

COLOR CONSTANCY IN VISUAL SCENE PERCEPTION

Christian Balkenius
Lund University Cognitive Science
Kungshuset, Lundagård
S-222 22 Lund, Sweden

Anders J Johansson
Electroscience
Lund University

Anna Balkenius
Cell and Organism Biology
Vision Group
Lund University

ABSTRACT

In RoboCup, all objects in the game are color coded to allow easy segmentation and identification. However, color vision is very sensitive to the exact color of the illumination which makes the identification task much harder than is often anticipated. Algorithms that work in the lab in fluorescent light may not function at all in an actual game where incandescent lights are used. This article is intended as a short tutorial on color vision and demonstrates a method for retrieving the color of image patches independently of the color of the illumination. The computational scheme is tested on a set of images of orange balls on a green background with varying illumination.

1 INTRODUCTION

To recognize objects in a video image in real time is a very difficult problem. Even if the objects are known beforehand and there are plenty of computational resources available, it is not yet possible to solve this problem. One way to make object recognition tractable in real time is to color code the objects such that the color of each object indicates its identity. For example, in RoboCup the goals are yellow and light blue, the field is green and the ball is orange. All robots are black with colored markers identifying each team. This allows the identity of each object to be recognized as soon as the color of the object is categorized.

Although color categorization is a much easier task than full object recognition, it is not as easy as is often imagined. Variations in the color of the illumination and the shading of objects will vary the spectral content of the reflected light considerably. A color recognition system that works in the lab in fluorescent light does not necessarily work at a game where incandescent lights are used.

The current rules leave some lighting parameters unspecified which can lead to poor performance of many vision systems. A very exact definition of the lights that could be used during the games has been proposed to overcome this problem (Johansson, Rasmus-Gröhn and Balkenius, 2002).

While it would be possible to devise a more strict control of the illumination in RoboCup, a better solution would undoubtedly be to make the color vision systems of the robots more robust and give them *color constancy*: the ability to recognize colors independently of the color of the illumination.

The human ability for color constancy is so good that we are not usually aware of how much effort our brain must spend before we are able to identify colors (Palmer, 1999). During a day, the light color temperature changes from 2000K in the morning to 8000K during mid day — from reddish dawn to bright white noon. The spectral content also changes when the sunlight passes through the gases in the atmosphere and clouds in the sky. Though the human ability for color constancy is a high level phenomenon that sometimes even depends on an understanding of the visual scene, color constancy is also present in relatively primitive animals such as moths (Kelber, Balkenius and Warrant, 2002) and honeybees (Werner, Menzel and Wehrhahn, 1988). This implies that it should be possible to design a color constancy system with rather limited computational resources.

The goal of the present article is to describe a color constancy system that can be easily implemented in a robot that solves the color categorization problem in an environment like the one in RoboCup. Before the system can be described, however, we need to consider exactly what color is.

1.1 WHAT IS COLOR?

While most of us have been taught in school that color is the wavelength of light, this is only true when we are talking about the color of a monochromatic light source. The color of an object is something much more indirect and intricate (Hardin, 1988).

Let us assume that the scene is illuminated by a light source with a spectral composition described by $I(\lambda)$, where λ ranges over the wavelengths of visible light (approximately 400-700 nm) and I is the intensity of the light at each wavelength¹. We now need to consider how the illuminated light is transformed when it is reflected at a surface patch on an object in the scene. Here, we will make the simplifying (but generally incorrect) assumption that the surface patch is *Lambertian*, that is, it looks equally bright independent of the viewing angle. How the surface patch reflects each wavelength of the illuminating light can be described with the function $S(\lambda)$. This means that the spectral content of the light reaching the eye or camera is given by $I(\lambda)S(\lambda)$. However, this spectral composition is generally not available directly (unless a photospectrometer is used), but is measured by the a number of photoreceptors (in the human eye) or photosensors (in a video camera).

