

# A SYSTEM TO REDUCE THE EFFECT OF CCDS SATURATION

Jordi Regincós-Isern

Dept. d'Informàtica i Matemàtica Aplicada  
Escola Politècnica Superior,  
Universitat de Girona,  
av. Santaló s/n, 17003 Girona (Spain)  
Email: jordir@ima.udg.es

Joan Batlle

Dept. d'Enginyeria Industrial  
Escola Politècnica Superior,  
Universitat de Girona,  
av. Santaló s/n, 17003 Girona (Spain)  
Email: jbatlle@ei.udg.es

## ABSTRACT

Colour image segmentation based on the hue component presents some problems due to the physical process of image formation. One of that problems is colour clipping, which appear when at least one of the sensor components is saturated. We have designed a system, that works for a trained set of colours, to recover the chromatic information of those pixels on which colour has been clipped. The chromatic correction method is based on the fact that hue and saturation are invariant to the uniform scaling of the three RGB components. The proposed method has been validated by means of a specific colour image processing board that has allowed its execution in real time. We show experimental results of the application of our method.

## 1. INTRODUCTION

Although most of the usual problems in computer vision can be solved on grey-level images, colour has been proved to be a useful information in computer vision systems. In some situations, having chromatic information can highly improve the accuracy of the algorithms (see, for instance, Luong [1] and Gershon [2] papers). One of the main topics of research in colour vision is the study of *colour constancy*, that is, to get the chromatic information of a region of the image independently of the characteristics of the illuminant, of the sensor and of the geometry of the scene.

When an image of an object is acquired, the colour of each pixel depends on several parameters: from the characteristics of the devices used on the acquisition process (illuminant, camera, acquisition board) to the physical process of image formation (see Novak [3]).

For a single pixel  $x$ , the response of the camera is given by the following equation:

$$C_x = \begin{bmatrix} r_x \\ g_x \\ b_x \end{bmatrix} = \begin{bmatrix} \int X(\lambda)\tau_r(\lambda)s(\lambda)d\lambda \\ \int X(\lambda)\tau_g(\lambda)s(\lambda)d\lambda \\ \int X(\lambda)\tau_b(\lambda)s(\lambda)d\lambda \end{bmatrix} \quad (1)$$

where  $X(\lambda)$  represents the amount of incident light,  $s(\lambda)$  correspond to the camera responsivity and  $\tau_r(\lambda)$ ,  $\tau_g(\lambda)$ ,  $\tau_b(\lambda)$  correspond to the filters transmittance of the three components red, green and blue (according to Shafer notation [4]).

Cameras are finite devices, so, there is a maximum value that cameras can provide called the dynamic range. It is usually assumed that camera output ranges from (0,0,0) to (1,1,1). Every value of  $r_x$ ,  $g_x$  or  $b_x$  on equation 1 greater than 1 is clipped by the camera to 1 and it is said that this component is saturated.

Some work has been done in order to recover the chromatic information of those pixels of which some components are saturated. These works are based on making assumptions about the actual colour of the pixel from the colour of the neighbouring pixels; see for example the works of Perez and Koch [5] and Klinker et al. [6].

We propose a method to recover the chromatic information of a pixel without looking at the neighbouring pixels.

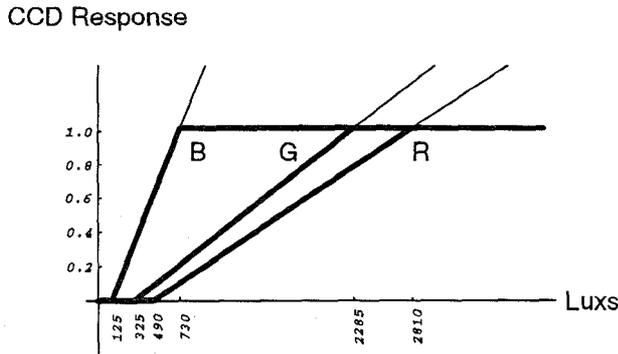
## 2. APPROACH

There is a lot of work already done in segmentation algorithms based on the chromatic characteristics of the regions, usually hue and saturation (see, for instance, Celenk [7] or Perez and Koch [5] papers). Hue and saturation are normally defined in terms of  $(r, g, b)$  values as:

