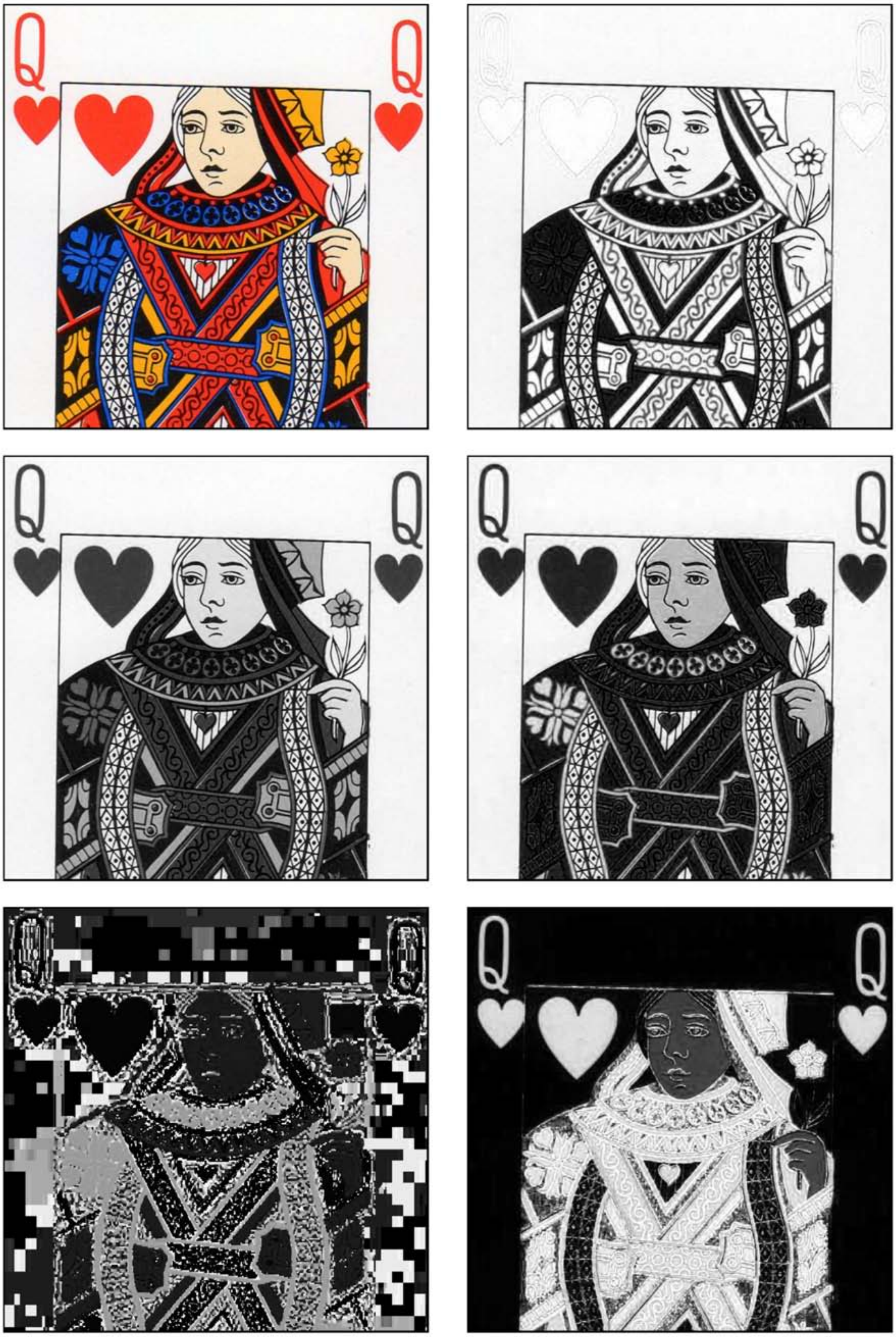
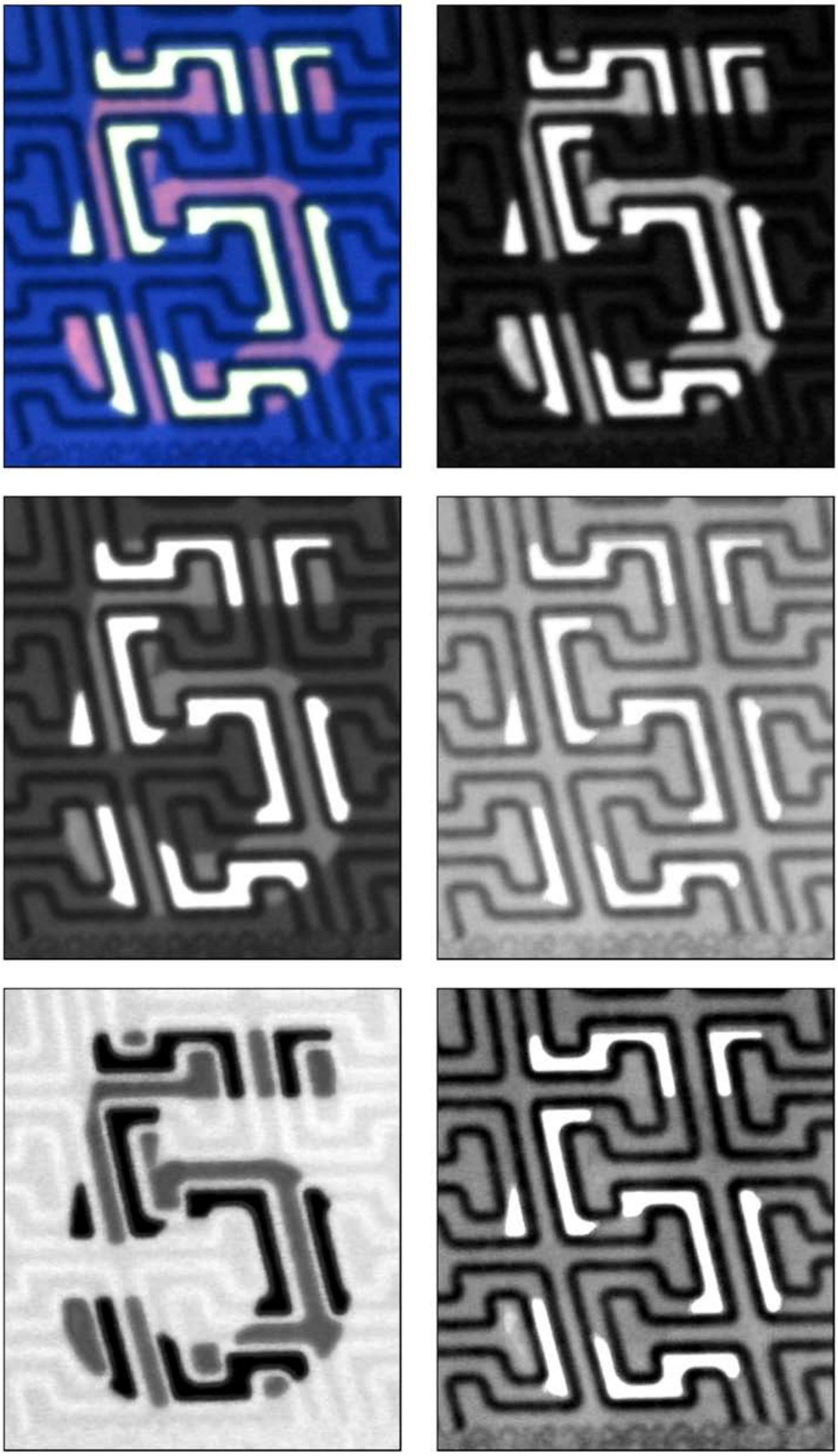
Situations in which intensity fails to provide satisfactory discrimination. (Top-left) Original image. [Child's ji-saw puzzle, printed card] (Top-right) Intensity. (Centre-left) Original image. [Thermo-chromic paint, used as thermometer] (Centre-right) Intensity. (Bottom-left) Original image [RFID] tag. (Bottom-right) Intensity.

Figure 6.1

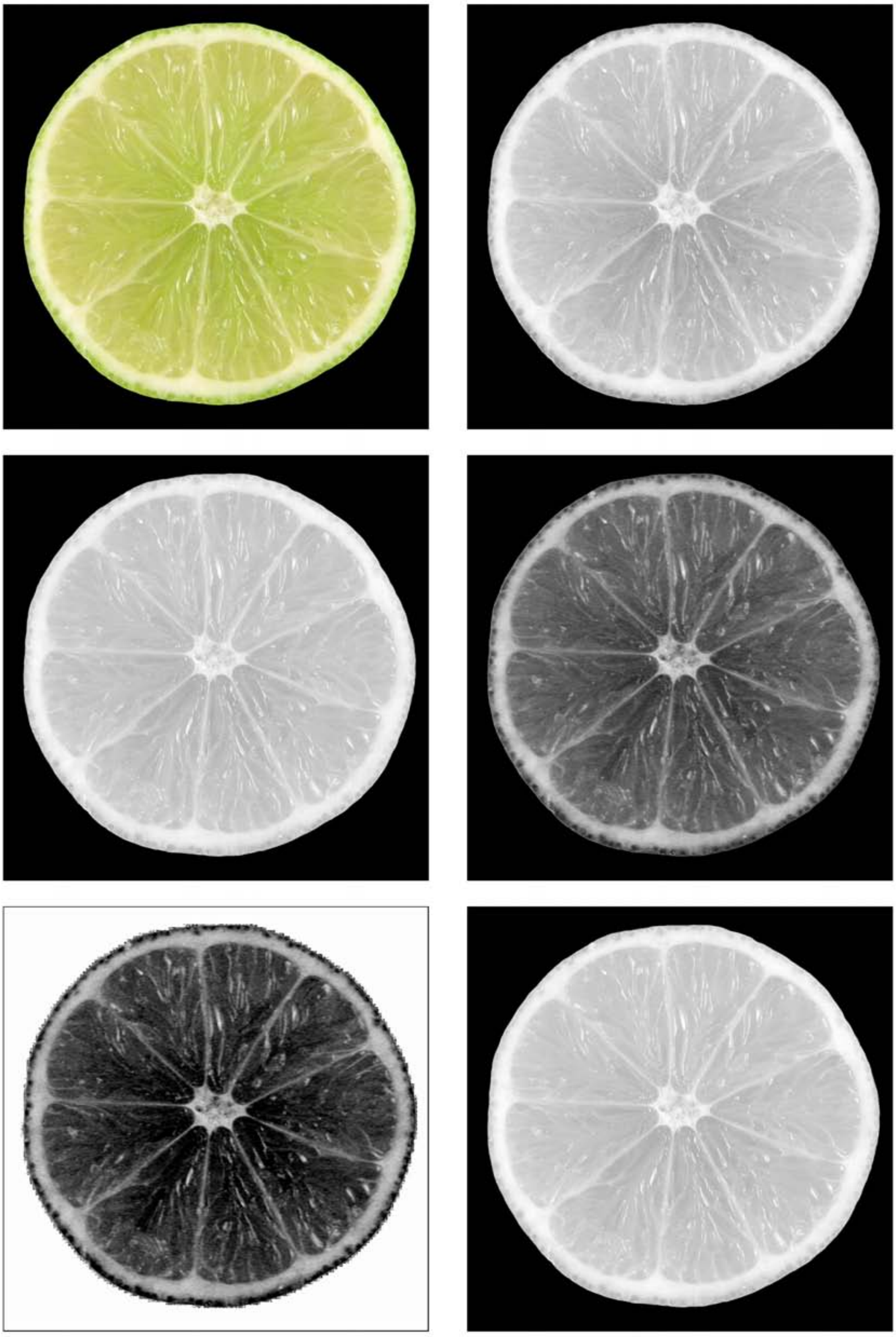


Components of a colour image. (Top-left) Original image. (Top-right) R-component. (Centre-left) G-image. (Centre-right) B-component. (Bottom-left) H-image. (Bottom-right) S-image.

Figure 6.2

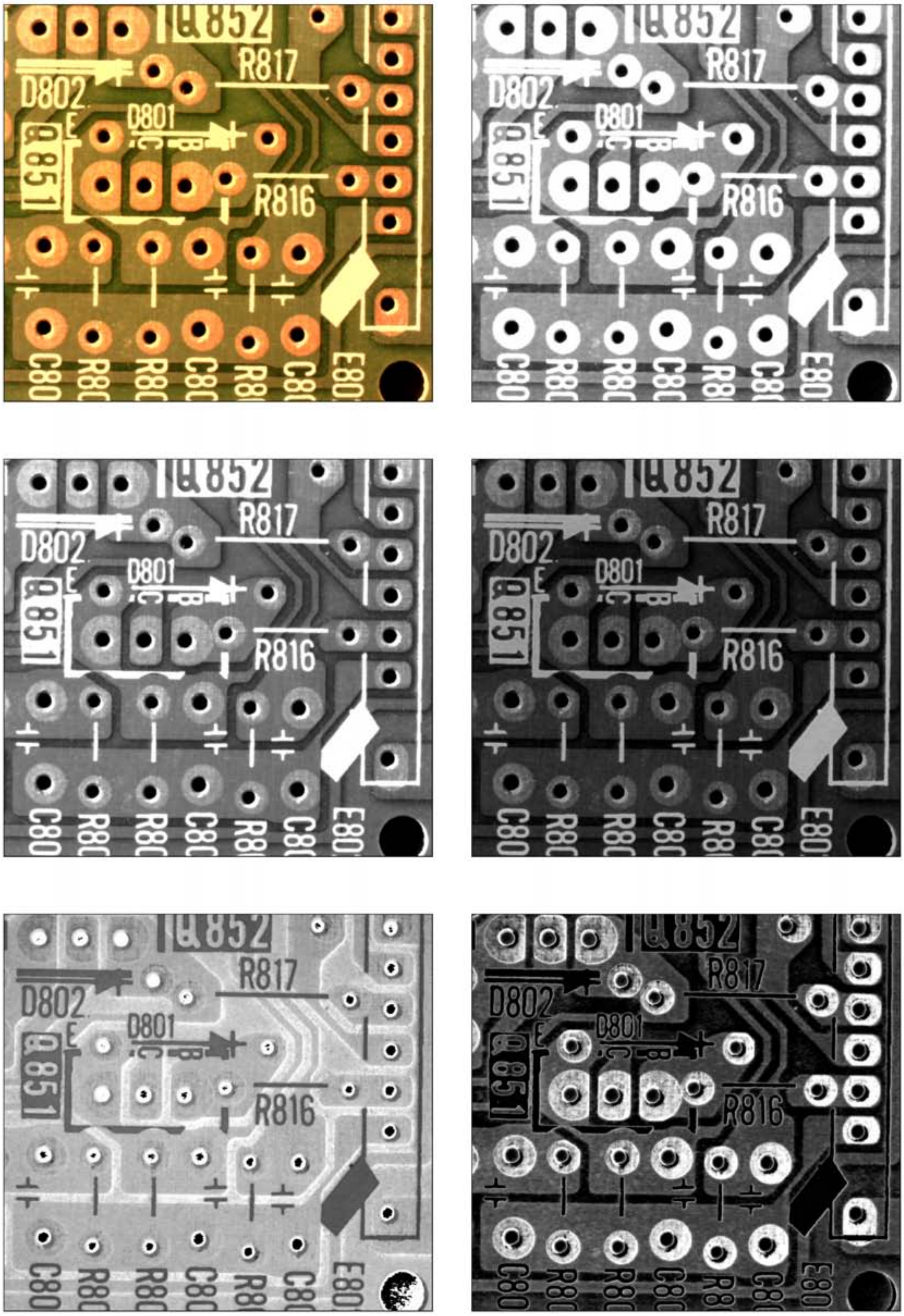


Identifying image features by colour separation. (Top-left) Original image (Flourescent pattern on a UK £5 note. UV lighting) (Top-right) R-image. (Centre-left) G-image. (Centre-right) B-image. (Bottom-left) S-image. (Bottom-right) I-image.