The photoreceptors of the human eye are tuned approximately to red, green and blue light (564, 533 and 437 nm, Dowling, 1987). As a consequence, a similar tuning is also often used in color cameras, for example 650, 530, 470 nm (Kodak, 2002). It is important to realize that the use of red, green and blue has nothing to do with the nature of light, it is simply a way to reduce the infinite dimensional spectrum to three measurements in an arbitrary way that parallels the way our eyes code colors at the receptor level. Some color cameras use a different coding internally, which is subsequently converted to an RGB representation.

The reason RGB coding is used on computer screens is that the three light sources on the screen tap directly into the three receptor types in our eye. When we see the same colors on the computer screen as in the real scene, it is not because they are identical, but a consequence of the limitation of the human color vision system. Other animals have different and sometimes even more types of receptors tuned to different wavelengths and would not see the same colors in the two cases. For example, the stomatopod *Odontodactylus scyllarius* have more than ten different types of photoreceptors tuned to varying wavelengths (Osorio, Marshall and Cronin, 1997).

The reaction of a photoreceptor or output from a sensor in a camera can be modeled in the following way (Mausfeld, 1998). Let i be the specific type of sensor and let $R_i(\lambda)$ be the spectral sensitivity of the sensor.

¹There are several more or less complicated units in which to measure the intensity of light. We will not discuss them here, but see for example Ryer (1997).

The output from the sensor q_i is described by,

$$q_i = \int_{400nm}^{700nm} I(\lambda)R_i(\lambda)S(\lambda)d\lambda.$$

If $i = R, G, B$, the above equation converts the spectrum of the light that reaches the camera to a three dimensional vector $\langle q_R, q_G, q_B \rangle$. How does this vector represent the color or the surface patch? According to the equation above it doesn't since the output from the photosensors depends on the three factors I, S and R , only one of which is related to the surface patch. Only in the ideal case when the illumination is perfectly white, that is when $I(\lambda) = 1$, does the camera give an unique estimation of the color of the surface.

1.2 COLOR CONSTANCY

The main problem of a color vision system is to calculate the vector q_i for each element in the camera image as if $I(\lambda) = 1$. If the spectrum $I(\lambda)$ is known, color constancy can be obtained by dividing the output of each sensor with its sensitivity to the illumination. Let q'_i be the color coordinates after compensation for the illuminant. The new coordinates are calculated as

$$q'_i = \rho_i q_i$$

where ρ_i is the inverse of the response of each receptor to the illumination, that is,

$$\rho_i = \left[\int_{400nm}^{700nm} I(\lambda)R_i(\lambda)d\lambda \right]^{-1}$$

The values ρ_i are called von Kries coefficients (von Kries, 1902, Mausfeld, 1998). By multiplying the color coordinates with these coefficients a partial color constancy is obtained.

There are two main obstacles that makes the above scheme intractable in practice. First, the spectrum of the illumination $I(\lambda)$ is not known. It can only be sensed indirectly from reflections in surfaces. Second, only the approximate spectrum is coded by q_i .

Clearly the color constancy problem is ill posed and cannot be solved without making further assumptions about the scene. Several such assumptions have been suggested in the literature. They are all based on the idea that the scene has some property that can be exploited in the calculation of the correct color.

If it is assumed that the average color of the image is gray, it is possible to scale the sensitivity of each sensor type until this becomes true. This will result in an insensitivity to the color of the illumination. This type of color compensation is often used in automatic white balancing in video cameras.

Another common assumption is that the brightest point in the image has the color of the illumination. This is true when the scene contains specular reflections which have the property that the illuminating light is reflected without being transformed by the surface patch.

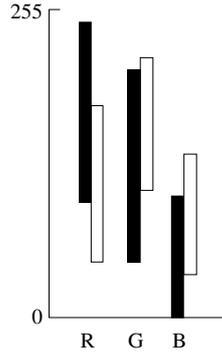


FIGURE 1: The RGB ranges of the orange ball (black) and the green background (white).

By using this information in a von Kries compensation, another form of color constancy can be obtained.

Still, another possibility is to calculate the derivative of the color change over the image since this will discard the illuminant. To get the color back, the brightest point is used as a reference. This is the approach taken by the Retinex theory of color vision (Land, 1977).

