normally 24 bits, and therefore the totoal number of the colours in a 24-bit RGB image is 16,777,216. **Figure 3** illustrates the 24-bit RGB colour cube that describes such a colour cube.

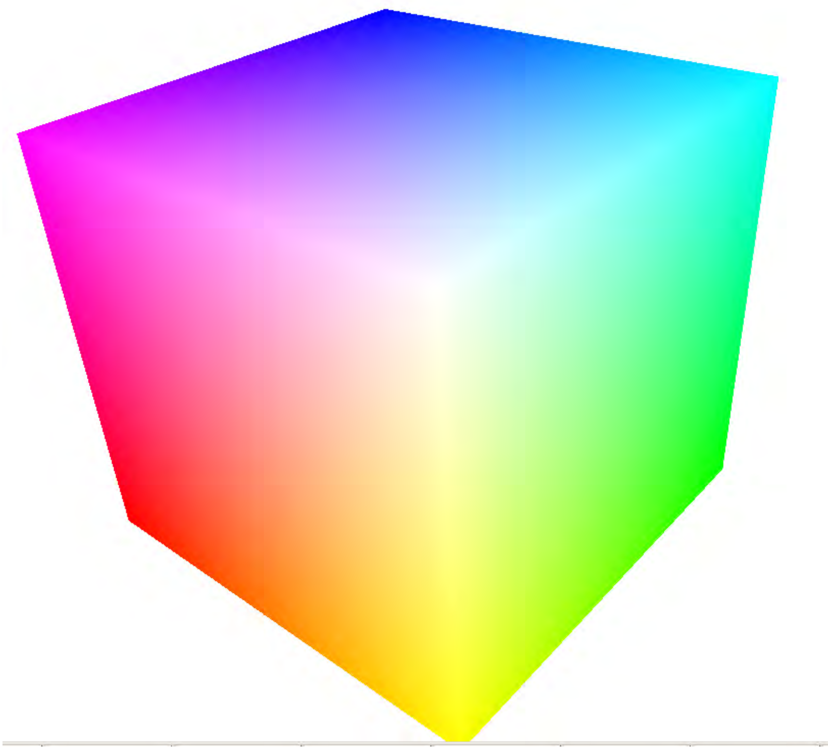


Figure 3 A colour cube ([10]).

CMY/CMYK colour models: These models contain cyan, magenta and yellow components, and can be formed from RGB using the following equation:

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (1.2.1)$$

HSI colour models: These models work as follows:

$$H = \begin{cases} \theta \\ 360 - \theta \end{cases} \tag{1.2.2}$$

Where the upper case is the result of $B \leq G$, and the lower case results from $B \geq G$. In the meantime,

$$\theta = \cos^{-1}\left\{\frac{0.5[(R-G)+(R-B)]}{[(R-G)^2+(R-B)(G-B)]^{1/2}}\right\} \tag{1.2.3}$$

The saturation is

$$S = 1 - \frac{3}{R+G+B}[\min(R,G,B)] \tag{1.2.4}$$

The intensity is given by

$$I = 1/3(R+G+B) \tag{1.2.5}$$

Figure 4 shows the separation of hue, sauration and intensity of a color image.

(a)

(b)

(c)

Figure 4 Illustration of Hue (a), Saturation (b) and Intensity (c) of a colour image.

1.3 Colour Image Processing

Colour image processing consists of pseudo- and full-colour image processing. Pseudo-colour image processing is about the assignment of colours to gray levels according to certain evidence. To do so, one of the options is to use a technique called intensity slicing. This is a simple but effective approach. In an image domain of intensity and spatial coordinates, the intensity amplitudes are used to assign the corresponding colours: The pixels with gray levels larger than the pre-defined threshold will be assigned to one colour, and the remainder will be assigned to another colour. One of the examples using the intensity slicing technique is shown in **Figure 5**, where 10 colours have been assigned to the various slices.

(a)

(b)

Figure 5 Illustration of intensity slicing and colour assignment.

Full-colour image processing is more complex than the pseudo-colour case due to the three colour vectors. First of all, one basic manipulation of colour images is namely colour transformation. For example, RGB is changed to HSI and vice versa.

If a colour transformation can be expressed as follows:

$$\chi_i = T_i(\tau_1, \tau_2, ..., \tau_n) \tag{1.3.1}$$

where $i = 1, 2, \ldots, n$, χ is target colour image, τ is the original colour image and T is the transformation function. In a very simple case, the three components in the RGB colour space can be

$$\chi_i = k\tau_i \tag{1.3.2}$$

where $i = 1, 2, 3$ and k is a constant. Similarly, the CMY space has the following linear transformation:

$$\chi_i = k\tau_i + (1-k) \tag{1.3.3}$$

Figure 6 demonstrates the colour transformation using three common techniques.

(a)

Hue Saturation Intensity

Figure 6 Examples of grouping colour components.