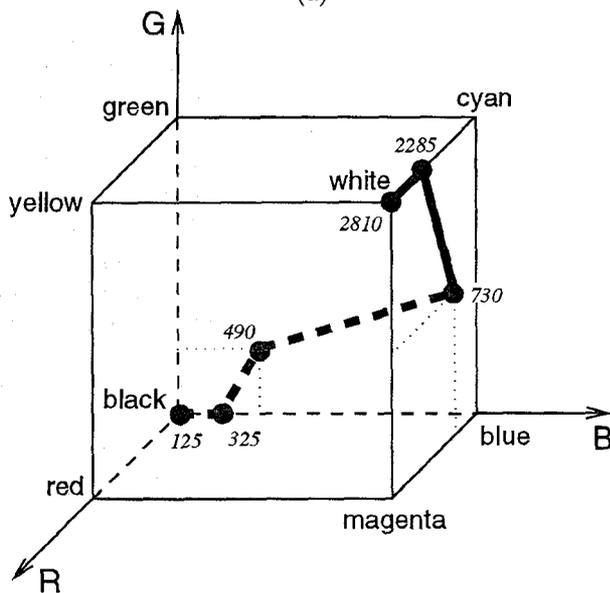
$$H(r, g, b) = \tan^{-1} \frac{\sqrt{3}(g - b)}{(r - g) + (r - b)} \quad (2)$$

$$S(r, g, b) = 1 - \frac{3 \min(r, g, b)}{r + g + b} \quad (3)$$

From the previous equation it can be seen that  $H(\alpha r, \alpha g, \alpha b) = H(r, g, b)$  and that  $S(\alpha r, \alpha g, \alpha b) = S(r, g, b)$ . That is, hue and saturation are stable under variations on the intensity of the illuminant (they have the scale-invariance property).



(a)



(b)

Figure 1: Colour clipping effect for a purplish blue coloured surface: (a) sensor response for each component depending on the illuminant intensity. (b) The same response represented on the RGB space.

Assuming that the CCDs output is linear, for a uniformly coloured surface, the camera response will follow a straight line on the RGB space while the intensity of the illuminant is increased. This linear behaviour does not hold when one of the components reaches the max-

imum value (see figure 1). This effect is known in the literature as colour clipping.

Segmentation algorithms based on hue and saturation does not work for those pixels affected by the colour clipping because the response of the camera is not linear. Figure 2 shows how hue values change when the illuminant intensity varies. In this case, hue has the right value only between 490 and 730 luxs.

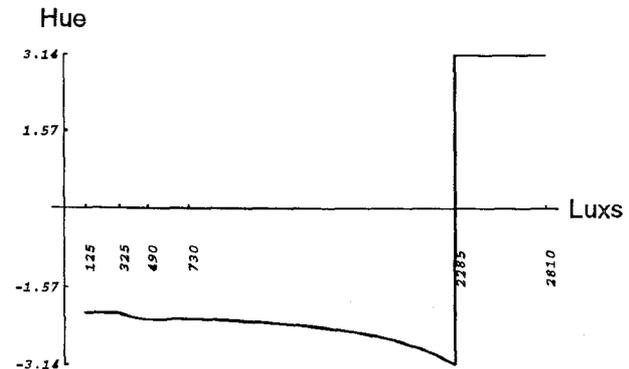


Figure 2: Behaviour of the hue component depending on illuminant intensity by the same colour shown at figure 1 (hue ranges from  $-\pi$  to  $\pi$ ).

Figure 3 shows one of the test images (a train toy) which has been acquired with a high illumination level. Figure 4 (top) shows the profile of the hue component on the line drawn on figure 3, and figure 4 (bottom) shows the result of the hue-based segmentation of the engine. As it can be observed, there are several pixels that have not been considered as part of the object.

Given an uniform coloured surface, it is possible to acquire a series of frames varying the intensity of the illuminant. From these images we can find the lines  $(\rho_r(l), \rho_g(l), \rho_b(l))$  that approximate the variation of the camera response for any intensity  $l$ , as it is shown on figure 5. Looking at this figure, it can be seen that there is an interval of luxs  $[l_1, l_2]$  in which the three components have a linear behaviour. For intensities below  $l_1$  at least one component is 0 and for those above  $l_2$  at least one component is saturated: if  $l > l_2$  the red component of the response of the camera will not be the expected  $\rho_r(l)$  but will be 1, and so, the hue value will change.