Separating the RGB and HSI components can sometimes improve contrast. (Top-left) Original image. (Top-right) R-image. (Centre-left) G-component. (Centre-right) B-image. (Bottom-left) S-image. (Bottom-right) I-image.

Figure 6.4



Selecting image features using combinations of the RGB components. (Top-left) Original image. (Top-right) R-image. (Centre-left) G-image. (Centre-right) B-image. (Bottom-left) S-image. Obscures the circular pads. (Bottom-right) The G-image subtracted from the R-image. Enhances the circular pads.

Figure 6.5

Adjusting the R-, G-, B-images separately. Details are given in the text.

Figure 6.6

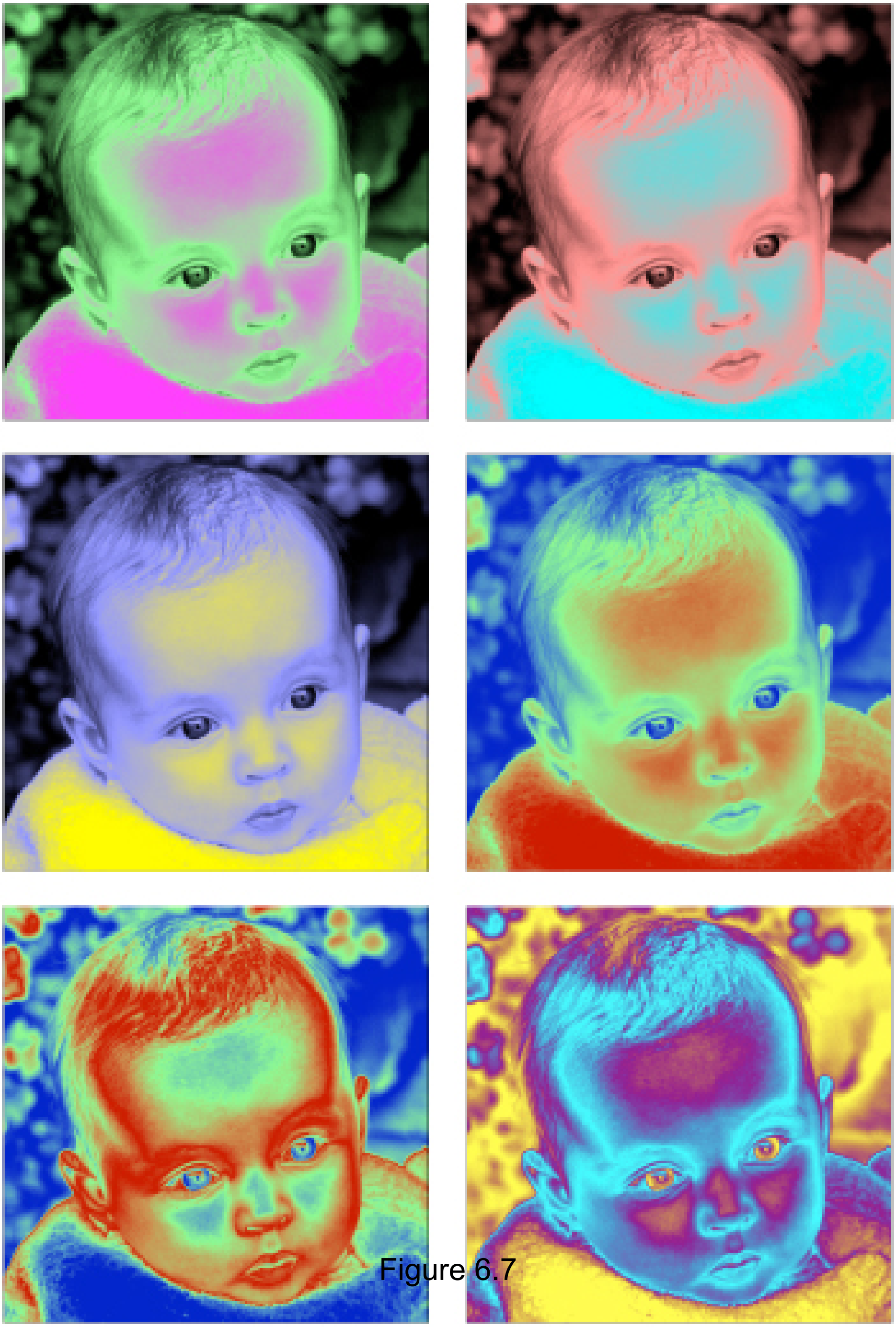
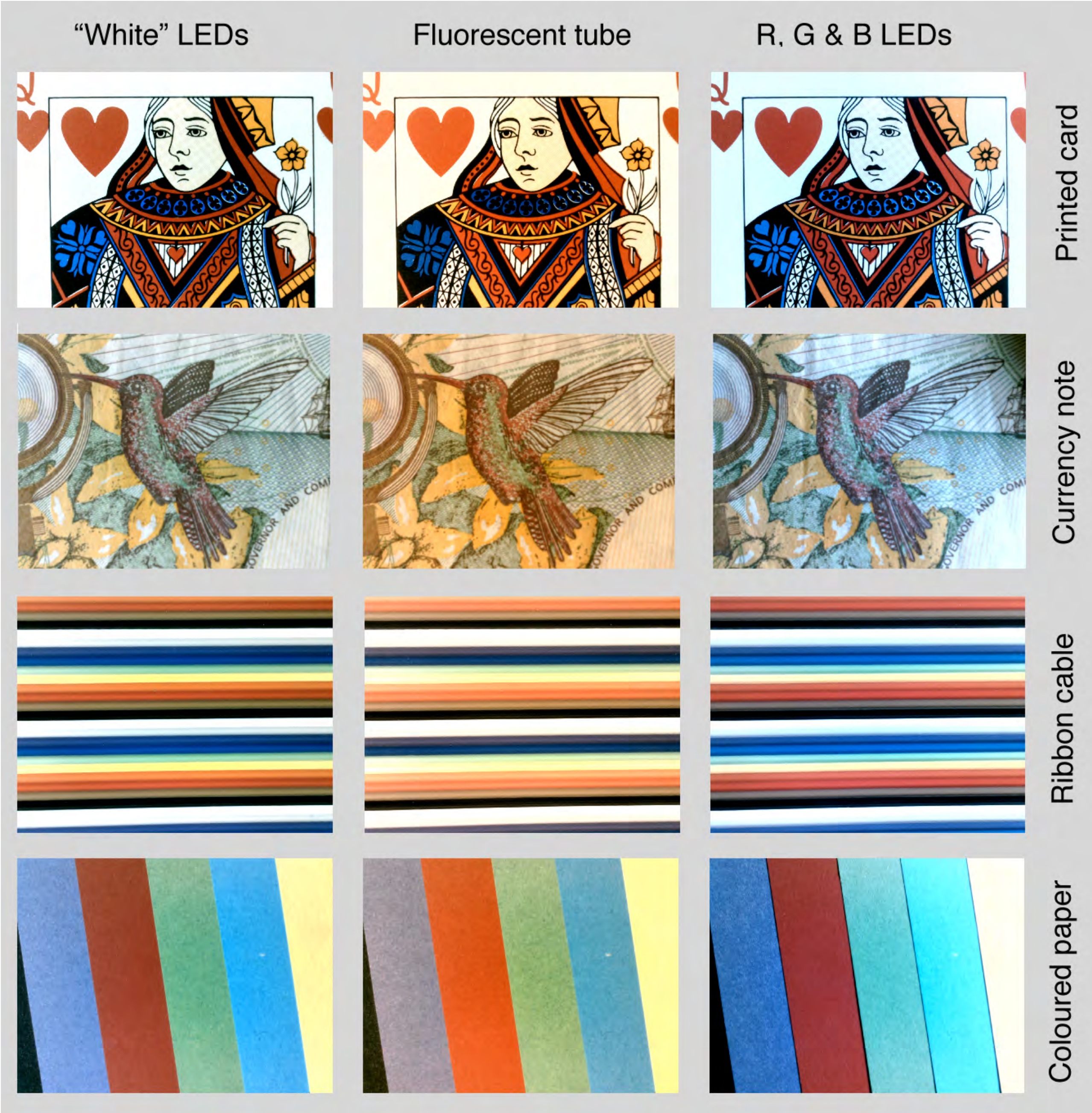
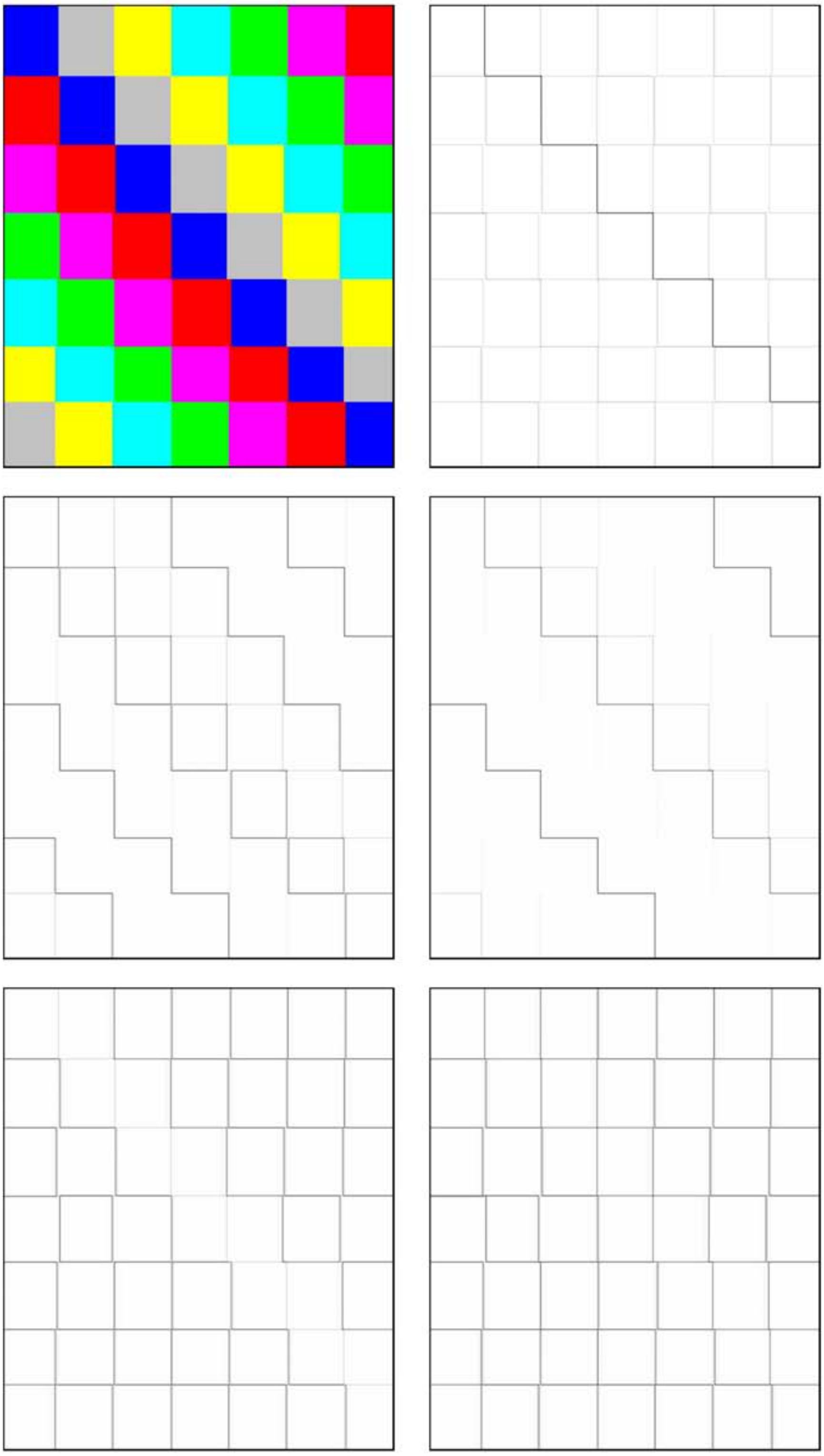


Figure 6.7



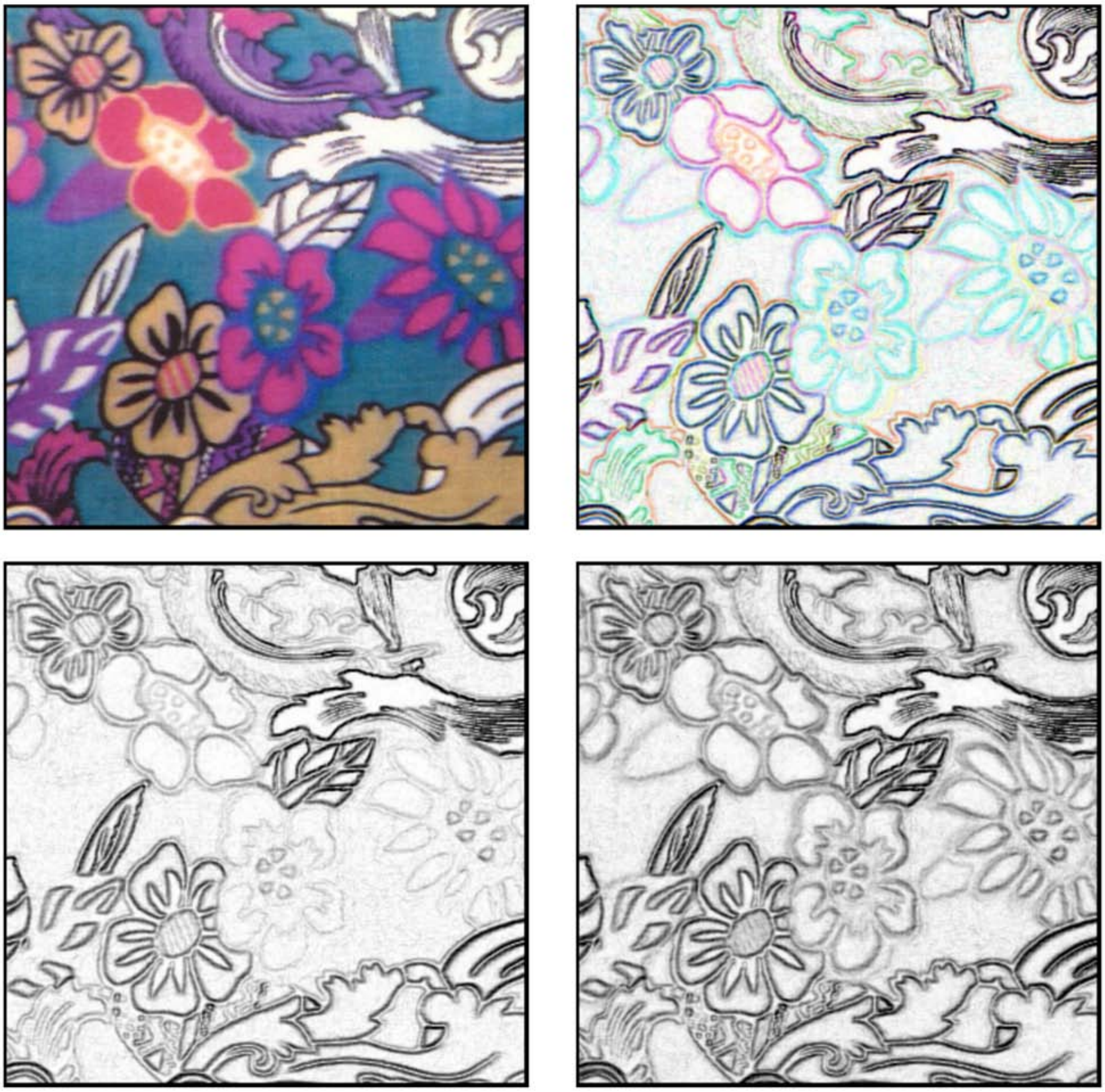
"White" lighting is not really white; people often refer to "harsh" (blue tint), or "soft" (red tint) lighting. To obtain these pictures, the camera's "White Balance" control was disabled. "*White LEDs"* emit short-wavelengths and illuminate a phosphor with a nearly-flat emission spectrum. *"R, G & B LEDs"* refers to tri-colour devices that emit light in three narrow spectral bands. By carefully mixing their outputs (red, green and blue light), the human visual system can be tricked into believing that white light is emitted. When illuminated with any of these sources, a person sees the paper strips as black, pale-blue, red, green, vivid-blue, yellow.