On the other hand, like intensity slicing, colour slicing is such a technique that

$$\chi_i = \begin{cases} 0.5, \\ \tau_i \end{cases} \tag{1.3.4}$$

where the former condition is $[|\tau j - aj|] > d/2$ (a colour cube with a width d).

Now the main attention is shifted to histogram analysis which has played a key role in image transformation. Particularly, histogram equalization is an example. To produce an image with an uniform histogram of colour values, one of the possible ways is to spread the colour intensities uniformly while leaving the colour values unvaried. See the outcome of the histogram equalization, shown in **Figure 7**.

Figure 7 Colour histogram equalisation.

1.4 Smoothing and sharpening

Smoothing and sharpening are two basic manipulation tools on colour images. They are two reverse processes, where the latter is a procedure of reproducing image intensities by adding more details and the former refers to an averaging process within a window.

The smoothing process can lead to the mean colour intensity as follows:

$$\bar{I}(x,y) = \begin{bmatrix} \frac{1}{\lambda} \sum_{(x,y)\in w} R(x,y) \\ \frac{1}{\lambda} \sum_{(x,y)\in w} G(x,y) \\ \frac{1}{\lambda} \sum_{(x,y)\in w} B(x,y) \end{bmatrix} \quad (1.4.1)$$

This smoothing can be illustrated in **Figure 8**, where RGB images of the original image are shown accompanying the mean and difference images. The strategy used in the averaging procedure is to apply a Gaussian mask (width = 3) to the original image.

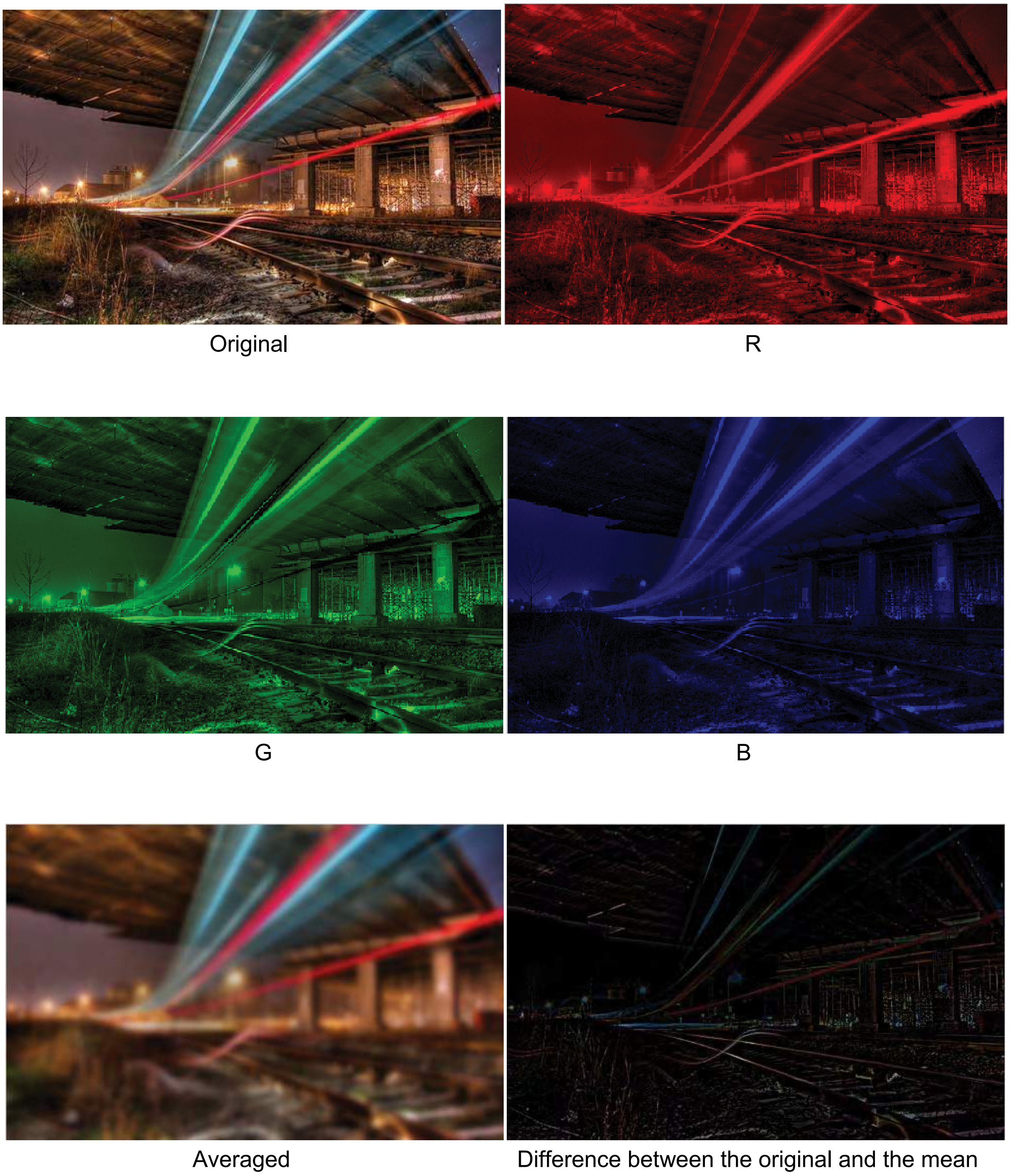
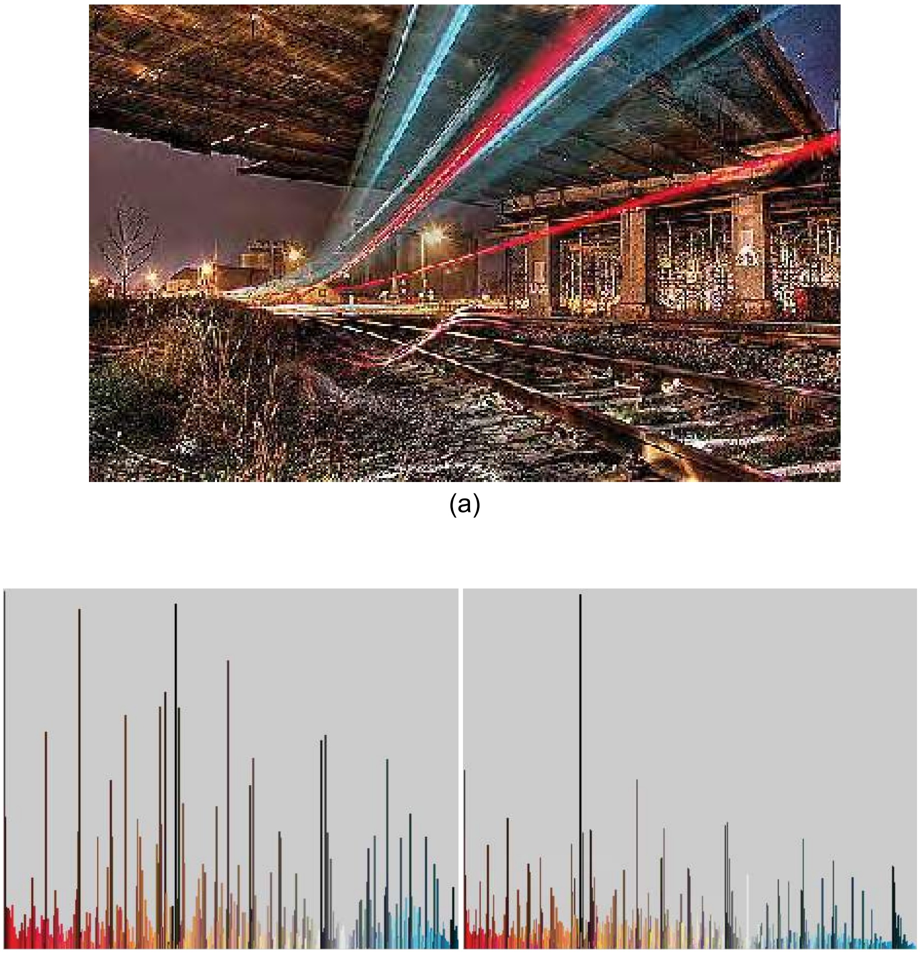