Our goal is to increase, virtually, the interval  $[l_1, l_2]$ , in order to get a linear response in a wider range of luxs. To carry it out, we have designed an algorithm based on the scale-invariance property of hue and saturation; that is, point  $(\rho_r(l), \rho_g(l), \rho_b(l))$  contains the same chromatic information that point  $(k\rho_r(l), k\rho_g(l), k\rho_b(l))$ . The idea of the algorithm is that if  $\rho_r(l)$  is greater than 1, we can find a value  $k$  such that  $k\rho_r(l)$  is equal or less

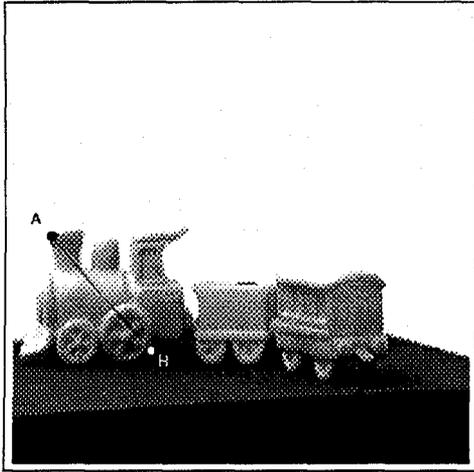


Figure 3: Original image of a toy train.

than 1 (and the same for the green and blue component).

Finally, the algorithm has been implemented on a colour image processing board and it works in real time.

### 3. ALGORITHM AND IMPLEMENTATION

Once the series of images has been acquired, we have a set of  $\{(r_i, g_i, b_i)\}$  triplets, one for each illuminant intensity used (figure 5 shows a possible representation of one set of values). From this values we identify, for each component, the interval of luxs on which its behaviour is linear. Then, a straight line is fitted to the values that fall into this interval using a least-squares method. The three lines  $\rho_r(l), \rho_g(l), \rho_b(l)$  are obtained. From this lines, the interval of luxs  $[l_1, l_2]$  on which hue is constant is obtained. In order to broaden this interval, a new value of  $l_2$  is calculated (for example, we can use the value  $l'_2$  such that for every  $l > l'_2$  at least two components are saturated). Then, a number  $k$  is calculated in order to guarantee that the line values  $(k\rho_r(l), k\rho_g(l), k\rho_b(l))$  falls in the interval  $[0, 1]$  when  $l$  ranges from  $l_1$  to  $l'_2$ . Finally, what the algorithm does is: for each  $l \in [l_1, l'_2]$ , the camera response  $(\rho_r(l), \rho_g(l), \rho_b(l))$  is estimated taking into account the colour clipping effect; and the algorithm assigns to pixels with these values the corrected ones  $(k\rho_r(l), k\rho_g(l), k\rho_b(l))$ .

Once the pixel values have been corrected, a transformation to the HSI colour space is performed in order to segment the object.

Figure 6 shows the results obtained when the proposed method has been applied on image 3. As it can

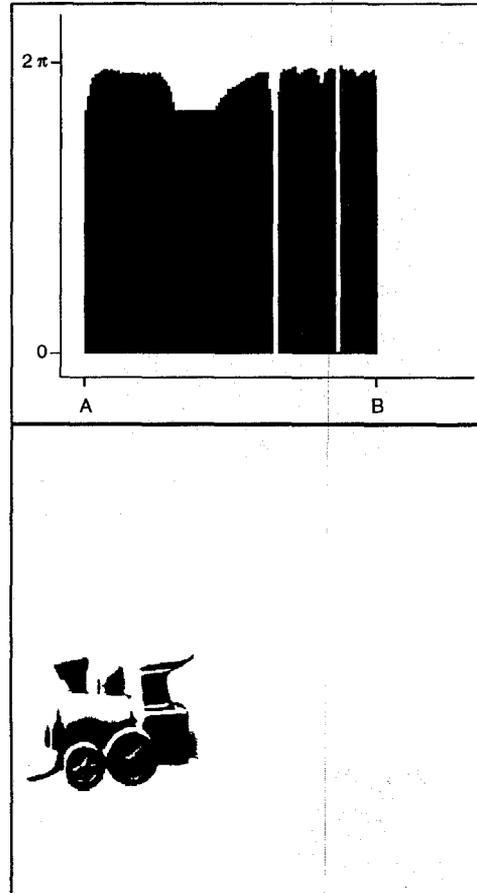


Figure 4: Profile of hue along the line AB painted on image 3 (top), region segmented based on hue values (bottom).