Although the reaction to white or gray is often used in color constancy algorithms, it is possible to use knowledge about the color of any surface in the scene to correct the others. If we know that the response to a surface is $\langle q_R, q_G, q_B \rangle$ in white light but it is sensed as $\langle p_R, p_G, p_B \rangle$ in the changed illumination, the von Kries coefficients can be estimated as $\rho_i = q_i/p_i$.

Algorithms that are based on any of these assumptions work well when their underlying assumption is valid. When it is not, however, the result can be far from satisfactory.

1.3 COLOR SPACES

In computer graphics, colors are often represented in the RGB color space, but many different color spaces are possible and the solution to many vision problems depends on the use of an appropriate color representation.

One problem with the RGB representation of color is that it mixes the two properties of color and intensity. It thus makes it hard to distinguish between effects caused by color and light intensity. For example, looking at the individual red, green and blue images one can see shadows and shading in all of them.

Fig. 1 illustrates the problem. Here, the ranges of the RGB values for an orange ball are plotted next to the RGB values for the green background. As can be seen, the two ranges overlap considerably. As a consequence, it is not a good strategy to use raw RGB values to find the orange ball in the scene.

The human visual system solves this problem by transforming the initial RGB-like representation into a more useful form that separates out lightness from

hue and saturation. Color is coded in two dimensions ranging from green to red, and yellow to blue, while lightness is coded in a separate channel (Palmer, 1999, Roberts, 2002).

It is possible to transform the RGB color coordinates into a more useful form without too much effort. A suitable color space is the CIE $L^*a^*b^*$ space that have been designed to more closely parallel the human visual system (CIE, 1978). Although it is not perfect, it avoids many problems of the RGB space.

Given a color coordinate $\langle q_R, q_G, q_B \rangle$, it can be transformed into $L^*a^*b^*$ space in the following two steps. The first step is a linear transformation from RGB space to the CIE XYZ color space (CIE, 1978):

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = M \begin{pmatrix} q_R \\ q_G \\ q_B \end{pmatrix}$$

where,

$$M = \begin{pmatrix} 0.412453 & 0.35758 & 0.180423 \\ 0.212671 & 0.71516 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{pmatrix}$$

Then follows a non-linear transformation to the CIE $L^*a^*b^*$ color space (CIE, 1978):

$$L^* = \begin{cases} 116(\frac{Y}{Y_n})^{1/3} - 16 & \text{for } \frac{Y}{Y_n} > 0.008856 \\ 903.3(\frac{Y}{Y_n}) & \text{otherwise} \end{cases}$$

$$a^* = 500(f(X/X_n) - f(Y/Y_n))$$

$$b^* = 200(f(Y/Y_n) - f(Z/Z_n))$$

where $X_n = 0.950456$, $Y_n = 1.000000$ and $Z_n = 1.088754$ are the values for the white reference.

$$f(t) = \begin{cases} t^{1/3} & \text{for } t > 0.008856 \\ 7.787t + \frac{16}{116} & \text{otherwise} \end{cases}$$

The $L^*a^*b^*$ representation has a number of useful properties: First, it separates the effect of lightness and color. When illuminated with reasonably white light, the shading of an object will appear almost entirely in the L^* channel while the color will appear in the a^* and b^* channels. This allows for independent analysis of shading and color. Second, under the same conditions, soft shadows will only appear in the L^* channel and not in the a^* and b^* channels. This is very useful when shadows need to be recognized or discarded in the image. Third, the coding of color parallels the opponent (red–green and yellow–blue) representation used by the human visual system. Transformations along any of the two dimensions will thus have predictable results. Fourth, the color can easily be represented in a polar

form which allows independent coding of hue φ^* and saturation σ^* ²:

$$\varphi^* = \arctan\left(\frac{a^*}{b^*}\right)$$

$$\sigma^* = \sqrt{(a^*)^2 + (b^*)^2}$$

2 COLOR BASED SEGMENTATION

In this section, we describe an algorithm that can be used to segment a scene based on object color. The algorithm takes as input a single RGB image, with each pixel described by a vector $\langle q_R, q_G, q_B \rangle$, together with a region R of known color and produces a hue-estimation φ^* for each differently colored region in the image (Fig. 2). The algorithm proceeds in the following steps:

Estimation of von Kries coefficients. The first step is to estimate the von Kries coefficients as described in section 1.2. To do this we need to know the sensor responses to a certain reference color in a reference illumination. The reference color is often chosen as white. The reference illumination should preferably be approximately a black body radiator such as the sun or an incandescent light.