Figure 8 Image smoothing and the individual components.

A simple sharpening stage is provided as an example. This process involves the Laplacian transformation of an image. In a RGB domain, the sharpening outcome is:

$$\nabla^2[I(x,y)] = \begin{bmatrix} \nabla^2 R(x,y) \\ \nabla^2 G(x,y) \\ \nabla^2 B(x,y) \end{bmatrix} \tag{1.4.2}$$

Figure 9 illustrates the sharpened image and two colour distributions before and after the sharpening. It is observed that the sharpening process has changed the colour distribution of the intensities.



(a)

(b) (c)

Figure 9 Image sharpening and colour bars: (a) is the sharpened image, (b) and (c) are the histograms before and after the sharpening.

1.5 Image segmentation

In this subsection, image segmentation is mainly conducted based on the colour differentiation. It is a grouping process that enables image pixels to be separated according to their colour intensities. One of the segmentation schemes is hard thresholding (or namely binarisation), where a threshold is determined manually or empirically. For example, a colour image can be segmented according to its histogram of intensity values (**Figure 10**). However, this segmentation easily leads to mistaken grouping outcomes if the image pixels are cluttered. In addition, it mainly relies on the experience of a professional user. To reduce erroneous segmentations, soft thresholding techniques are hence developed. These approaches perform automatic and adaptive determination of the thresholds. In this section, only a couple of examples of the "soft" thresholding approaches will be presented, besides the classical neural networks, genetic and evolutionary algorithms.