Figure 6.8



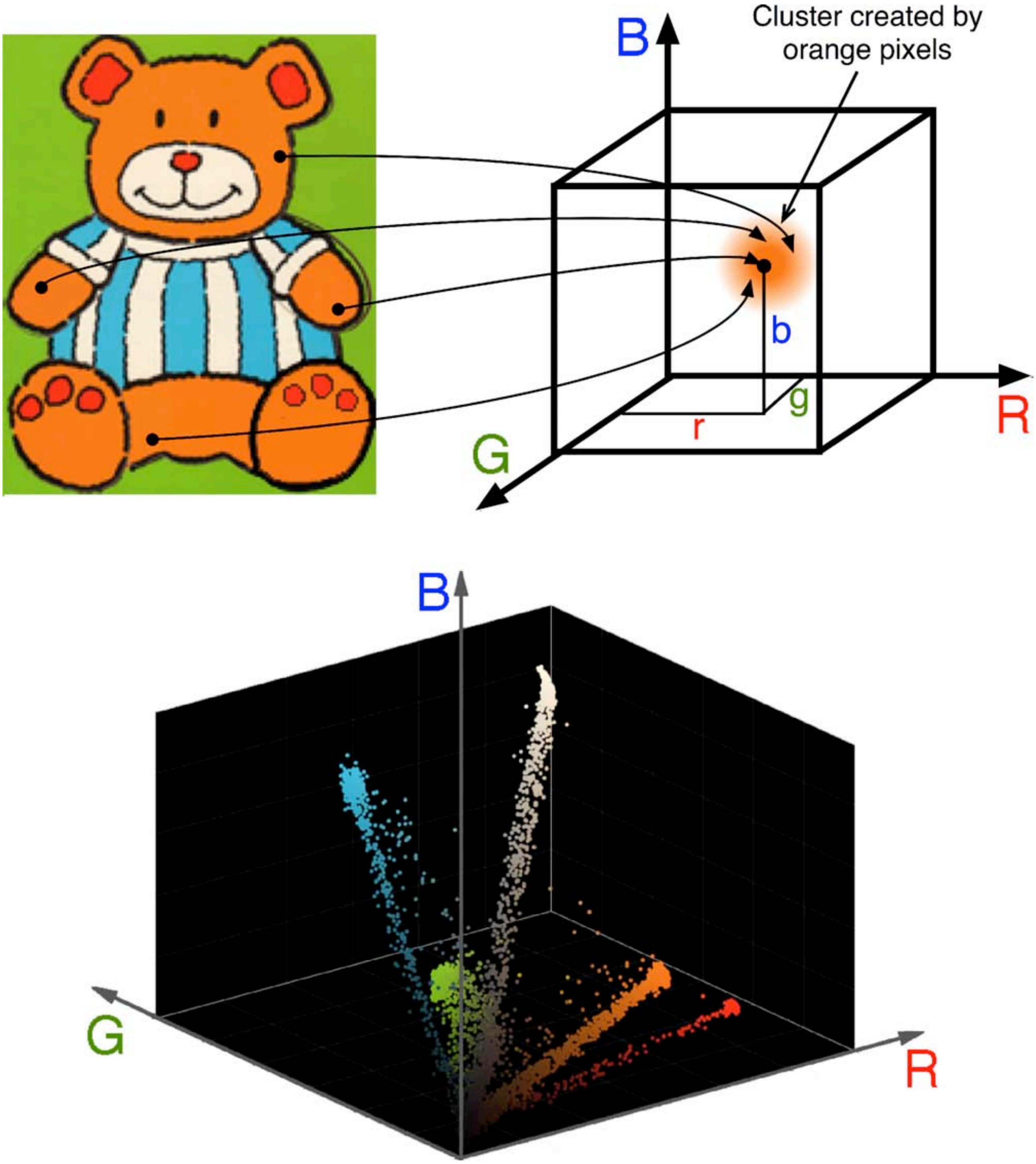
Edges in a colour image. (Top-left) Test image, created for this book. (Top-right) Edge detector applied to the I-image. Centre-left) Edge detector applied to the R-image. (Centre-right) Edge detector applied to the G-image. (Bottom-left) Edge detector applied to the B-image. (Bottom-right) Pixel-by-pixel minimum of [CL], [CR] and [BL].

Figure 6.9



Colour edges. (Top-left) Original image. (Dress fabric) (Top-right) Edge detector applied to the R-, G-, B-images separately. (Bottom-left) Applying an edge detector to the I-image. Notice that some edges are missing and others have very low contrast. (Bottom-right) Edges in the GB components merged using the pixel-by-pixel minimum function.

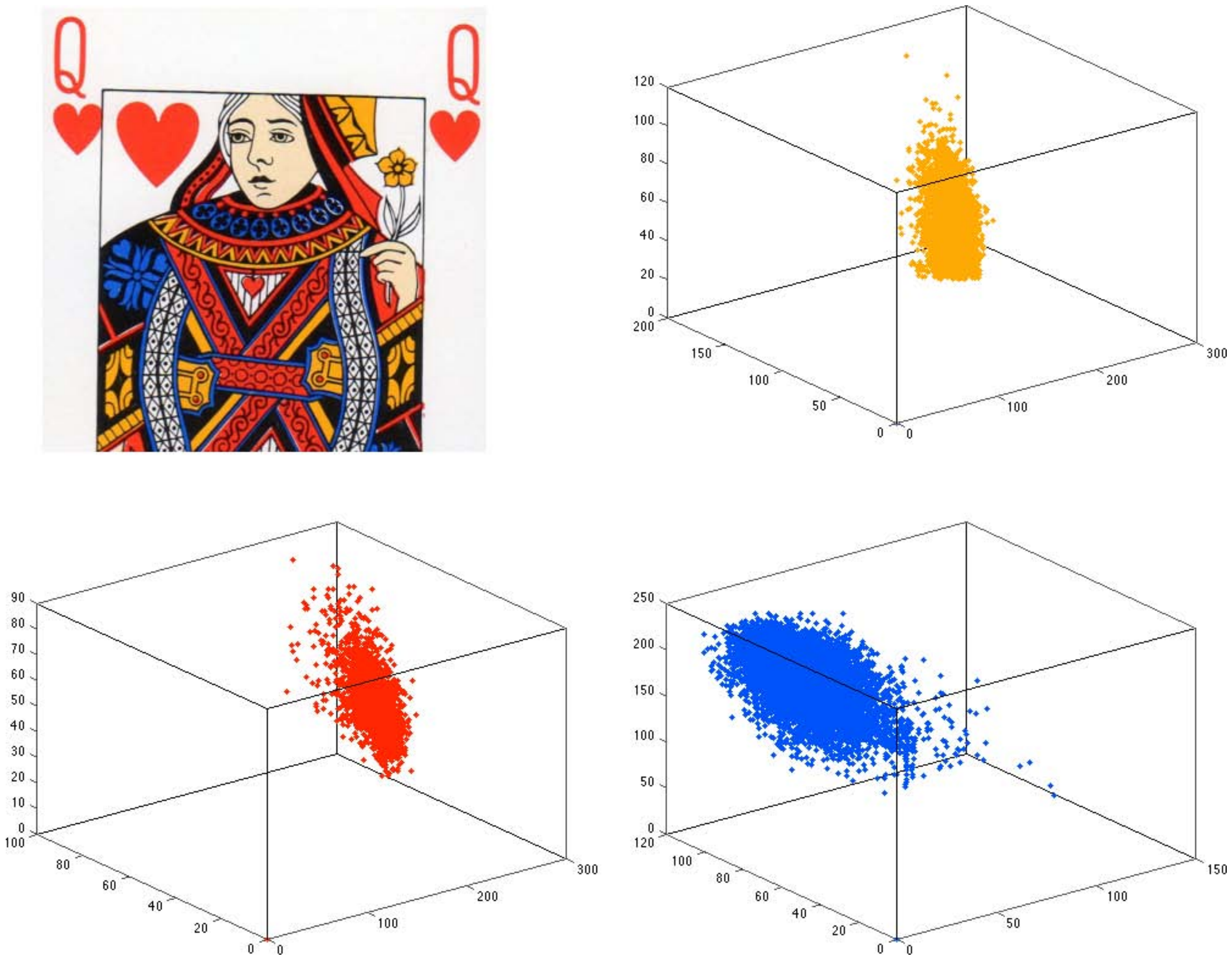
Figure 6.10



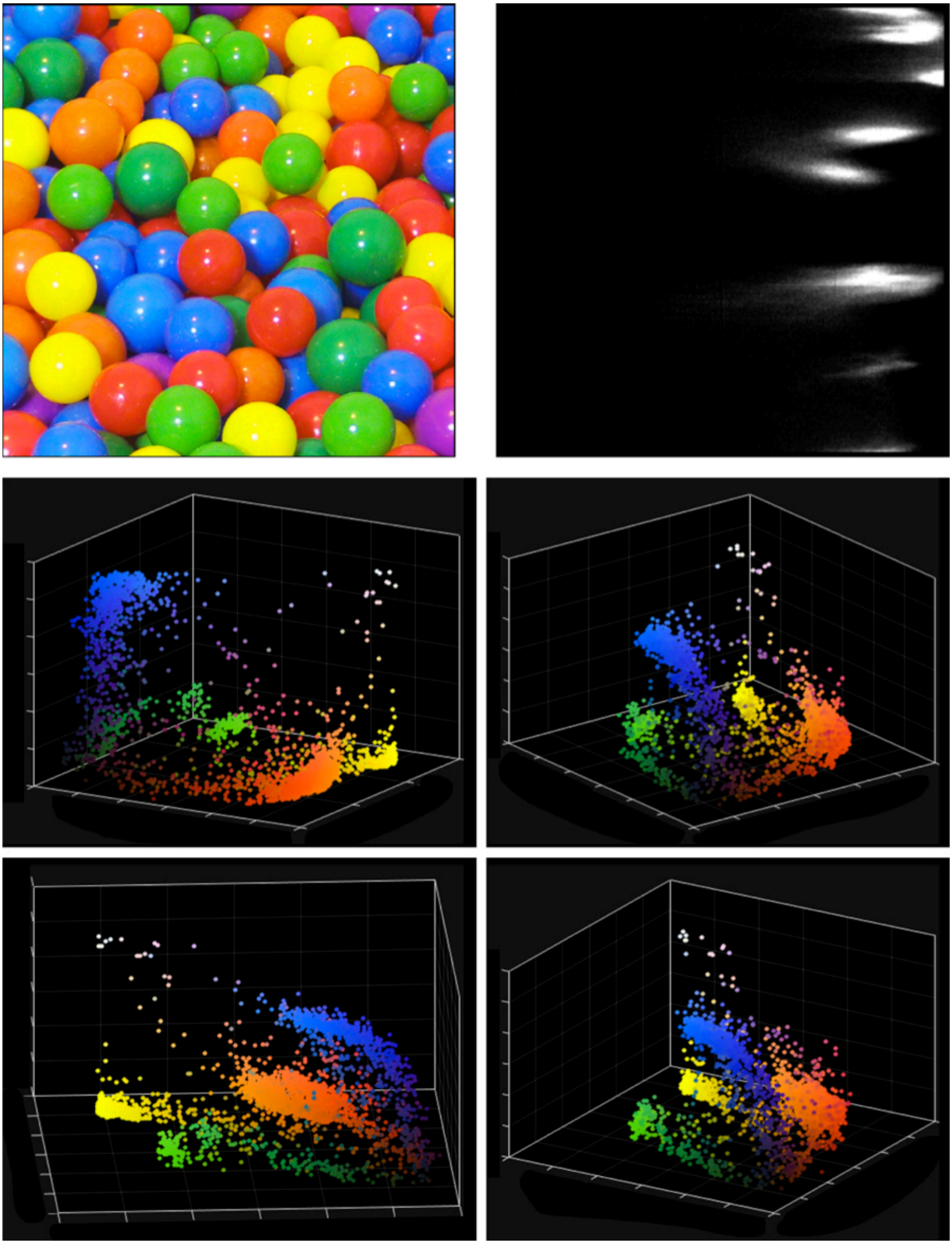
Generating the RGB scattergram. (Top) All orange pixels contribute to this cluster. Red, cyan, green and white clusters are also created In addition, a “black” cluster with a small population is also present but is not evident here.

(Bottom) RGB scattergram created from [TL]. The “comet tails” correspond to pixels that lie partially within blocks of colour and the black lines separating them.

Figure 6.11

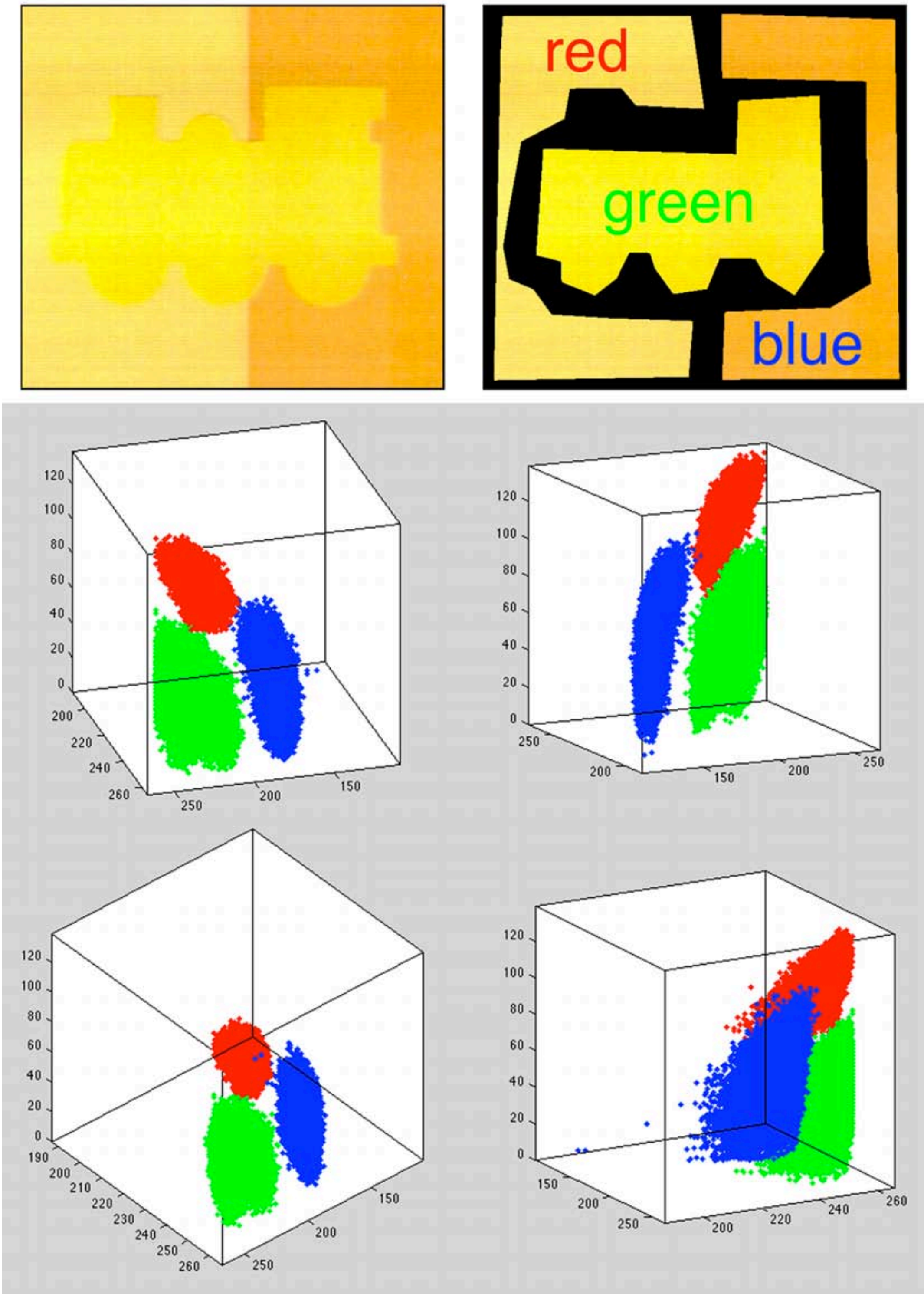


Clusters in RGB space
(Top-left) Original colour image.
(Others) Clusters in RGB space, plotted separately for the red, yellow and blue parts of the original image.