The von Kries coefficients $\langle \rho_R, \rho_G, \rho_B \rangle$ are set to the responses to the reference color in the reference image divided by the responses to the reference color in the current image. Since the illumination can be assumed to change slowly, it is not necessary to have a region of known color in every image. Instead, the von Kries coefficients can be estimated at regular intervals or when a region of known color is present in the image.

RGB Correction. In the second step, the von Kries coefficients are used to scale the initial RGB values for each pixel as describes in section 1.2 to the vector $\langle q'_R, q'_G, q'_B \rangle = \langle \rho_R q_R, \rho_G q_G, \rho_B q_B \rangle$.

Conversion to L*a*b* Coordinates. When the RGB coordinates have been scaled, the color coordinates are transformed into L*a*b* color space as described in section 1.3. Since lightness is not used, each pixel is now represented by two values $\langle a^*, b^* \rangle$.

Contour Extraction. Before the average colors of the different objects in the scene can be calculated, it is necessary to find the borders between differently colored areas. This is done using a contour algorithm that works independently in the a^* and b^* spaces.

Since we are interested in contours between different colors, we do not use the lightness channel. This avoids

²Beware that arctan only works in one quadrant. In a C program, use `atan2(a, b)`, rather than `atan(a/b)`, to get the correct result in all quadrants.

many problems associated with finding real contours in the lightness channel. For example, artificial contours will not be introduced around most diffuse shadows.

Several contour algorithms have been proposed that can be used in this stage of the algorithm, for example by Grossberg and Mingolla (1985) and von der Heydt (1995). Here, we use the algorithm described by Månsson (2000). These algorithms have the property that they do not only find edges in the scene, but also extend them to continuous contours. This contrasts with the much used Canny edge detector which does not introduce edge elements without some support in the image (Canny, 1986). When contours have been found in the a^* and b^* channels they are merged into a single contour representation.

Color Averaging. When we know the borders of the elements in the image, the color coordinates $\langle a^*, b^* \rangle$ for all pixels in each area are averaged to give one estimate $\langle \bar{a}^*, \bar{b}^* \rangle$ per region.

Hue Estimation. In the final stage, the hue φ^* of each region is calculated from the \bar{a}^* and \bar{b}^* values (See section 1.3).

3 MATERIALS AND METHODS

The algorithm above was tested on a number of images. A scene was arranged with an orange Ping-Pong ball on a green paper surface similar to the ones used in RoboCup. In addition, a white paper patch was placed in the scene as reference. The scene was lit by a halogen lamp as the only light source. The light was filtered using different color filters (Rosco, Supergel) to simulate the effect of varying illumination. The different filters are listed in Table 1. The effect on the illumination using these filters is much larger than what can be expected at the RoboCup games. The very colored illuminations were used to test the limits of the algorithm rather than to simulate natural variations in illumination.

Images were taken with a digital Canon IXUS V camera at 1600×1200 resolution. The images were scaled to a resolution of 160×120 pixels and aligned in Adobe Photoshop. This allowed the same pre-calculated segmentation of all images to be used for validation of the color constancy algorithm.

The color of the ball in each image was categorized by a human observer in three categories (Table 1): orange (*), different from the background (+), or same as the background (-).

We calculated von Kries coefficients based on either a known white region in the image or the green background.