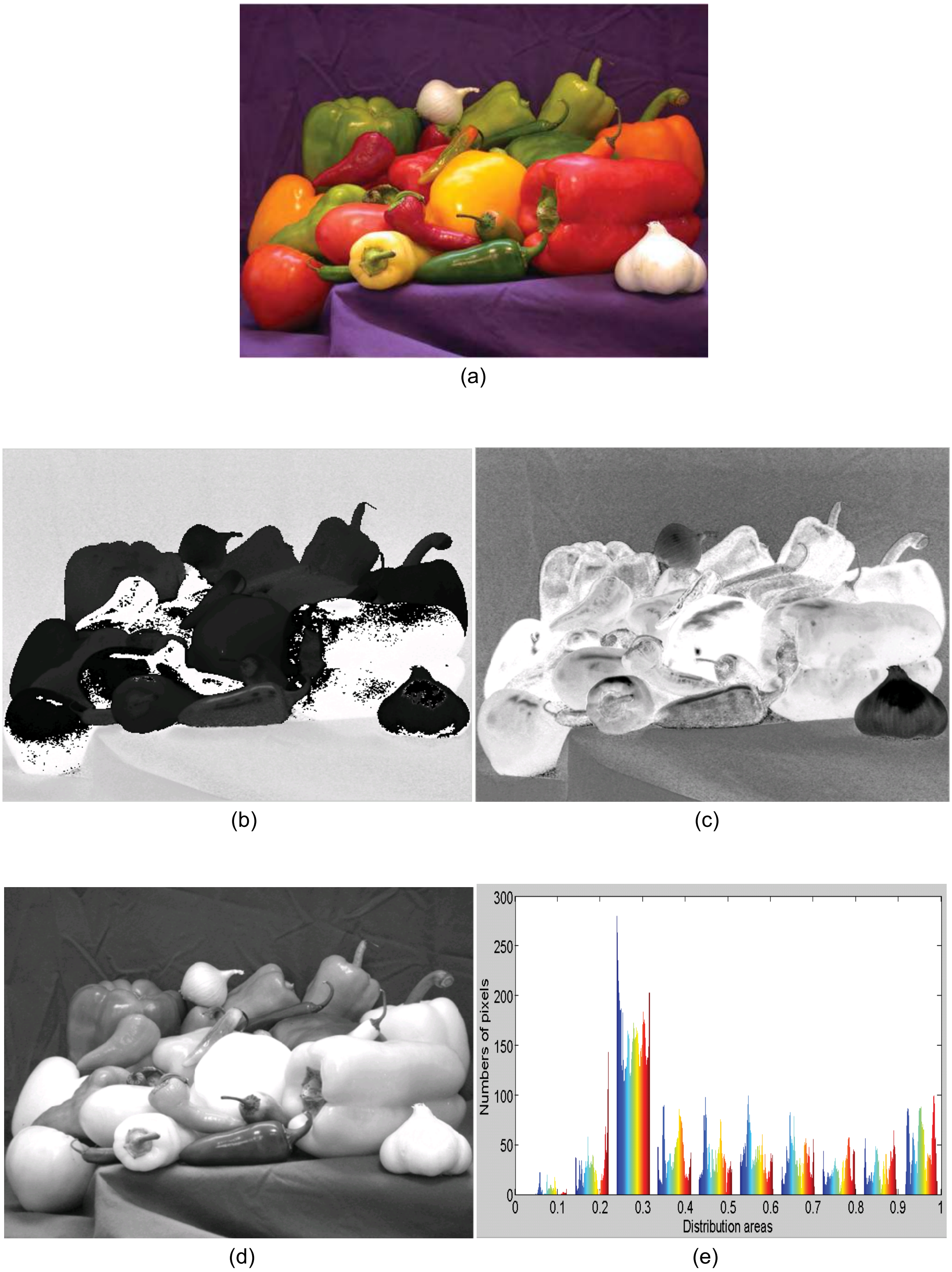


Figure 10 Illustration of a colour image and HSV decomposition: (a) original image, (b) hue, (c) saturation, (d) intensity value and (e) histogram.

K-means segmentation

K-means segmentation is a technique that aims to partition observations into a number of clusters where each observation belongs to the cluster with the nearest mean. The observations closer to a specific cluster will be assigned a higher weight and this helps remove the effects of some outliers. Suppose that there is a set of observations (x_1, x_2,…, x_n), where each observation can be a multi-dimensional vector. Therefore, these observations will be grouped into k sets $S = (S_1, S_2,\ldots, S_k)$ which must satisfy the following minimization of sum of squares [11]:

$$\arg\min_{S} \sum_{i=1}^{k} \sum_{x_j \in S_i} \| x_j - \nu_i |^2 \qquad (1.5.1)$$

where ν_i is the mean of S_j.