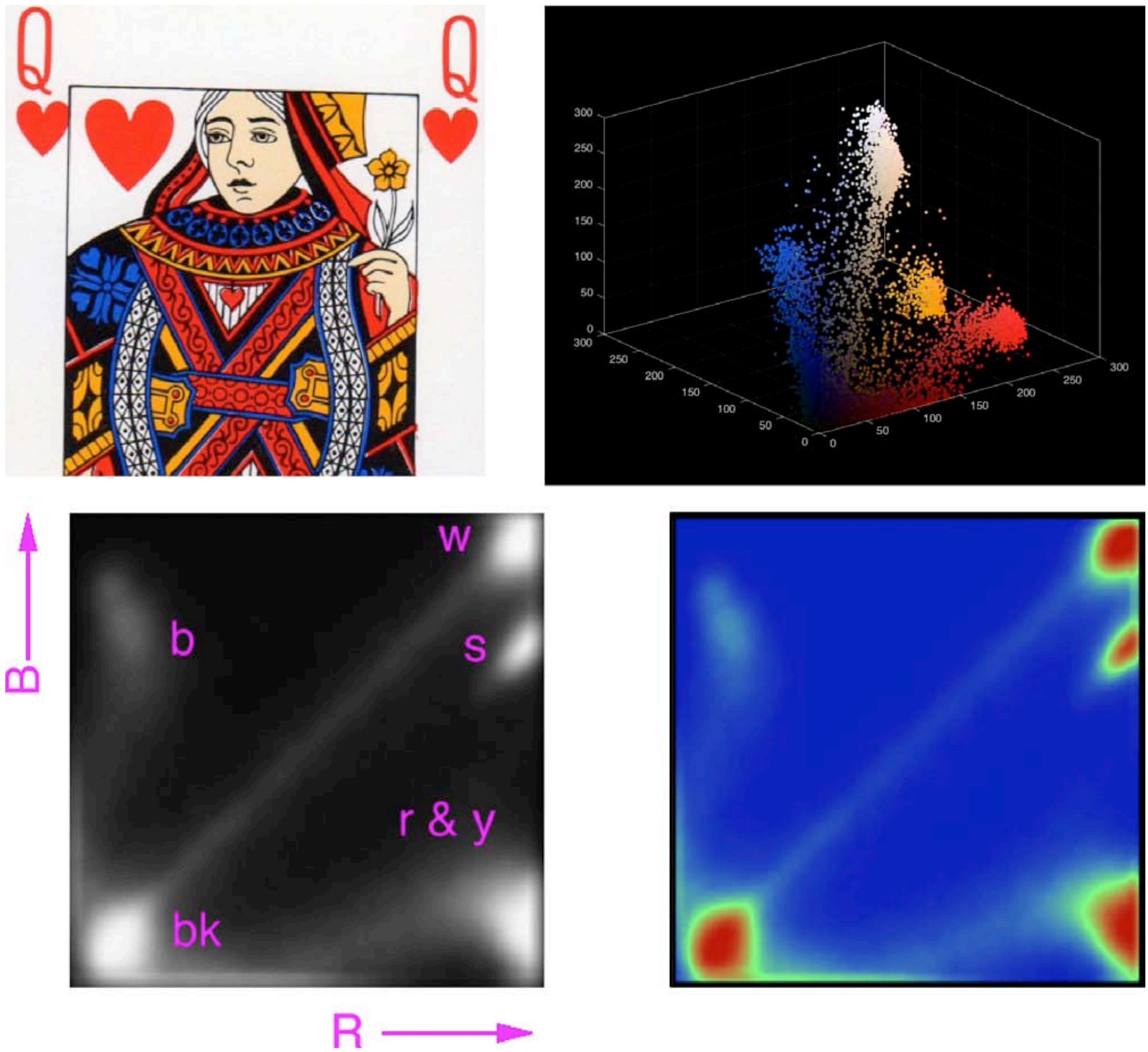


Colour scattergrams. (Top-left) Original image. Sweets. (Top-right). HS scattergram. (Centre & Bottom) RGB scattergram viewed from four different direction. The spot colour is the same as that of the corresponding pixel.

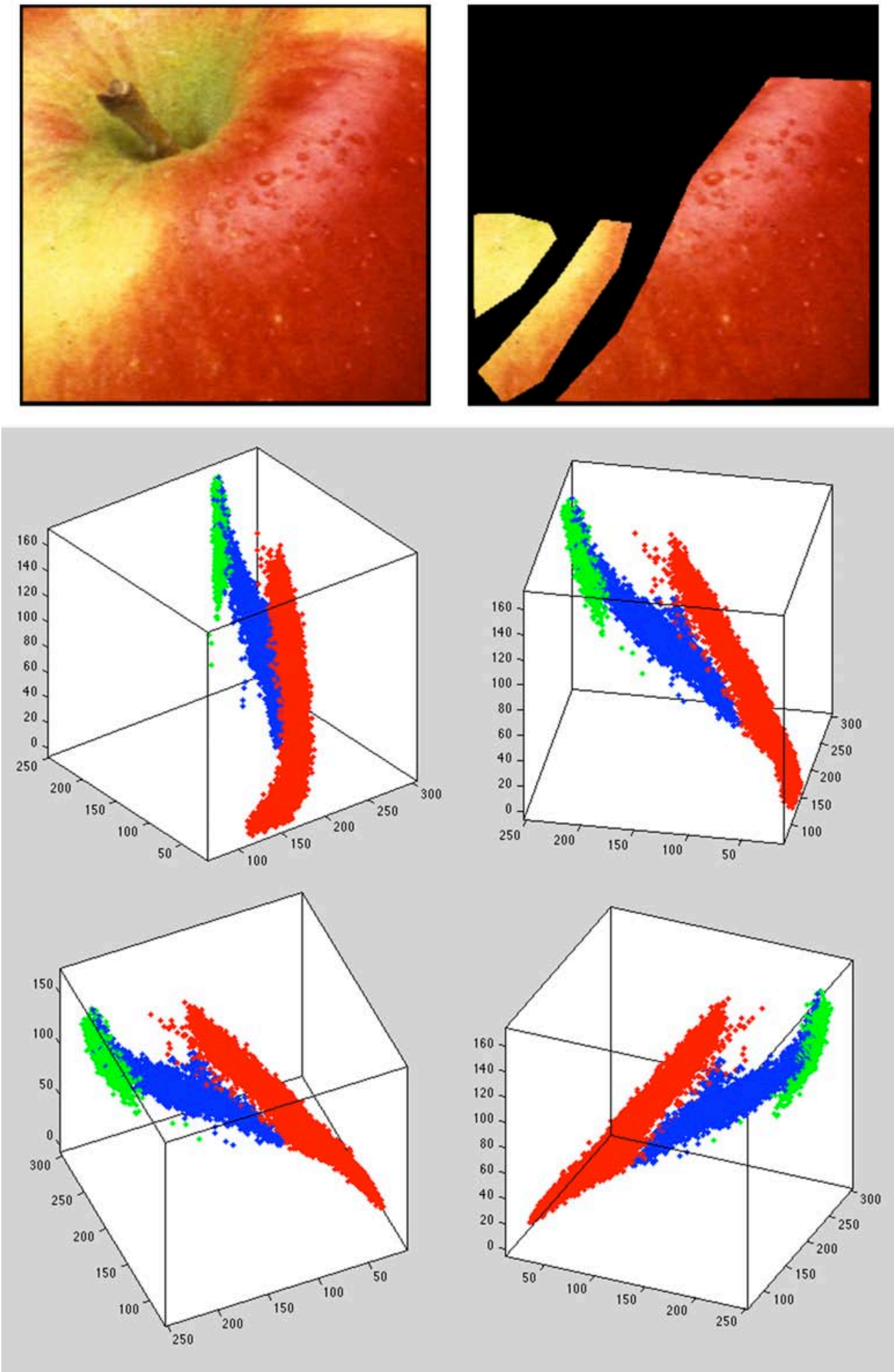
Figure 6.13



Clusters in RGB-space are investigated interactively using MATLAB™. (Top-left) Original image. (Three samples of coloured paper) (Top-right) Sample areas used to construct the scattergrams. (Centre and Bottom) RGB space viewed from different directions. The cluster colours correspond to the sample areas identified in [TR].

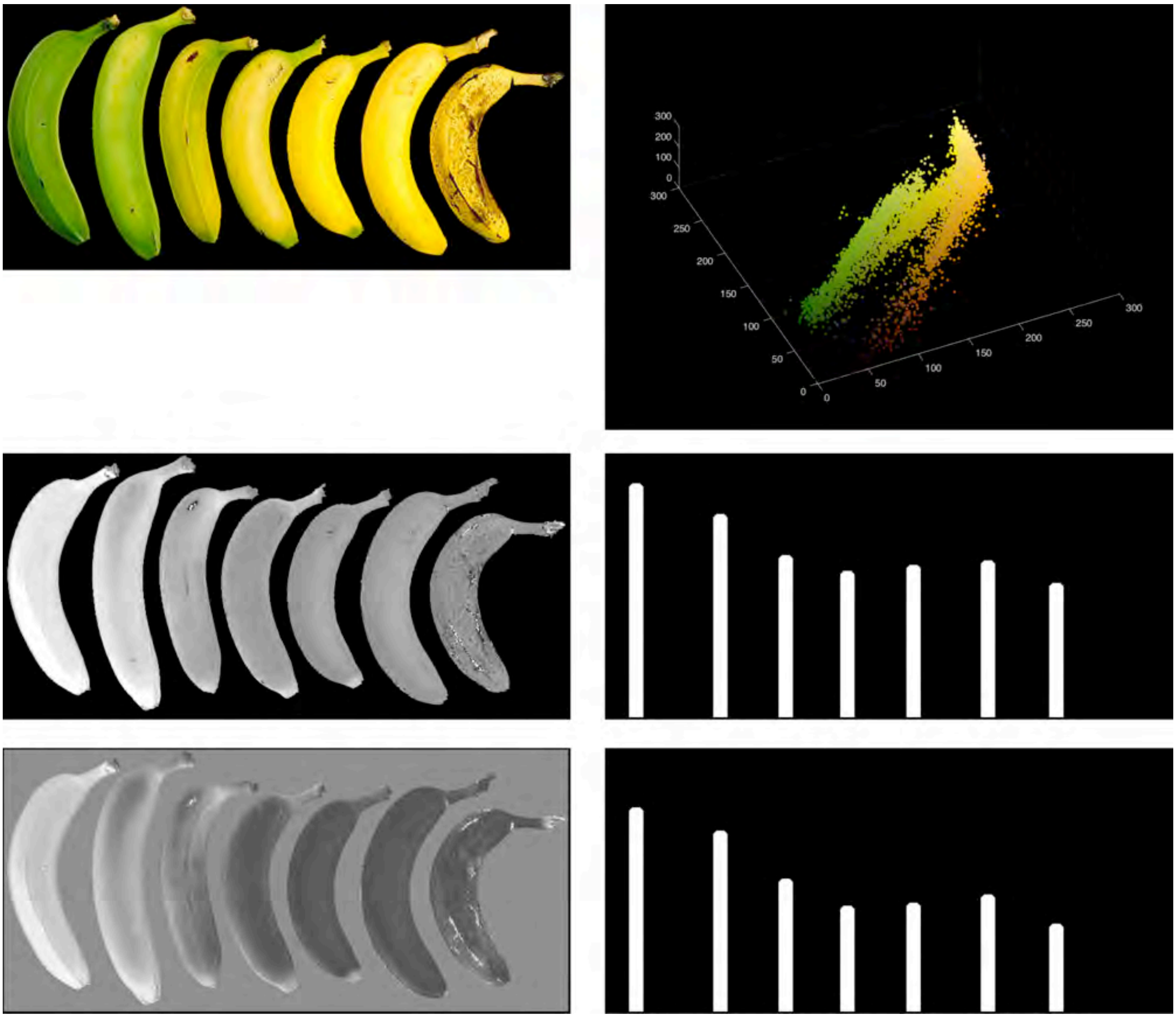


Colour clusters in RGB space and the RB-plane. (Top-left) Original colour image. (Top-right) Colour scattergram in RGB space. Bottom-left) Scattergram in the RB plane Key: w- white; s - “skin”; r & y - red & yellow (overlapping clusters); b - blue; bk - black. (Bottom-right) Pseudo-colour rendering of [BL] makes dark features more obvious.



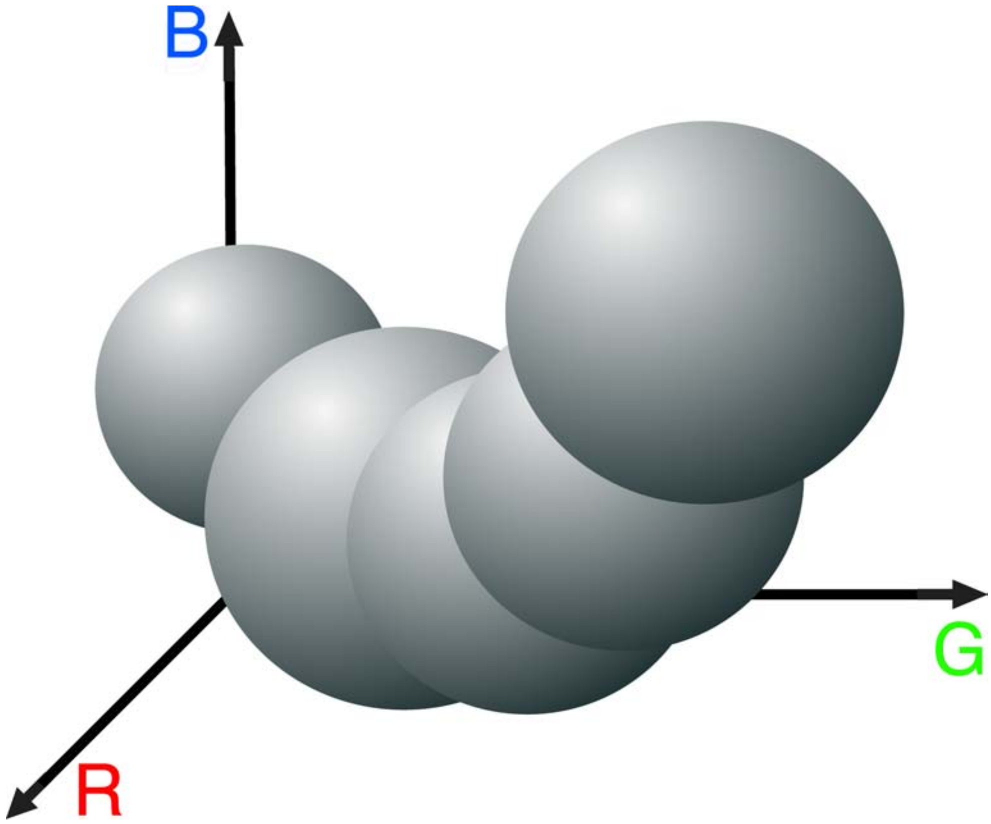
Investigating the range of colours on an apple. (Top-left) Original image. (Top-right) Sample areas used to build the scattergrams. (Centre and Bottom) RGB space viewed from different directions. Red points correspond to pixels within the red sample area. Green points correspond to pixels within the yellow sample area. Blue points correspond to pixels within the orange sample area.