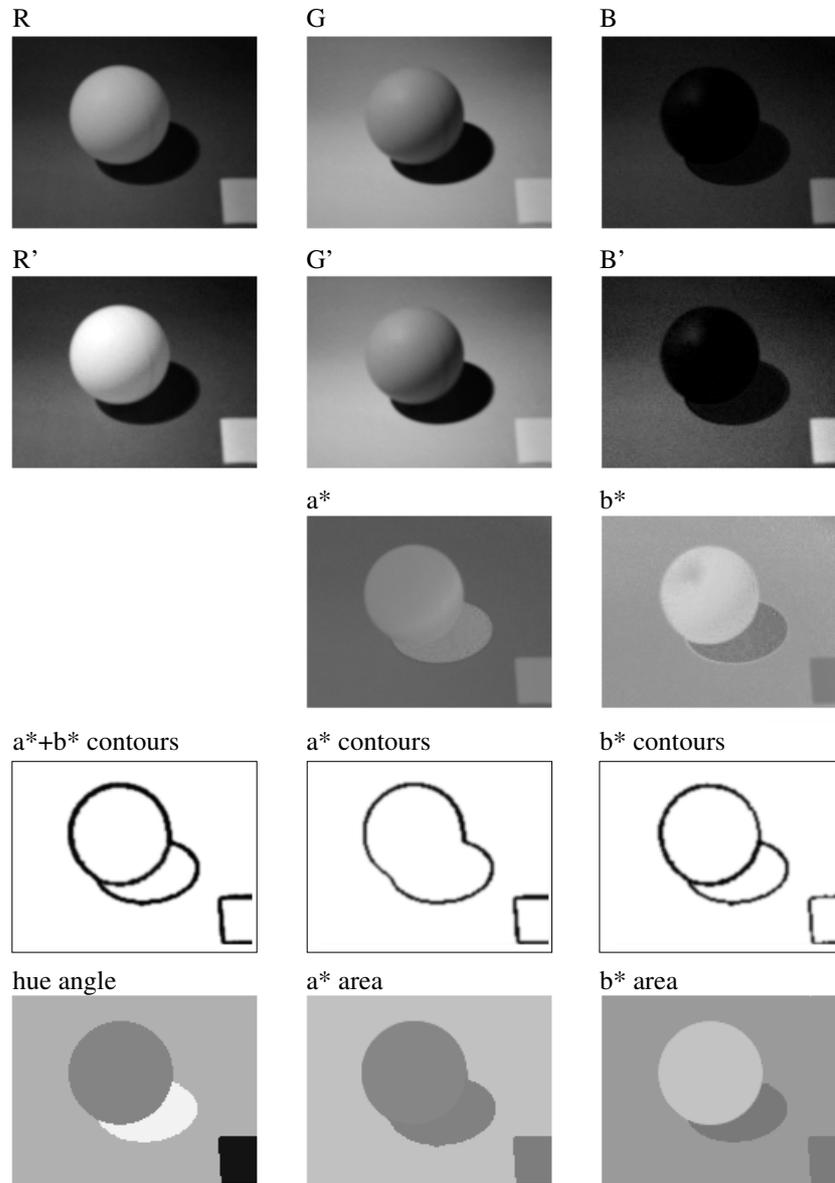


FIGURE 2: Stages of the color processing algorithm. R , G , and B are the original image. R' , G' and B' are the channels after von Kries compensation. The a^* and b^* images shows the red-green and yellow-blue channels. The next row shows the contours in the a^* and b^* channels and their combination. The bottom row shows the a^* and b^* values assigned to each area and the final hue angle φ^* assigned to each area. (The contours are here included in the areas.)

TABLE 1: Overview of the images used in the experiment. Rosco Supergel filter numbers and the colors of the illumination. The human color categorization of the images is given in the column H, and the performance of the algorithm in column A: colors are identified as orange and green (*), colors are identified as different (+), no color difference can be seen (-).

Image	Filter	Color	H	A
1	-	None	*	*
2	01	Light Bastard Amber	*	*
3	05	Rose Tint	*	*
4	11	Light Straw	*	*
5	26	Light Red	-	
6	27	Medium Red	-	
7	46	Magenta	-	
8	52	Light Lavender	*	*
9	66	Cool Blue	*	
10	80	Primary Blue	+	
11	382	Kongo Blue	+	
12	388	Gaslight Green	*	
13	389	Chroma Green	+	
14	90	Dark Yellow Green	+	
15	96	Lime	*	*
16	120	Red Diffusion	-	
17	122	Green Diffusion	-	
18	127	Amber Cyc Silk	-	

TABLE 2: Standard deviation for the normalized RGB values compared to the normalized L*a*b* angles for images marked with * in Table 1.