The standard algorithm to achieve this K-means segmentation is executed in an iterative style. Given an initial state of K means m_1^1, …, m_k^1, which can be obtained through empirical study or random guess, we then conduct the following steps. Then, the entire scheme operates as follows:

Initialization step: Each observation is assigned to the cluster with the closest mean.

$$S_i^t = \{x_j : \| x_j - m_i^t | \le | x_j - m_{i^+}^t | \;\; for\_all\_i^+ = 1,\ldots,k\} \qquad (1.5.2)$$

Update step: Calculate the new means to be the centroid of the observations in the cluster.

$$m_i^{t+1} = \frac{1}{|S_i^t|} \sum_{x_j \in S_i^t} x_j \tag{1.5.3}$$

These two steps will be iterated until a pre-defined threshold is met. This algorithm is illustrated in **Figure 10**.

As an extension and variant of K-means, fuzzy c-means recently has been well investigated. This algorithm works without a need to assign the initial locations of the cluster centres. Due to the limit of the pagination only its performance is demonstrated in this section (**Figure 10**).

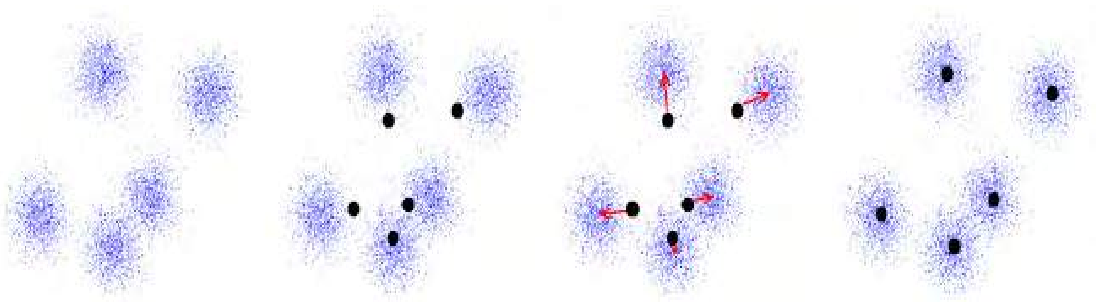


Figure 11 Illustration of K-means segmentation algorithm, where dots are the centres and red arrows refer to the moving direction.

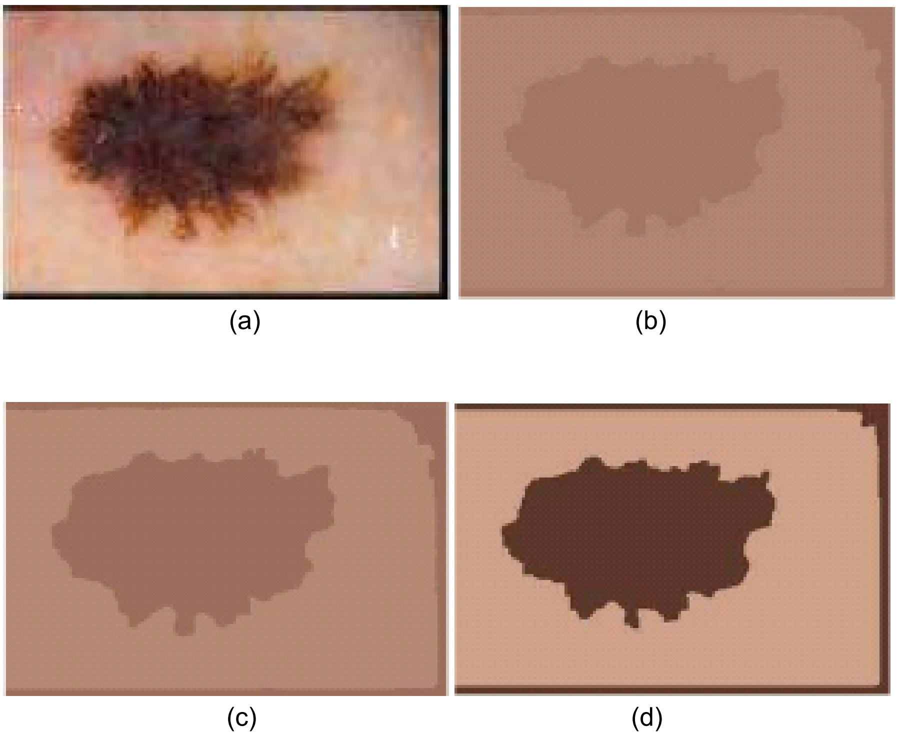


Figure 12 An evolving fuzzy C-means segmentation process.

Mean shift segmentation

Mena shift segmentation is a segmentation/clustering algorithm recently developed. There is no assumption made for the probability distributions. The aim of this algorithm is to find the local maxima of the probability density given by the observations. The algorithm of the mean shift segmentation is followed:

- Start from a random region;
- Determine a centroid of the estimates;
- Continuously move the region towards the location of the new centroid;
- Repeat the iteration until convergence.