Figure 6.16



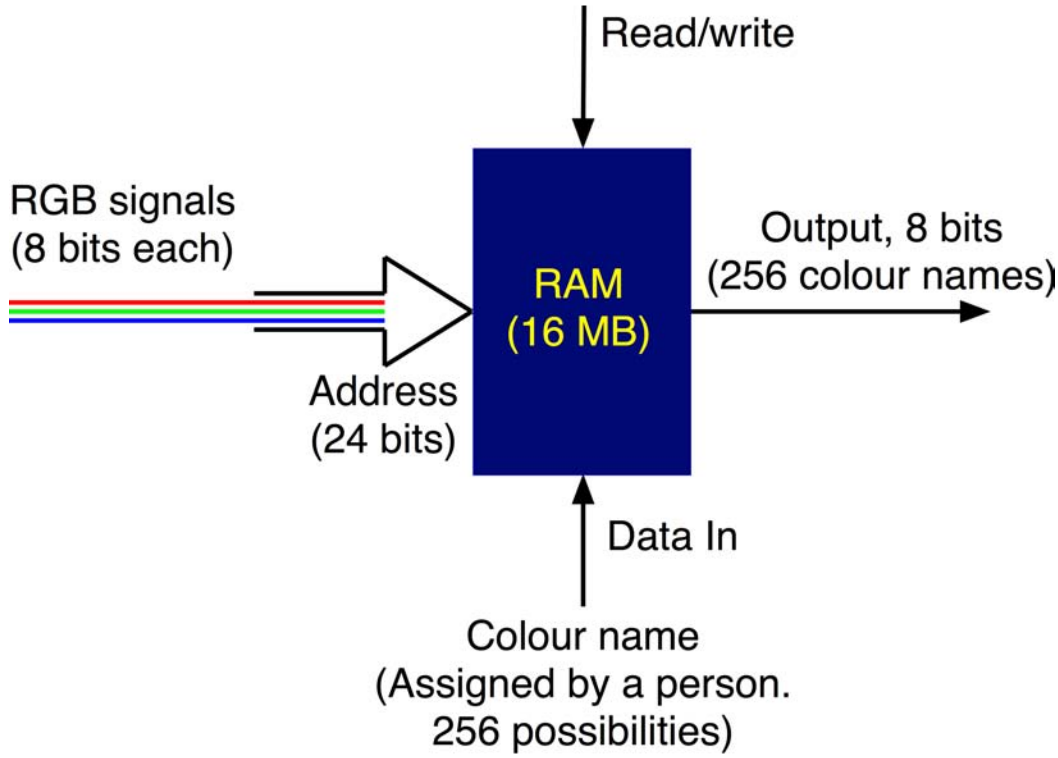
Monitoring the ripening of bananas. (Top-left) Original image. (Top-right) Colour scattergram. Notice the “folded” ripening path, resembling an upper-case “N”. (Centre-left) G-image. (Centre-right) Estimate of the average intensity of the central part of each banana in [CL]. (Bottom-left) [TR] subtracted from [CL]. (Bottom-right) Estimate of the average intensity of the central part of each banana in [BL].

Figure 6.17



Compound Classifier uses a series of overlapping spheres to recognise colours in RGB space. Suppose the spheres have previously been placed judiciously, so that together they enclose the cluster in RGB space representing a named colour, such as *"yellow",* and do not contain large voids where there are no such points. Then, any RGB measurements that place a point inside any one of the spheres will be classified as *"yellow"*. If a point lies outside all of the spheres, then is classified as *"not yellow".* The latter includes classes such as *"red", "green"*, *mauve", "orange,* etc. A representative sample of the class *"yellow"* can be built by merging examples of more restrictive colour sets, such as"butter", "sulphur", "lemon", "canary", "banana", etc.

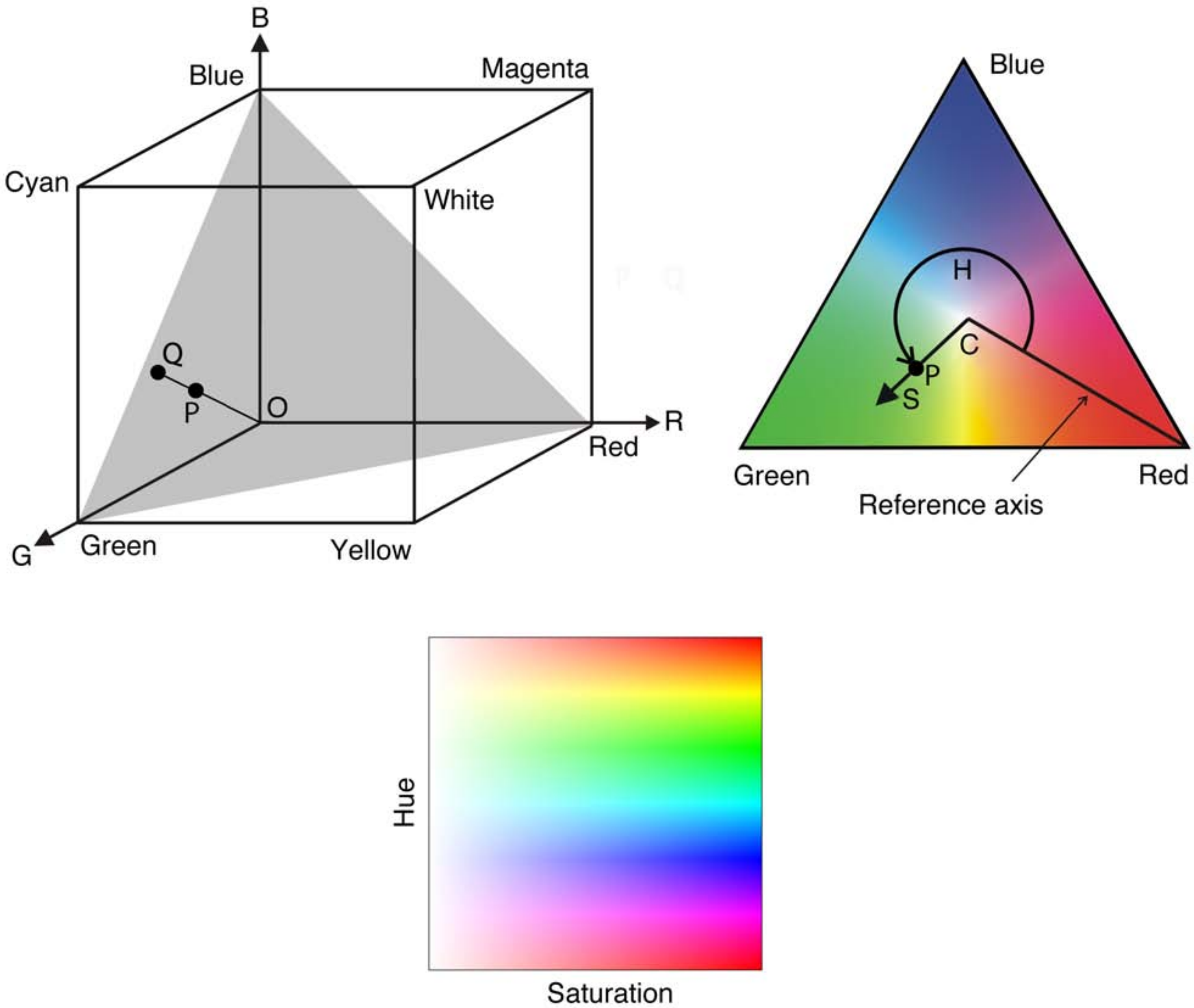
Figure 6.18



Storing a colour name for every point in RGB space (8-bits/channel) requires 16MB of storage, which is well within the capacity of a modern random access memory (RAM) chip. The input is formed by "bundling together" the R, G and B signals (3x8 bits) to form a 24 bit address. A RAM chip with a capacity of 16777216 (2^{24}) bytes can accommodate 256 (2^8) different colours. Additional points:

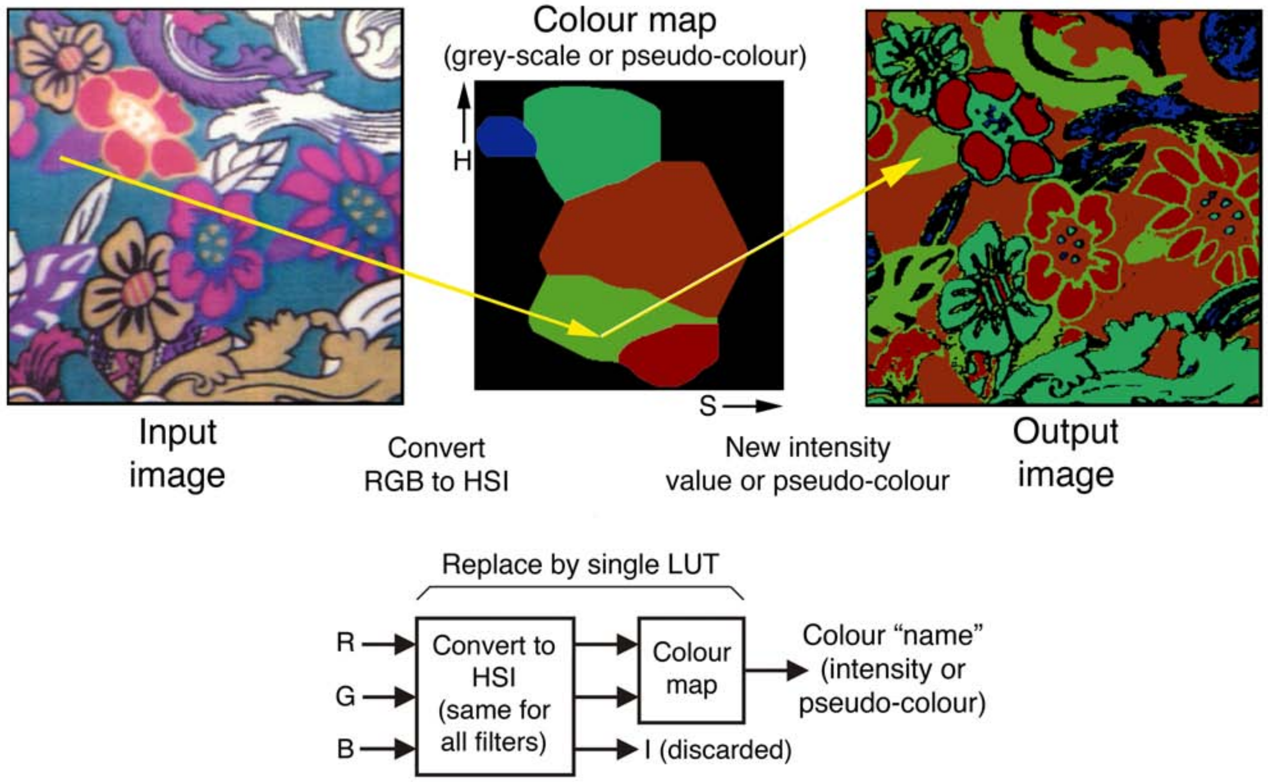
- Learning colours simply requires writing the data into the RAM at the address defined by the RGB inputs.
- All 16777216 data entries in the table must be set before this device can be used to identify colours. It has no ability to interpose, or generalise, given a few scattered data points in RGB space,
- Recognising colours requires reading data held in the RAM, at the address defined by the RGB inputs.

Figure 6.19



Relating hue and saturation to the RGB components. (Top-left) RGB space containing the Colour Cube. the point defined by RGB always lies inside this cube.The grey area is called the Colour (or Maxwell) Triangle. The point P is at the intersection of line OQ with the colour triangle. All points along OQ have the same values for hue and saturation. The intensity (I) increases with the distance from O to Q. (Top-right) The colour triangle. Its centre is at point C. Hue (H) is measured by the angle referred to some arbitrarily chosen reference axis. Saturation (S) increases with the distance from C to P. (Bottom) The HS plane represents the same information as the Colour Triangle in a way that is more convenient for programming. The HS plots in this chapter are all based on this model.

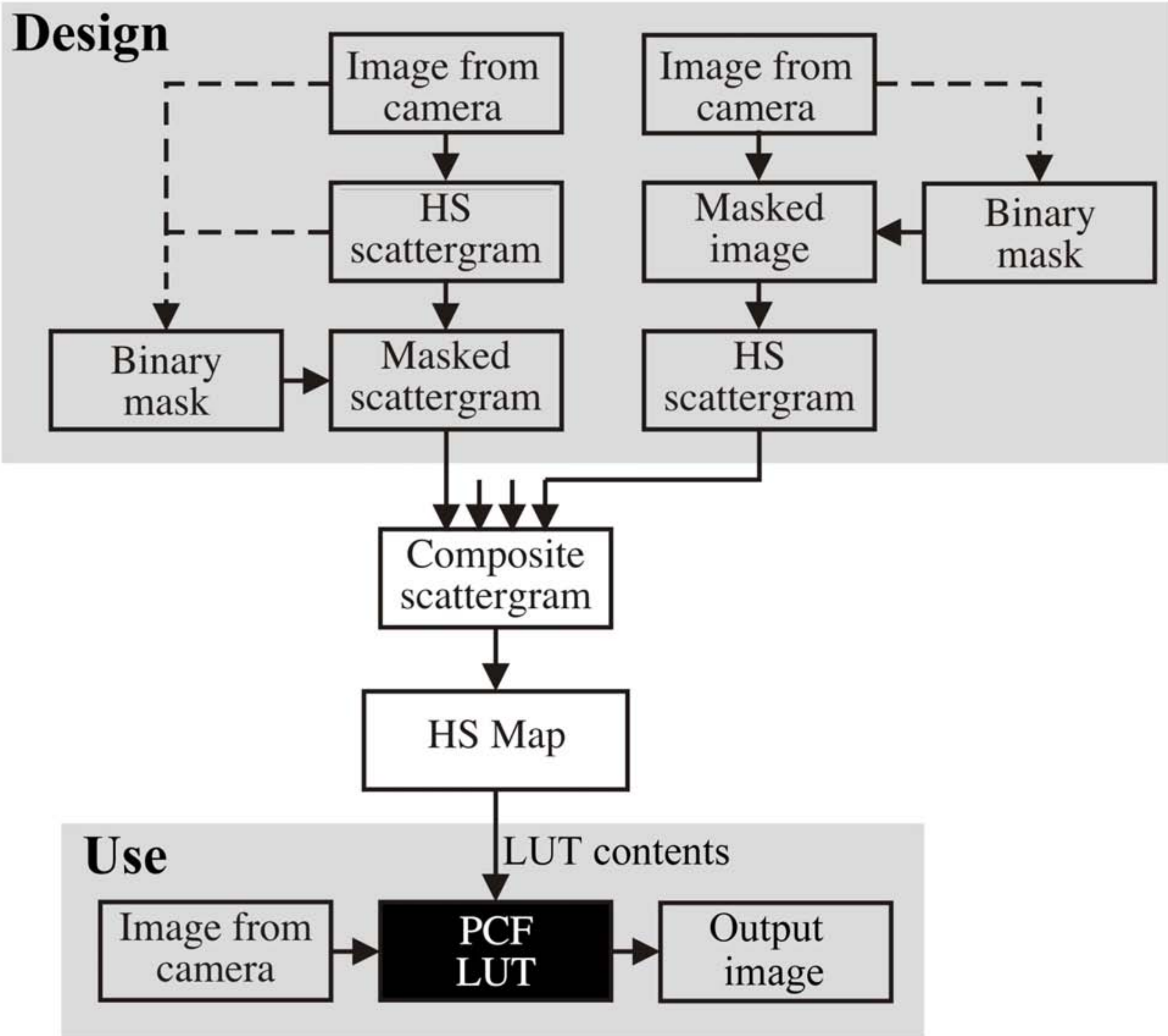
Figure 6.20



Colour recognition using a colour map based on the HS plane. (Top) Showing how a typical pixel is mapped into the output image. The RGB values for each input pixel are converted to hue and saturation (H and S). These values are used to address the colour map, which determines the new intensity value, or pseudo-colour as in this case, for the corresponding pixel in the output image. The filter is "programmed" by adjusting the colour map. (Bottom) A CRF has two parts. The RGB-to-HSI conversion is the same for all filters. Hence the "calculation" for colour recognition can be achieved using just one LUT. Once the colour map has been fixed, the whole process can be implemented conveniently in a single look-up table.