Reference	R	G	B	ϕ^*
None	0.081	0.136	0.098	0.0199
White	0.045	0.040	0.083	0.0044
Green	0.071	0.028	0.057	0.0069

4 RESULTS

In Table 2, the standard deviations of the RGB values of the ball are shown in the initial images and after RGB correction using either white or green as reference. These values are compared to the hue angles ϕ^* before and after color correction.

The table shows that the standard deviation of the RGB values decreases as an effect of the RGB color correction. However, the main effect results from moving from RGB coordinates to L*a*b* coordinates. For this transformation, the standard deviation is much reduced in all cases. The best result is obtained when the RGB correction with white reference is followed by a transformation to L*a*b* space.

Fig. 3 shows the hue angles before and after compensation for the correctly identified images with white or green as reference color. The graph shows how the hue of the ball is moved toward the actual color. As could

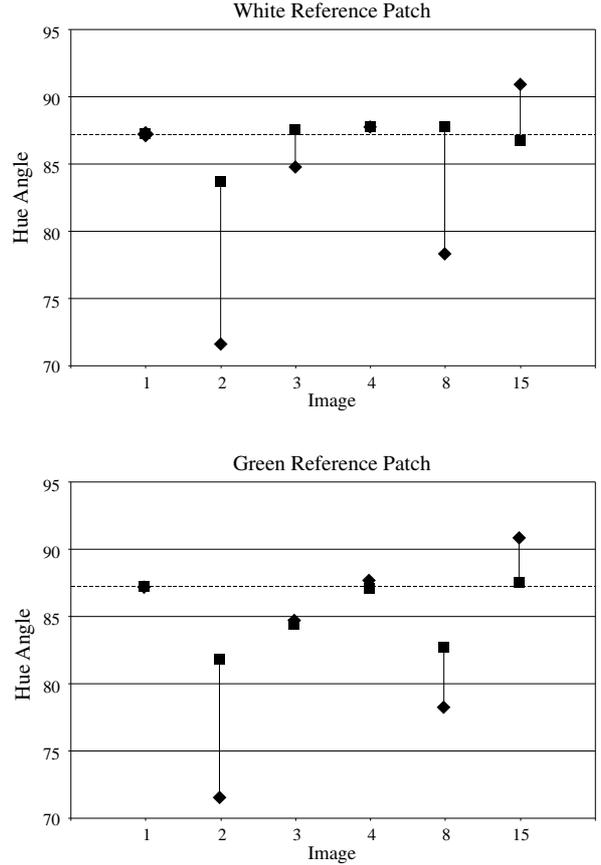


FIGURE 3: Top: The performance for the algorithm for the correctly classified images using the white reference patch. Bottom: The performance for the same images but with green as the reference color. Diamonds: before color correction. Square: after color correction.

be expected, the result is better for the white reference than for the green, but the effect is clearly seen in both cases.

For the majority of the images in which the human eye can identify the colors, the algorithm correctly classifies the color of the ball. In the other two cases, the von Kries adaptation overcompensates the hue angle. These images were taken in a very saturated illumination.

5 DISCUSSION

The proposed method is able to restore the colors in images under varying illumination. The performance is comparable to that of a human observer but fails in a few cases where humans can identify the colors. However, the illuminations used for the test images had a much larger variation than the illumination in RoboCup.

The success of the method depends on three processing stages: RGB correction, area averaging, and coding in L*a*b* coordinates. Of these three stages, the main

effect comes from transforming the image into $L^*a^*b^*$ space, but the other stages also contribute.

The method works best with the white reference patch since it reflects more of the illumination. This will make the estimation of the von Kries coefficients more accurate. The green surface reflects very little in the red region of the spectrum which makes the von Kries coefficient for the red channel less reliable.