Given a set of observations *x*, a kernel function *k* and a constant c_k, then the probability distribution function can be expressed as follows:

$$K(x) = c_k k(\| x |^2) \tag{1.5.4}$$

The kernel function can be an Epanechnikov kernel which has the form like this:

$$k(g) = \begin{cases} 1 - g \\ 0 \end{cases} \tag{1.5.5}$$

where $g = ||x||^2$. The upper case is true when g ≤ 1; otherwise the lower case stands. The kernel density of the estimated states of the data is described by the following equation:

$$\tilde{f}(x) = \frac{1}{h^d} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h} \right) \tag{1.5.6}$$

where *d* is the dimension of the data. When the algorithm reaches a maxima or minima in the iteration, this equation must be satisfied:

$$\nabla \tilde{f}(x) = 0 \tag{1.5.7}$$

Hence,

$$\nabla \tilde{f}(x) = \frac{2c_k}{h^{d+2}} \sum_{i=1}^{n} \hat{K}_i \left(\frac{\sum_{i=1}^{n} x_i \hat{K}_i}{\sum_{i=1}^{n} \hat{K}_i} - x \right) = 0 \tag{1.5.8}$$

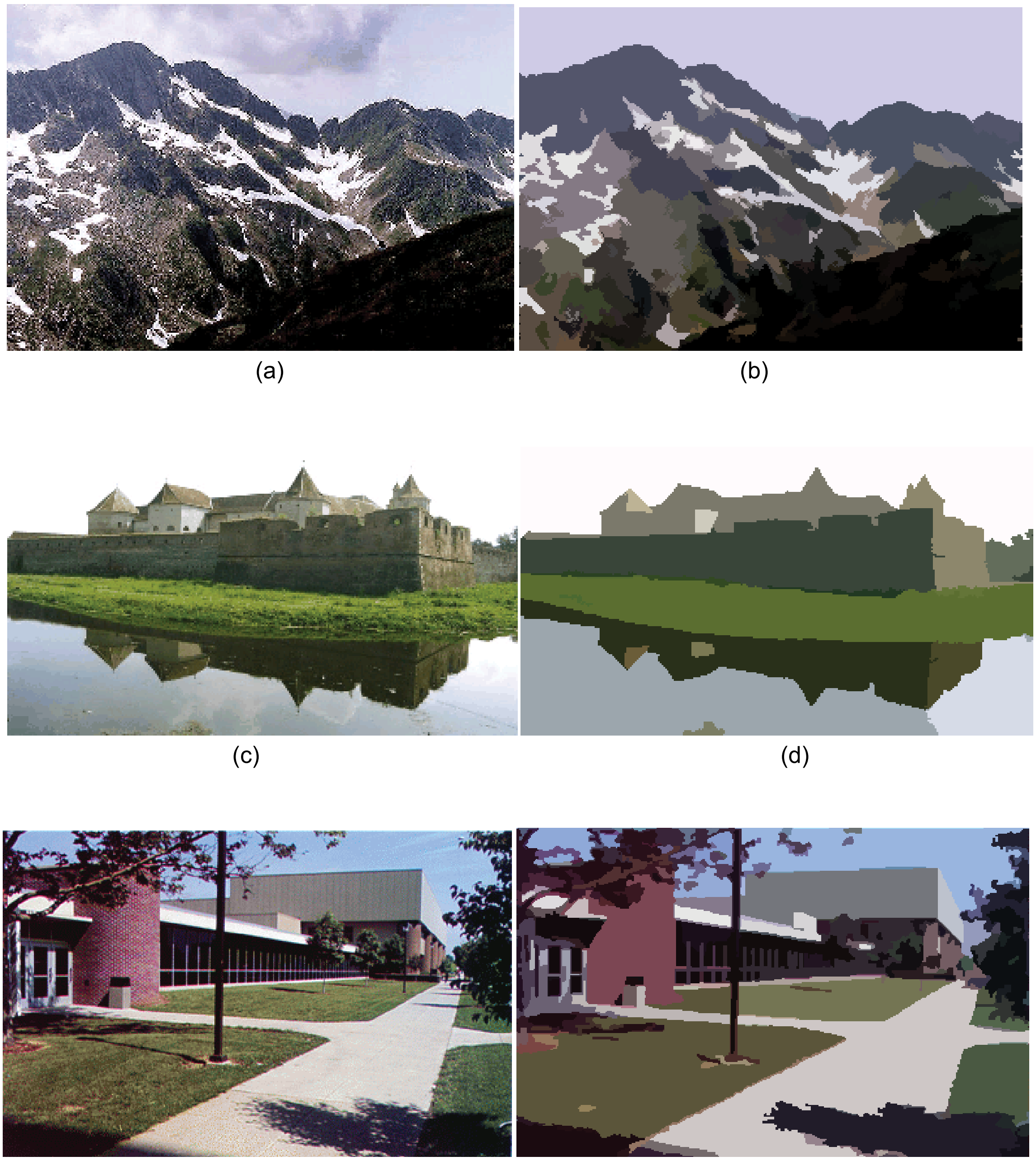
where the intermediate functions

$$\begin{cases} \hat{K}(g) = k'(g) \\ \hat{K}_i = K(\| x - x_i) / h |^2) \end{cases} \tag{1.5.9}$$

Finally, the mean shift vector is obtained in the computation loop:

$$m(x) = \frac{\sum_{i=1}^{n} x_i \hat{K}_i}{\sum_{i=1}^{n} \hat{K}_i} - x \tag{1.5.10}$$

To demonstrate the performance of the mean shift scheme, **Figure 13** shows some examples of mean shift segmentation. In general, the segmentation results reflect the embedded clusters in the images and therefore the mean shift algorithm works successfully.