This completely ignores intensity, which has very little affect on colour perception in bright-light viewing conditions. (Photopic vision) Of course, under low-light conditions, human colour perception disappears. (Scotopic vision) This can be modelled in a CRF by simply adjusting the appropriate LUT entries corresponding to low-intensities.

Figure 6.21

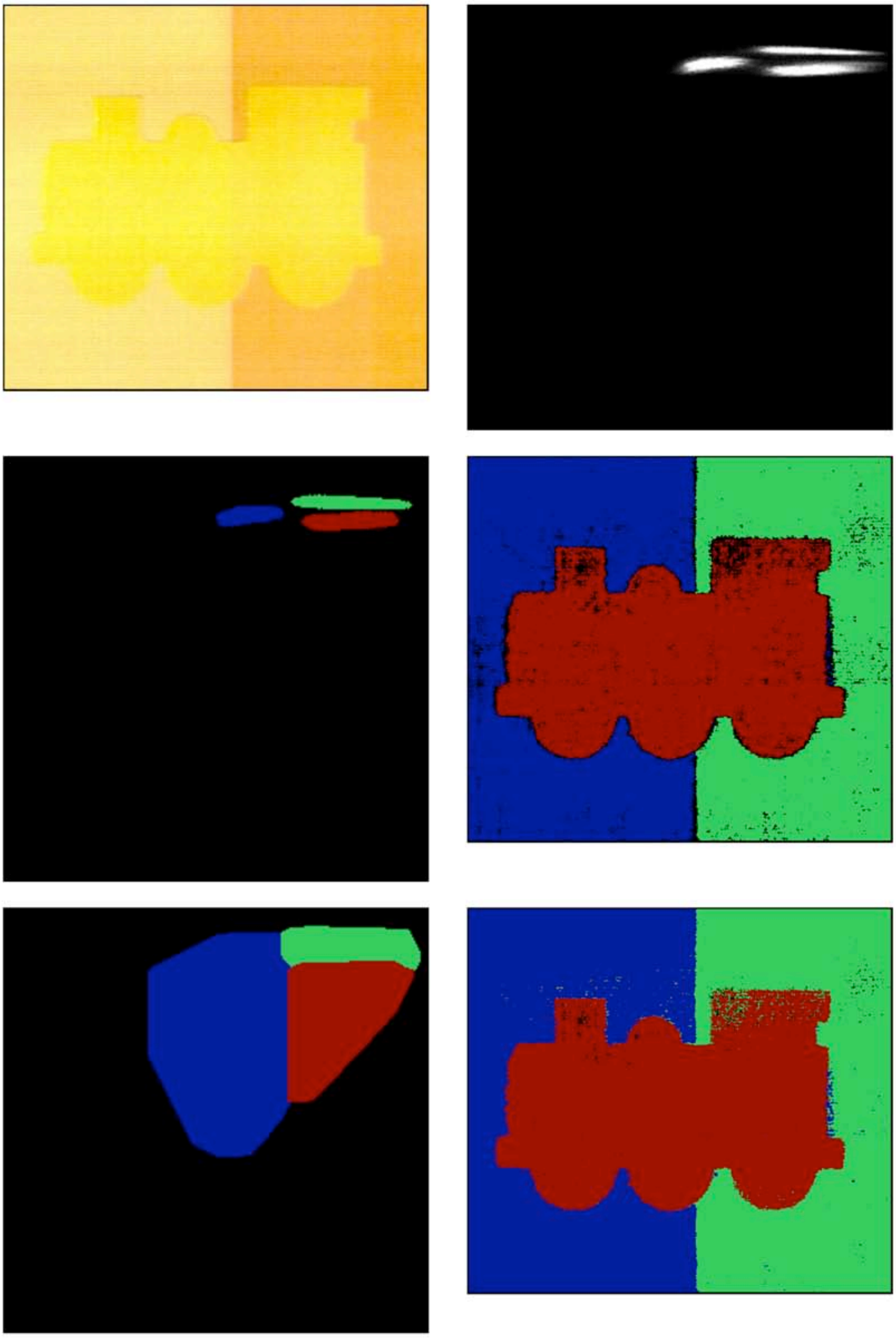


Design and use of a colour recognition filter. In practice, the colour map is designed interactively; the vision engineer must make subjective judgements, based on experimental try-it-and-see, to design an effective CRF. The design process is slow but, once the colour map has been constructed, the CRF is very fast.

Figure 6.22

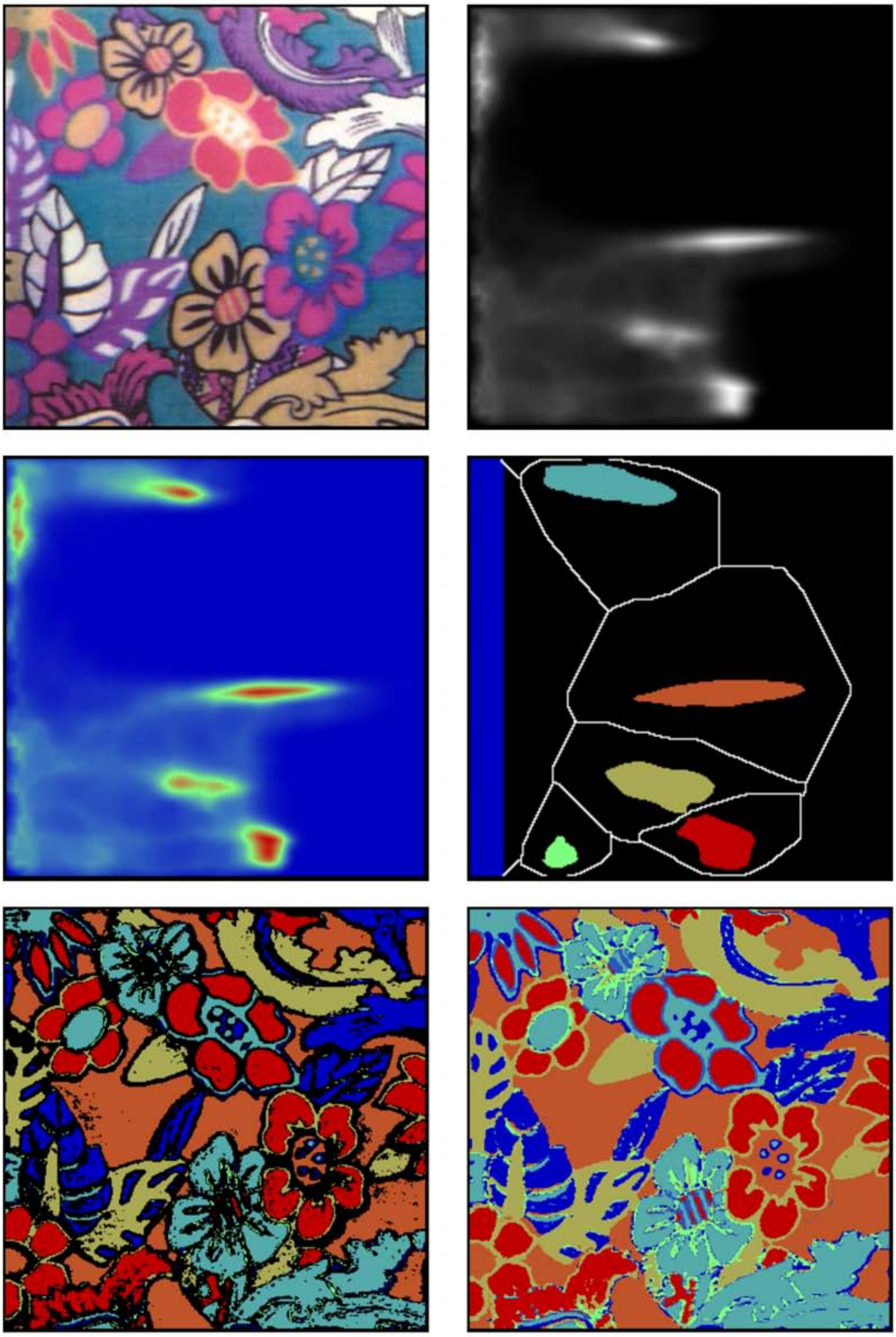
Colour recognition using a programmable colour filter. (Top-left) Original colour image. (Top-right) Scattergram in the HS plane. (Centre-left) [TR] has been thresholded and noise was removed by morphological filtering. (Centre-right) Applying the CRF using the colour map in [CL]. There are only four pseudo-colours here. (Bottom-left) Red pixels only. (Bottom-right) "Skin" pixels.

Figure 6.23



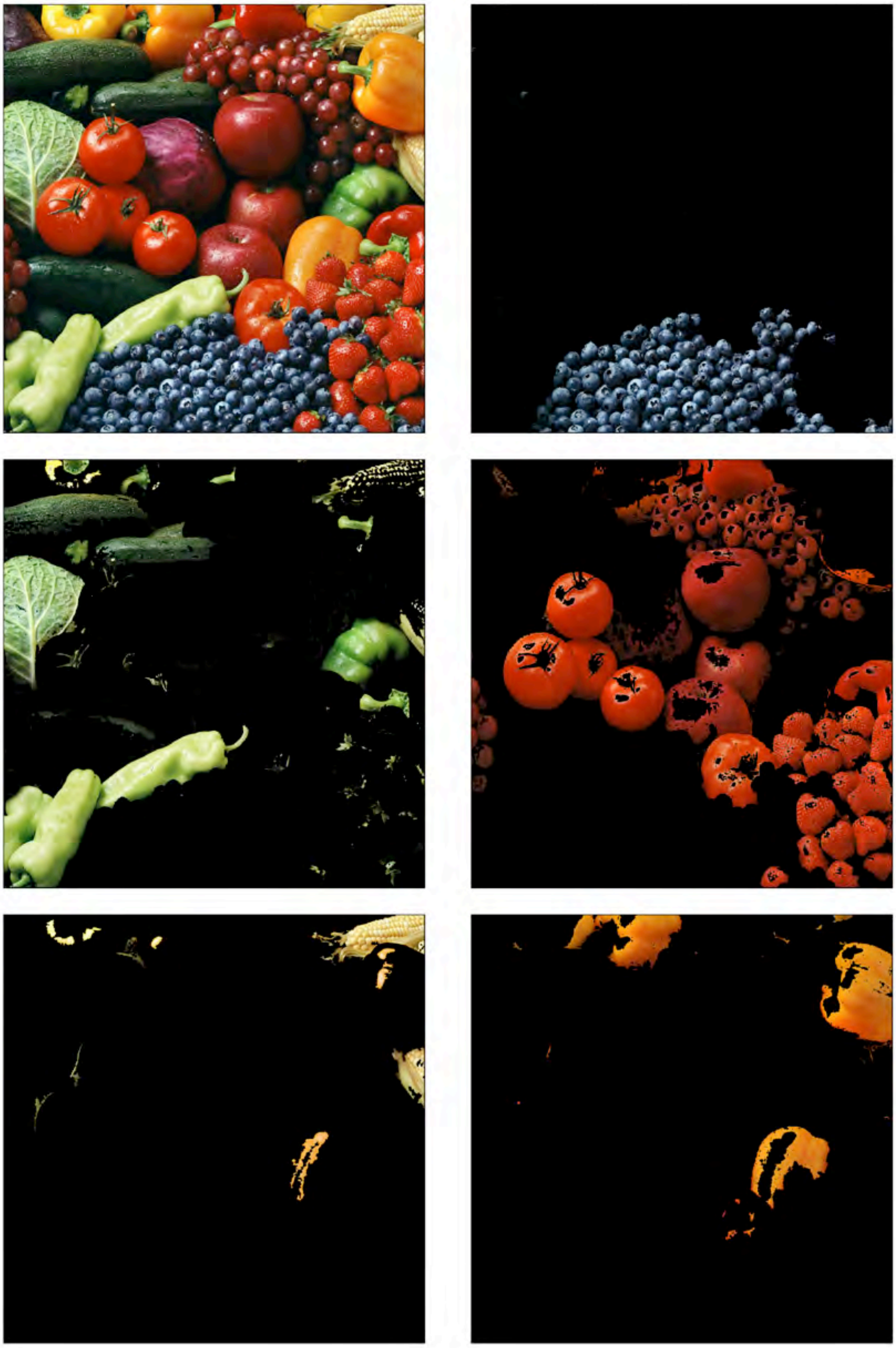
Distinguishing similar shades of yellow. (Top-left) Original image. (Top-right) Scattergram in the HS plane. (Centre-left) [TR] was thresholded, dilated slightly and pseudo-coloured. (Centre-right) Using [CL] as the mapping for a CRF. (Bottom-left) The blobs in [CL] have been enlarged. (Bottom-right) Using [BL] as the colour map for a CRF.

Figure 6.24



Colour recognition filter. (Top-left) Original image. [Dress fabric] (Top-right) Scattergram in the HS plane. (Centre-left) After pseudo-colouring. (Centre-right) The coloured blobs define Map1. The white lines indicate the boundaries of the enlarged blobs and define Map2. not shown explicitly (Bottom-left) Using Map1 in the CRF. (Bottom-right) Using Map2 in the CRF.