The color of the illumination combined with the color of the reference patch also influences the performance of the algorithm. For example, when the illumination is greenish, the color of the background will not change as much as the color of the orange ball. This will make the color correction for the ball inaccurate when the green is used as reference.

In simulation with infinitely narrow tuning of the sensors, the algorithm works perfectly. Since, the response curves for the sensors in the camera differs considerably between manufacturers, the performance of the algorithm is likely to depend on the camera used.

In the images where the algorithm identified that the colors of the ball and background are different, but does not restore these colors correctly, the method overcompensates for the illumination. It appears likely that this overcompensation can be reduced by including additional knowledge of the sensor tuning in the algorithm.

In principle, the algorithm is able to identify the color of shadows and merge them with the background. However, in the test images, the shadows were almost completely black and could thus not be identified as parts of the green background.

With the illuminations used, the required range for the orange color category is less than ten degrees. This would allow 36 different colors to be used for object identification. When less colored light is expected, even more categories could be used. This is substantially more than the number of colors used in RoboCup today.

6 ACKNOWLEDGMENTS

Thanks to Mark Ruzon for the MATLAB implementation of the RGB to $L^*a^*b^*$ transformation. Some of the code used in this article is available as part of the IKAROS project at <http://www.lucs.lu.se/IKAROS/>.

REFERENCES

- Canny, J. F. (1986). A Computational Approach to Edge Detection, *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 8, 769–798.
- Commission Internationale de l’Eclairage (CIE) (1978). *Official recommendation on color spaces, color difference and metric color terms*, CIE-Publication No. 15.
- Dowling, J. E. (1987). *The Retina: An Approachable Part of the Brain*, Cambridge: Harvard University Press.
- Grossberg, S. and Mingolla, E. (1985). Neural Dynamics of Form Perception: Boundary Completion, Illusory Figures and Neon Color Spreading, *Psychological Review*, 92, 263–276.
- Hardin, C. L. (1988). *Color for Philosophers*, Indianapolis/Cambridge: Hackett Publishing Company.
- von der Heydt, R. (1995). Form Analysis in Visual Cortex, In Gazzaniga, M. *The Cognitive Neurosciences*, Cambridge, MA: MIT Press.
- Johansson, A., and Rasmus-Gröhn, K. and Balkenius, C. (2002) Lightning of RoboCup games, *LUCS Minor*, 7.
- Kelber, A., Balkenius, A. and Warrant, E. (2002) Scotopic color vision in nocturnal hawkmoths, *Nature*, 419, 922–925.
- Kodak, (2002). *Device performance specification, Kodak KAC-1310 Image sensor*, Rev 4.
- von Kries, J. (1902) Chromatic adaptation. In MacAdam, D. L. (Ed.) *Sources of color vision*, 109–119, Cambridge, MA: MIT Press.
- Land, E. H. (1977). The Retinex theory of color vision, *Scientific American*, 237, 6, 108–128.
- Månsson, J. (2000). *Occluding Contours: A Computational Model of Suppressive Mechanisms in Human Contour Perception*, Lund University Cognitive Studies, 81.
- Mausfeld, R. (1998). Color Perception: From Grassman Codes to a Dual Code for Object and Illumination Colors, In Backhaus, W. G. K., Kliegl, R. and Werner, J. S. (Eds) *Color Vision*, Berlin: de Gruyter.
- Osorio, D., Marshall, N. J. and Cronin, T. W. (1997) Stomatopod photoreceptor spectral tuning as an adaptation for colour constancy in water. *Vision Research*, 37, 23, 3299–3309.
- Palmer, S. E. (1999). *Vision Science*, Cambridge, MA: MIT Press.
- Roberts, D. (2002). *Signals and Perception: The Fundamental of Human Sensation*, New York: Palgrave.
- Ryer, A. (1997). *Light Measurement Handbook*, Newburyport, MA: International Light.
- Werner, A., Menzel, R and Wehrhahn, C. (1988) Color constancy in the honeybee, *Journal of Neuroscience*, 8, 1, 156-159.