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

Figure 13 Examples of mean shift segmentation (image courtesy of [12]).

1.6 Colour Image Compression

In this subsection, image compression is discussed. The reason why this issue is important to talk about is the fact that the number of bits of a colour image is three or four times greater than its counterpart in gray level style. Storage and transmission of this colour image takes tremendous time with a more complicated process, e.g. encoding and decoding. If this colour image can be reduced in terms of its bits, the relevant process will be much simplified.

A comprehensive introduction to the colour image compression is non-trivial and this will be detailed in a later study and other references. In this section, some recently developed techniques are briefly introduced. These techniques are mainly comprised of two types, “lossless” and “lossy” compression. Digital Video Interface (DVI), Joint Photographic Experts Group (JPEG) and Motion Pictures Experts (MPEG) are the widely used techniques. No doubt, the lossy techniques normally provide greater compression ratio than the lossless ones.

Lossless compression: These methods aim to retain lower compression ratios but preserve all the pixels in the original image. The bits of the resulting image are larger than the lossy compression. The common methods are Run-Length Encoding (RLE), Huffman encoding, and entropy coding. RLE checks the image stream and inserts a special token each time a chain of more than two equal input tokens is found. Huffman encoding assigns a longer code word to a less common element, while a weighted binary tree is built up according to their rate of occurrence. In the entropy coding approaches, if a sequence is repeated after a symbol is found, then only the symbol is part of the coded data and the sequence of tokens referred to the symbol can be decoded later on.

Lossy compression: These approaches retain higher compression rates but sacrifice with a less resolution in the final compressed image. JPEG is the best known lossy compression standard and widely used to compress still images. The concept behind JPEG is to segregate the information in the image by levels of their importance, and discard the less important information to reduce the overall quantity of data. Another commonly used coding scheme is namely "transform coding" that subdivides an N-by-N image into smaller n-by-n blocks and then performs an unitary transform on each block. The objectives of the transform are to de-correlate the original image, which results in the image energy being distributed over a small amount of transform coefficients. Typical schemes consist of discrete cosine transform, wavelet and Gabor transforms. **Figure 14** demonstrates the performance of a wavelet analysis in the image compression and reconstruction of the compressed image.

(a)

(b)

(c)

Figure 14 Colour image compression using wavelet analysis: (a) original, (b) compressed image and (c) reconstructed image.

The algorithm of the transform coding can be summarized as follows:

- Image subdivision
- Image transformation
- Coefficient quantization
- Huffman encoding

Another commonly used compression scheme is vector quantization. This is a transform from a higher dimensional Euclidean space to a finite subset. The subset can be the vector codebook. One of the best vector quantization algorithms is described as follows:

- Subdivide the training set into N groups, which are associated with the N codebook letters.
- The centroids of the partitioned regions become the updated codebook vectors.
- Compute the average distortion. If the percent reduction in the distortion is less than a pre-defined threshold, then stop.

In addition, segmented image coding and fractal coding schemes can be used to handle different circumstances. For example, segmented image coding considers images to be composed of slowly varying image intensity. These slowly moving regions will be identified and then used as the main structure of the encoded image.

1.7 Summary

In this chapter, the concepts of radiance, luminance and brightness have been introduced. The chromaticity diagram was used to illustrate the complexity of colours. In the colour space, RGB, CMY and HSI colour models have been summarised. Afterwards, intensity slicing and colour assignment were also introduced. To further improve the quality of a colour image colour equalisation was presented to generate uniformly distributed colour intensities.

Colour smoothing and sharpening are two important methods that can be used to enhance the quality of an image. One example of smoothing by using a Gaussian mask is denoted. The image sharpening is demonstrated using a Laplacian operator. In the following sections, image segmentation and compression have been respectively discussed. The former include two examples, k-means and mean shift. The latter has looseless and lossy compression techniques. In particular, the application of a wavelet analysis based compression is shown.

In general, image smoothing/sharpening, segmentation and compression are the key contents in this section. In spite of their brief introduction, these descriptions demonstrate the necessity of these algorithms in real life. In addition, it has been observed that further investigation for a better image quality must be taken into account. These issues will be addressed on the later sections.

1.8 References

[10] www.knowledgerush.com/kr/encyclopedia/Colour/, accessed on 30 September, 2009.

[11] http://dx.sheridan.com/advisor/cmyk_color.html, accessed on 30 September, 2009.