Figure 6.25

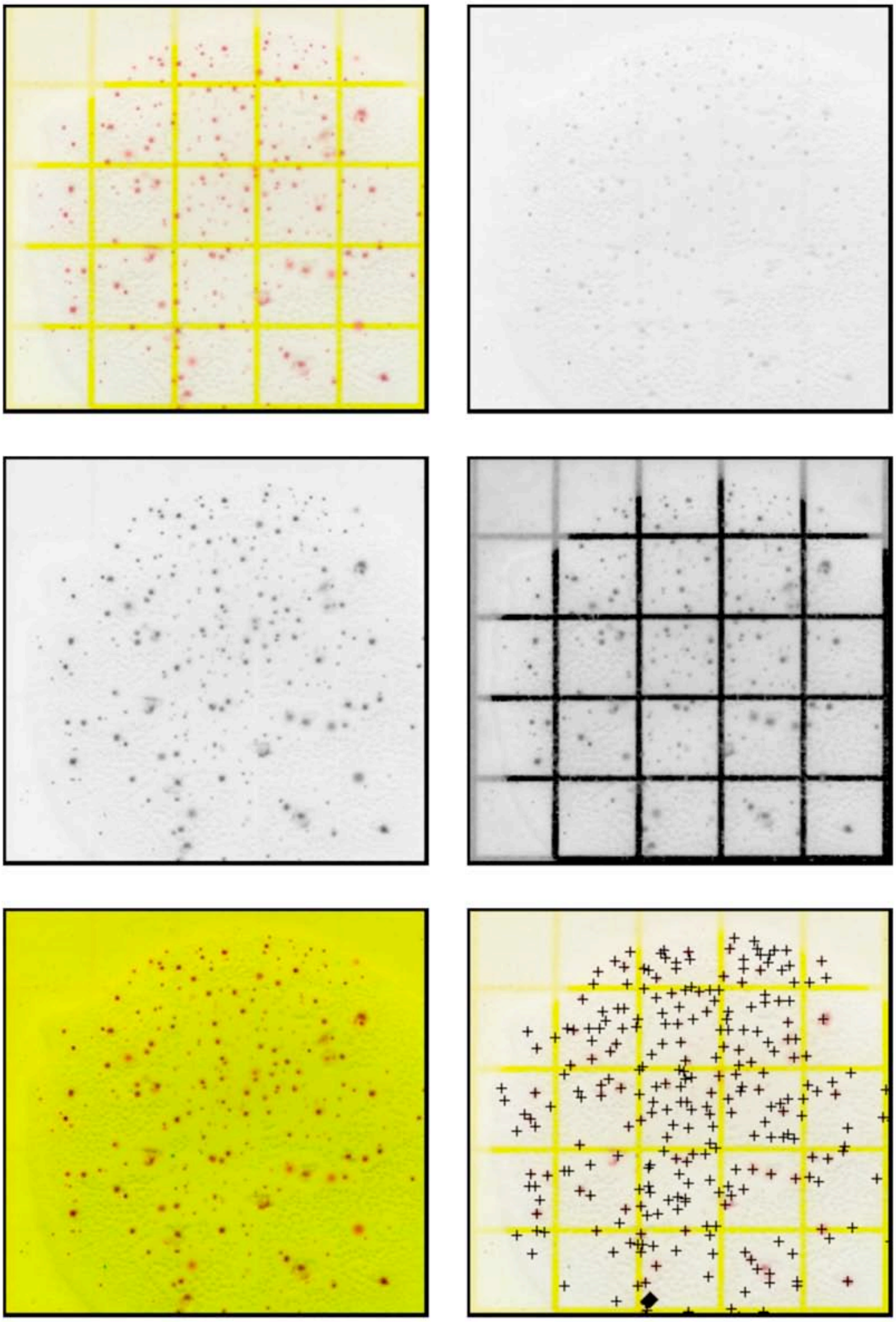


Using a colour recognition filter (CRF) to segment a scene with several distinct colours (Top-left) Original image.

Figure 6.26

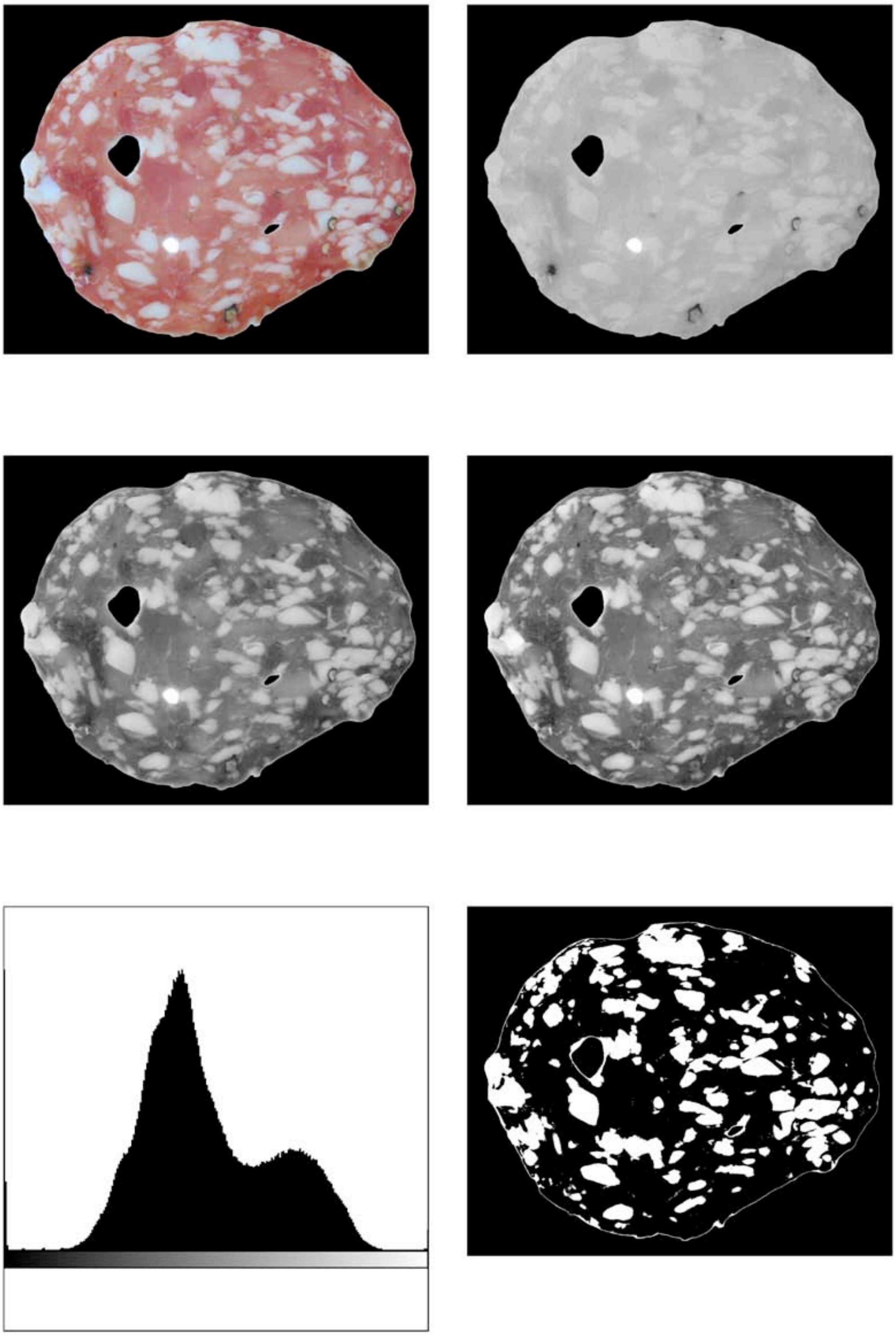
Tomatoes. (Top-left) Original image. (Top-right) Colour map derived from the red parts of [TL]. (Centre-right) RGB space for the red parts of [TL]. (Centre-right) CRF applied to [TL] using the (monochrome version of the) map in [TR]. (Bottom-right) [TL] masked using [CR]. (Bottom-right) Red parts in [BL] have been removed.

Figure 6.27



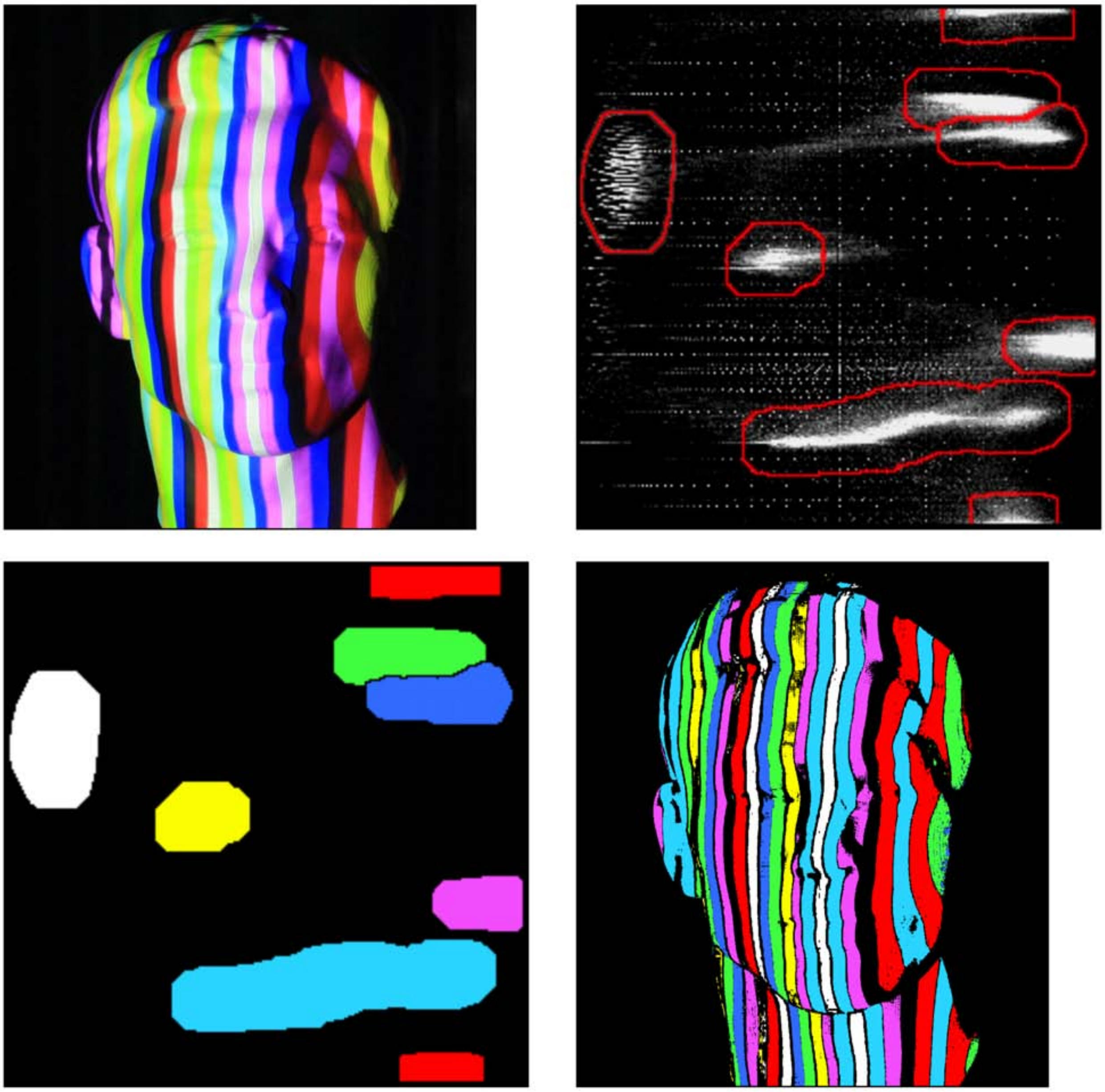
Counting bacteria on a culture plate. (Top-left) Original image. (Top-right) R-image.(Centre-left) G-image. (Centre-right) B-image. (Bottom-left) Each of the RGB channels was enhance separately. (Bottom-right) Crosses mark the centroids of red spots found by the CRF.

Figure 6.28



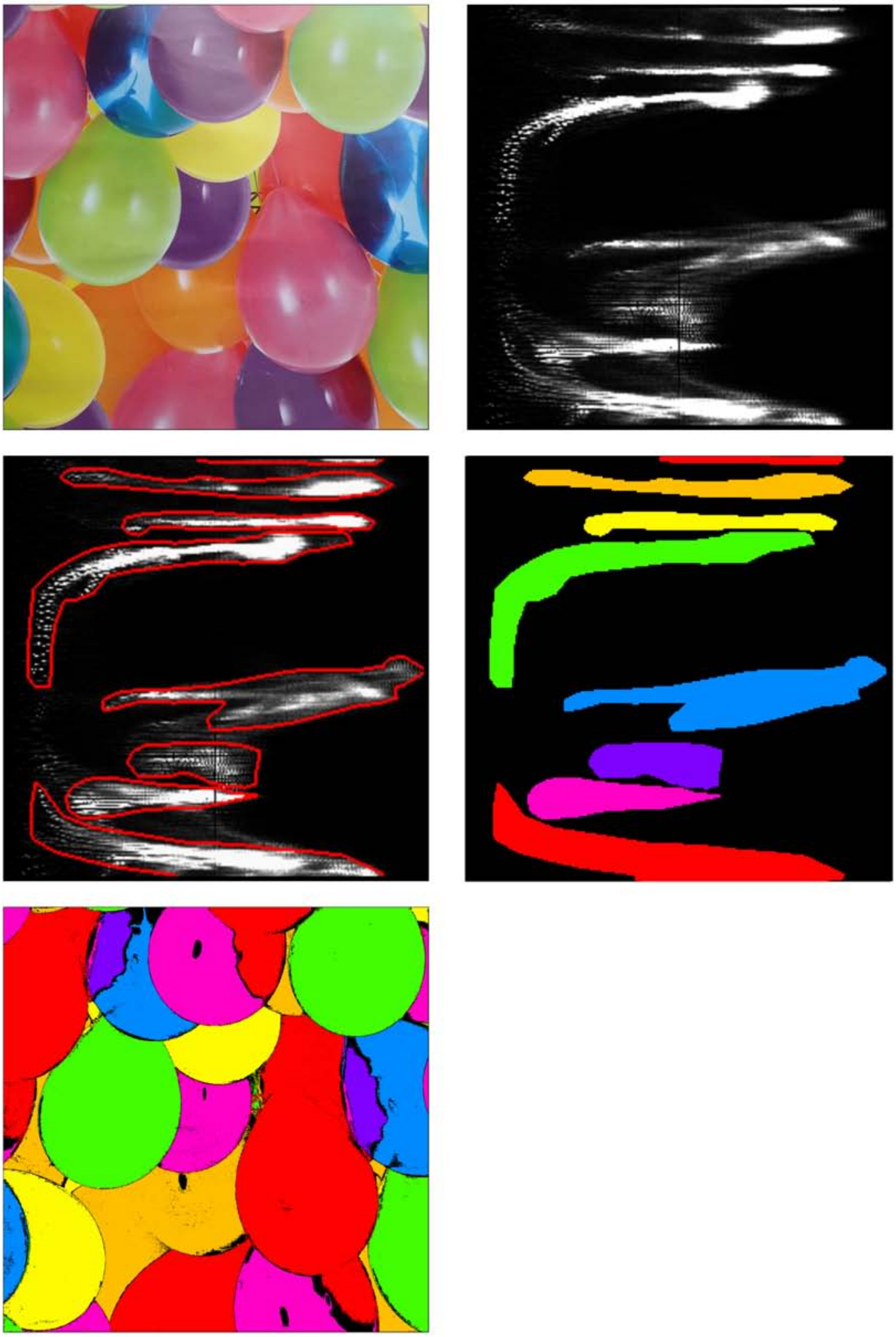
Measuring the fat content of cooked meat. (Top-left) Original image. (R-image.(Centre-left) G-image. (Centre-right) B-image. (Bottom-left) Intensity histogram of the G-image. (Bottom-right) G-image thresholded. Parameter corresponds to the bottom of the valley in [BL].