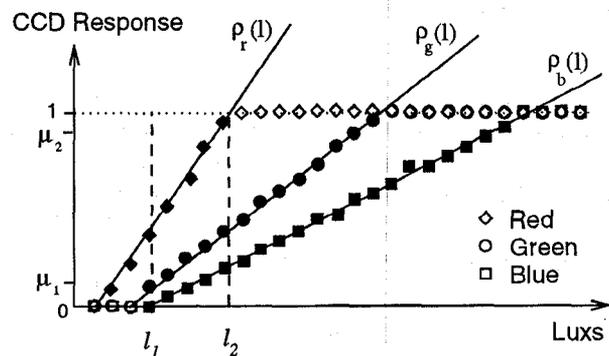


Figure 5: Response of a colour camera for a uniform coloured surface (orange in this case) varying the intensity of the illuminant. Filled symbols represent the linear variation range for each component.

be seen, the hue profile has become more uniform, and the number of pixels which have been detected as part of the engine has been increased.

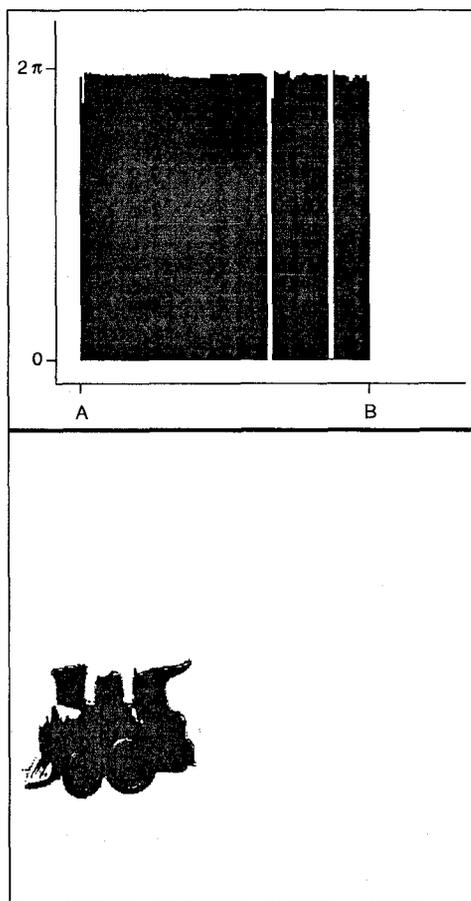


Figure 6: Profile of hue along the line AB painted on image 3 once colour has been corrected (top), region segmented based on corrected hue values (bottom).

The algorithm is easily implemented using LUTs: they will be addressed by the  $(r, g, b)$  values given by the camera. From the LUTs we get the hue and saturation values corresponding to the corrected  $(r', g', b')$  triplet. Previously, LUTs have been loaded with the appropriate hue and saturation values.

#### 4. CONCLUSIONS

We have exposed a simple preprocessing system, but it is a newness that contributes robustness to segmentation systems based on the hue and saturation components, expanding the dynamic range of the camera virtually. The cornerstone of our method is the scale-invariance property of the transformation from

$RGB$  space to  $H$  and  $S$  components of the  $HSI$  space; i.e. colours  $(r, g, b)$  and  $(ar, ag, ab)$  are chromatically equivalent.

Our system can be applied in areas like the tracking of a moving known object, allowing to reduce the effects that changes in the incident light intensity produce on the hue component while the object is moving.

We have tested its functioning in different scenes. As it is expected, with the method presented the final segmentation is greatly improved when the number of pixels which suffers from colour clipping is high.

We are working in a better characterization of the colours set that our system could discriminate simultaneously.

#### 5. REFERENCES

- [1] Q.-T. Luong, "Color in computer vision," in *Handbook of Pattern Recognition and Computer Vision* (L. P. C.H. Chen and P. Wang, eds.), pp. 311–368, World Scientific Publishing Company, 1993.
- [2] R. Gershon, "Aspects of perception and computation in color vision," *Computer Vision, Graphics and Image Processing*, vol. 32, pp. 244–277, 1985.
- [3] C. L. Novak, S. A. Shafer, and R. G. Willson, "Obtaining accurate color images for machine vision research," in *Proceedings of the Conference on Perceiving, Measuring and Using Color*, SPIE, Volume 1250, February 1990.
- [4] S. Shafer, "Using color to separate reflection components," *Color Research and Application*, vol. 10, no. 4, pp. 210–218, 1985.
- [5] F. Perez and C. Koch, "Toward color image segmentation in analog VLSI: Algorithm and hardware," *International Journal of Computer Vision*, vol. 12, pp. 17–42, February 1994.
- [6] G. J. Klinker, S. A. Shafer, and T. Kanade, "The measurement of highlights in color images," *International Journal of Computer Vision*, vol. 2, pp. 7–32, June 1988.
- [7] M. Celenk, "A color clustering technique for image segmentation," *Computer Vision, Graphics and Image Processing*, vol. 52, pp. 145–170, 1990.