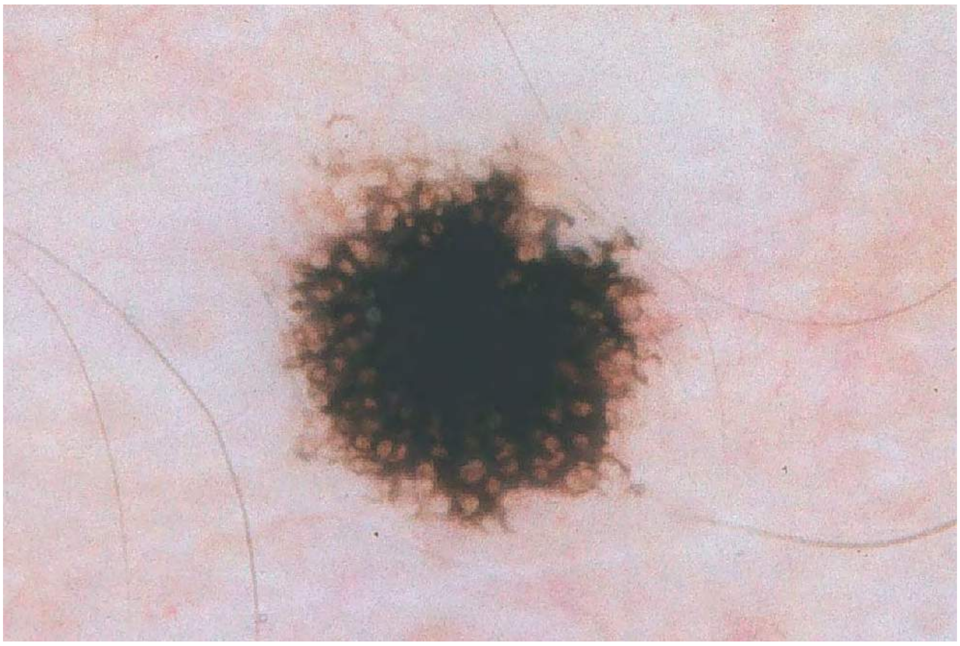
[12] http://luminous-landscape.com/forum/index.php?s=75b4ab4d497a1cc7cca77bfe2ade7d7d&showtopic=37695&st=0&p=311080&#entry311080, accessed on 30 September, 2009.

[13] http://en.wikipedia.org/wiki/K-means_clustering, accessed on 4 October, 2009.

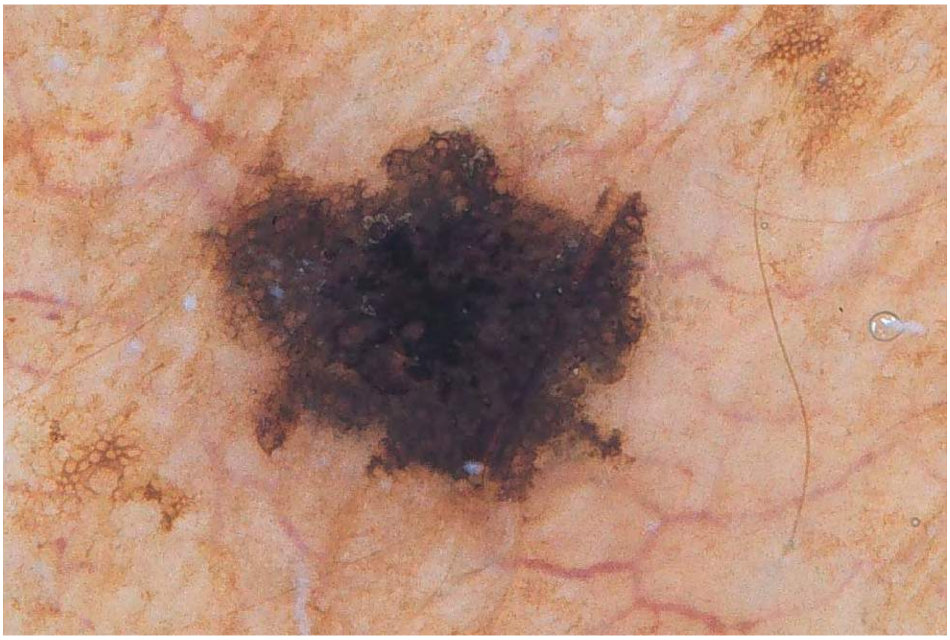
[14] D. Comaniciu, P. Meer: *Mean Shift: A Robust Approach toward Feature Space Analysis*, IEEE Trans. Pattern Analysis Machine Intell., Vol. 24, No. 5, 603–619, 2002.

Problems

1. What is a colour model?
2. What is image smoothing and sharpening? Try to apply Gaussian smoothing and edge sharpening respectively to following image:



3. How to perform image segmentation? Hints: One example can be used to explain the procedure.
4. Try to apply mean shift algorithms for image segmentation of the following image:



5. Is this a true statement? Image compression is a process of reducing image size.
6. Can you summarise the algorithm of RLE compression?