Figure 6.29

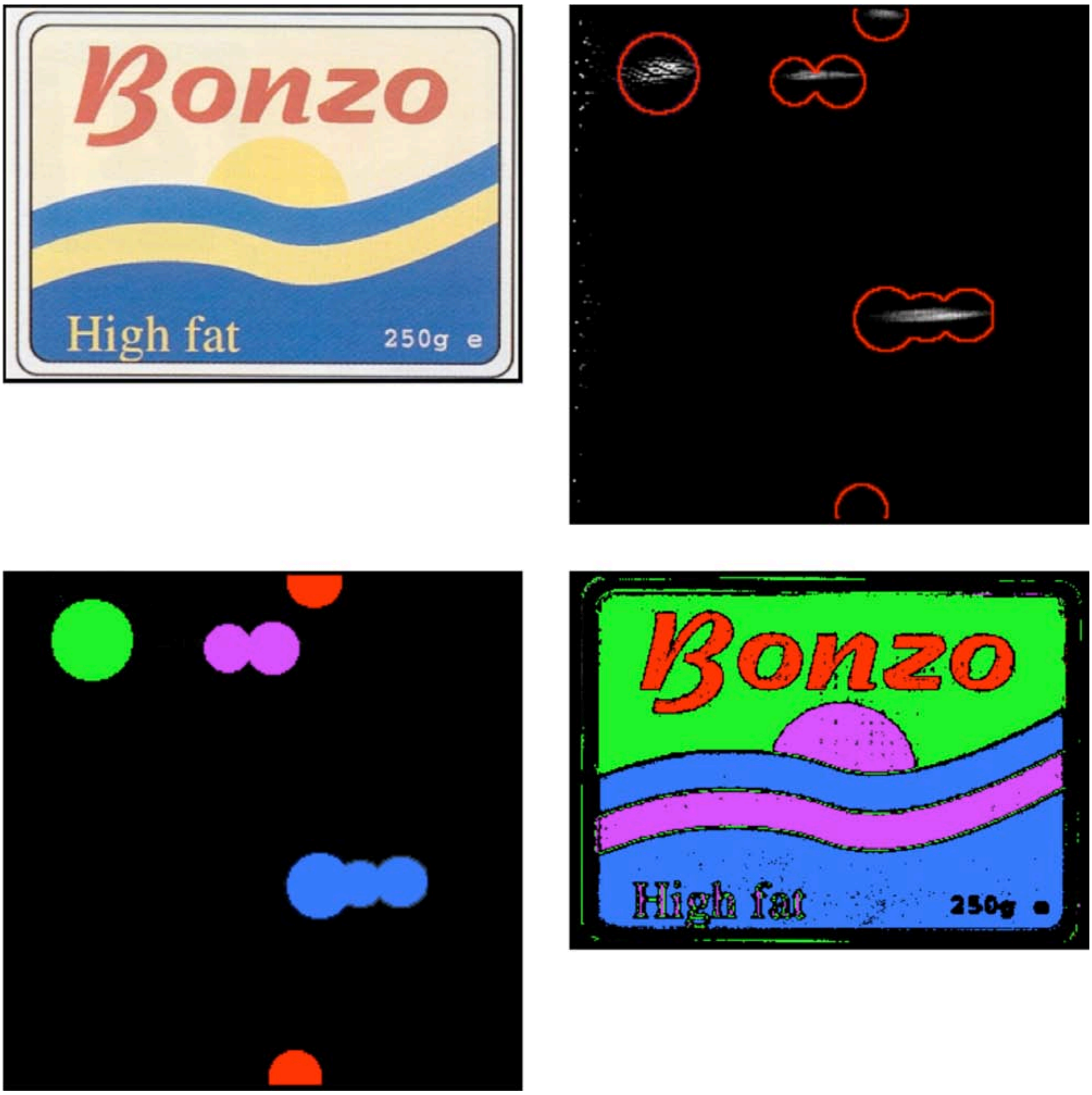


Multi-colour structured lighting. [Muti-coloured light stripes are projected onto an object surface, which is viewed from a different direction. This is a popular method for 3D shape analysis.] (Top-left) Original image. (Top-right) Scattergram in the HS plane. The red contours were drawn manually. (Bottom-left) Colour map, derived from [TR]. Two blobs, both associated with shades of red, are given the same pseudo-colour. [Hue is a cyclic property.] (Bottom-right) Output from the CRF using the colour map in [BL].

Figure 6.30

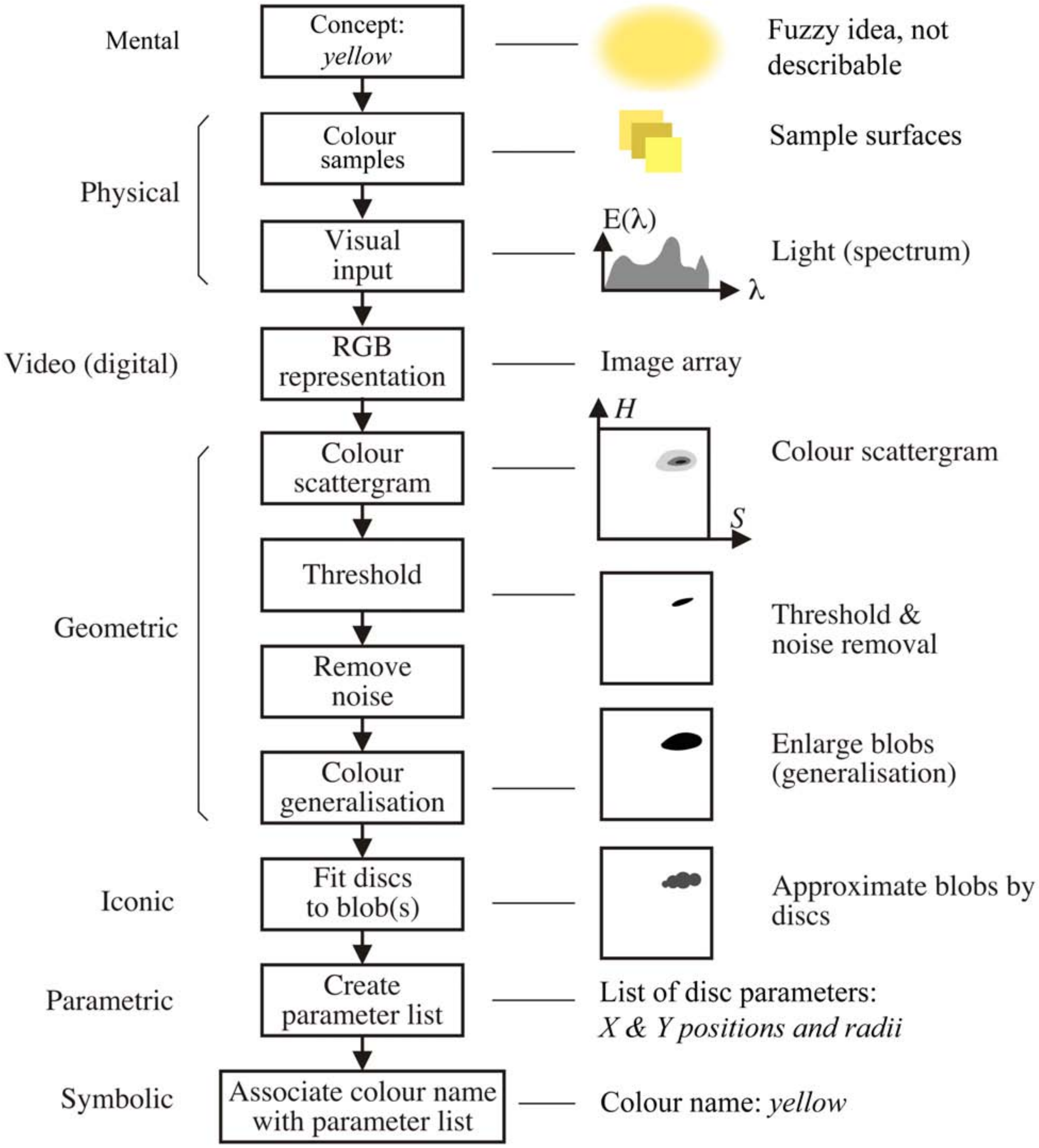


Designing a CRF manually. (Top-left) Original image. [Wrapping paper] (Top-right) Scattergram in the HS plane. (Centre-left) Polygons drawn around the scattergram spots. (Centre-right) Colour map, derived from the red contours in [CL]. (Bottom-left) Output of the CRF using [CR] as the colour map.

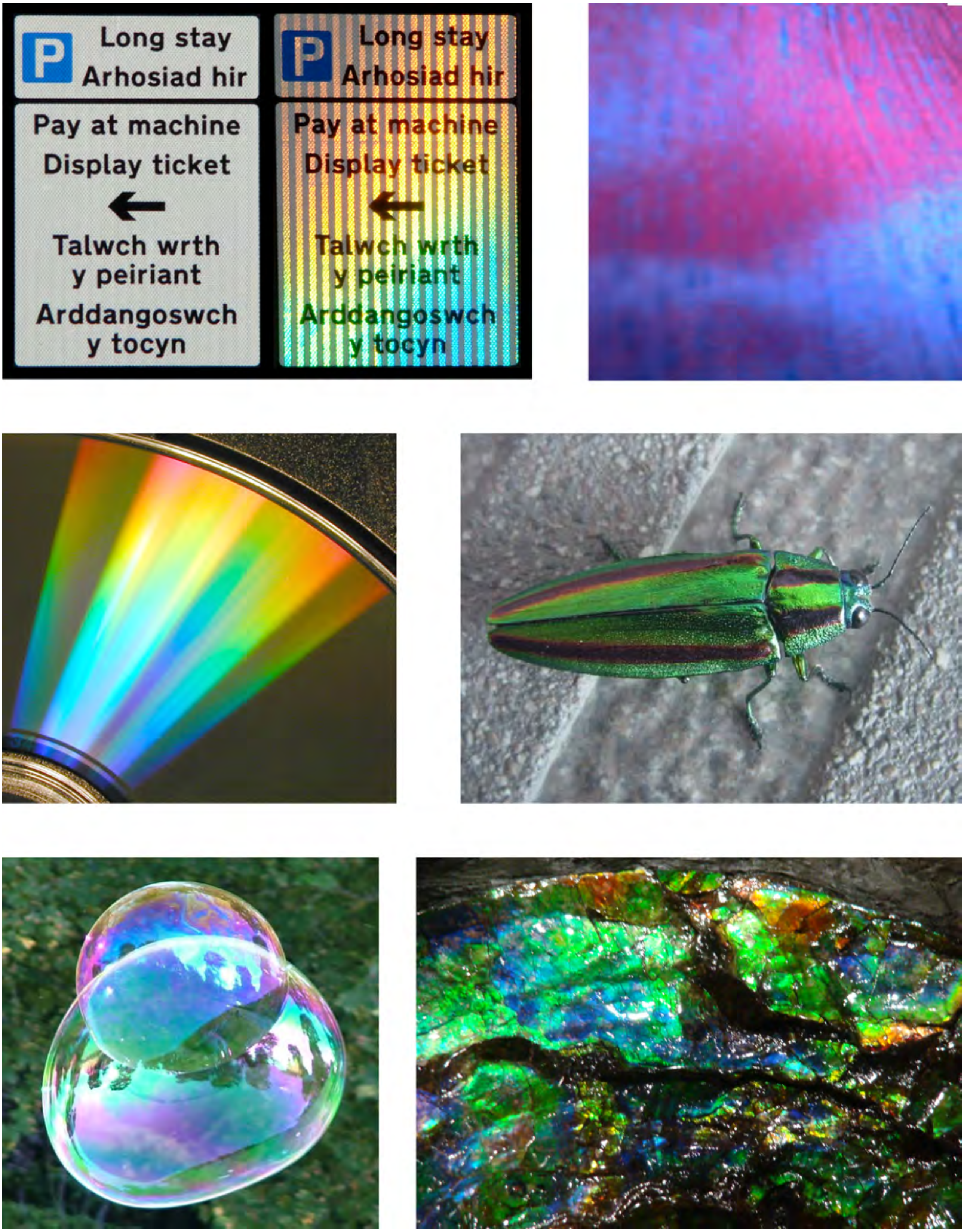


Approximating the HS scattergram with overlapping discs.
(Top-left) Original image. [Plastic food container]
(Top-right) HS scattergram. (The red circles (edges of the discs) do not contribute to the CRF results.)
(Bottom-left) Colour map derived from (TR].
(Bottom-right) Applying the CRF with the colour map in [BR].

Figure 6.32

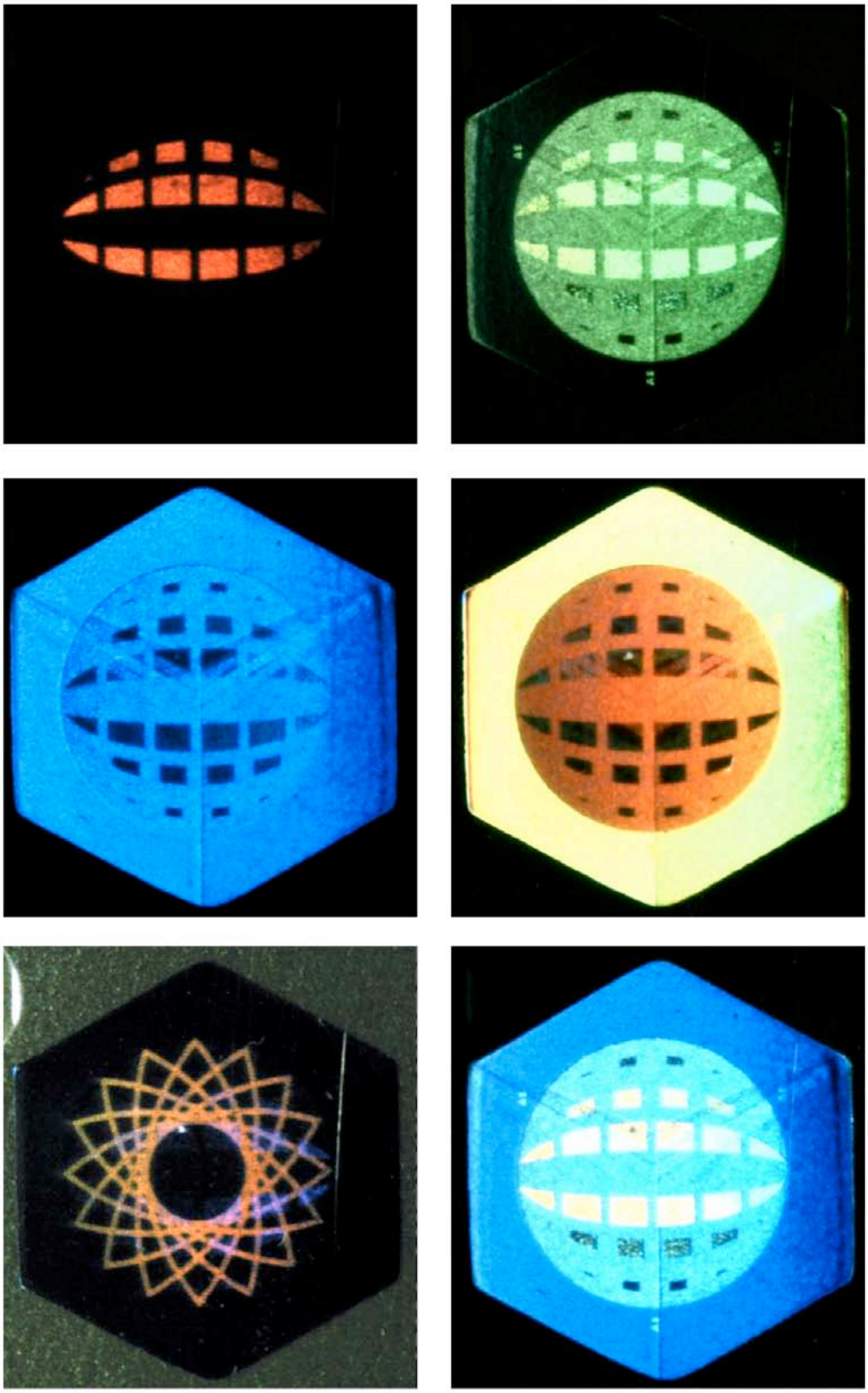


How colour is represented while designing a CRF.



Structural colour. (Top-left) Sunlit street sign. The angle of view was altered slightly. (Top-right) Silk fabric. (Centre-left) DVD, interference effect. (Centre-right) Beetle interference. (Bottom-left) Soap bubble, interference. (Bottom-right) Ammolite, interference.

Figure 6.34



A hologram under varying lighting-viewing conditions. In [TL], [TR] and [CL], light is projected towards the hologram from the left, by red, green and blue LEDs, respectively. In the other three images, light is projected from the right, by other types of light source.

Figure 6.35