Image Specific Color Representation.

Ido Omer Michael Werman
School of computer science and engineering
The Hebrew University of Jerusalem
91904 Jerusalem, Israel
{idom,werman}@cs.huji.ac.il

Abstract

This paper suggests a new color model for images taken by digital cameras. We show that this color model is useful for overcoming nonlinearities of digital camera sensors. This model is suitable for color image segmentation and enables one to easily manipulate and enhance the colors of an image.

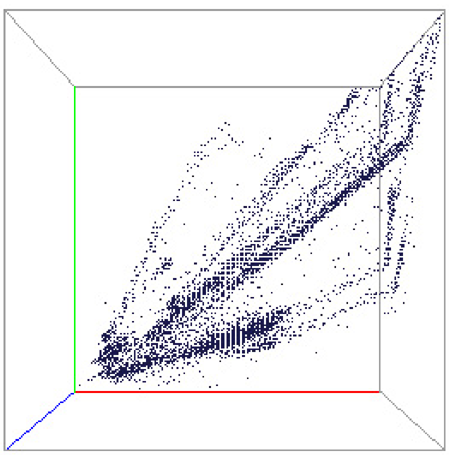


Figure 1. An RGB histogram.

1. Introduction

The problem of color representation affects almost every field in computer vision. Many ways have been suggested for modelling and representing colors. The *RGB* color space is widely used for image capture and display, however it is not considered an appropriate representation for color. This is due to the strong correlation between the three coordinates in the *RGB* color space. Other color spaces have been suggested in order to separate color from intensity and create a more intuitive color representation. The HSV color space and its variants try to decorrelate the information by separating color into hue, saturation and value [2]. The CIELUV and CIELAB color spaces separate color into one luminance coordinate and two color coordinates and try to create a perceptually uniform color system [2, 1]. Attempts have also been made to find application specific color spaces. A good example of this is the $I_1 I_2 I_3$ color space suggested by Ohta et al [13] as most appropriate for segmentation tasks. All these color representations do not take into account the color distortion made by digital cameras and assume perfect color preservation of the camera. Unlike lenses and geometric distortion which have been extensively researched, digital camera color distortion is a topic that still requires a lot of research. Besides sensor response there are other mechanisms in the camera like white balance and color enhancement that effect the RGB values of the image. In this paper we chose to focus on sensor response and design a simple model that yields good results. Until recently digital camera sensors had a very nonlinear response to illumination. The histogram of an image taken by such a camera was usually very noisy and it was hard to identify clear structures in it. The advancement of digital technology has given camera sensors a better and more linear response. When looking at the *RGB* histogram of an image taken by a modern camera it is possible to see structures as can be seen in figure 1.

In this paper we suggest a color model that is image specific. We outline the advantages of this model in image segmentation and present different applications that use such a segmentation. We show that this simple model yields very good results for image segmentation and enables simple and efficient color manipulation.

1.1. Image Capturing

A color image is a function of many parameters, the important of which are: source light color, source geometry(shape), scene and object geometry, object albedo, camera parameters (both geometric and light sensitivity parameters) and image processing done by the camera. Assuming our world consists only of Lambertian objects, we can divide these parameters into two groups. The first group consists of the geometric parameters that have no effect over the wavelength of the light which leaves the object's surface. The geometric parameters only affect the intensity of the light reaching the camera and not its *R, G, B* ratio. The second group includes source light color, the object's surface color the camera sensors' response to light and color enhancement done by the camera, which are the main factors affecting color. Since the first group of parameters does not affect the wavelength (or hue), but only changes the light's intensity, we ignore it for the investigation of color. If we assume we have a camera with linear sensor response and with no color enhancement, all pixels belonging to the same region with homogeneous color lie on a straight line through the origin in the *RGB* histogram. This is in accordance with the laws of colorimetry [4]. This model yields the following formulation which is widely used for describing the *RGB* values of each pixel:

$$\begin{aligned} R &= \int E(\lambda) S_R(\lambda) d\lambda \\ G &= \int E(\lambda) S_G(\lambda) d\lambda \\ B &= \int E(\lambda) S_B(\lambda) d\lambda \end{aligned} \tag{1}$$

Where S_R, S_G and S_B are the sensors response to the incoming radiance $E(\lambda)$, and λ is the light's wavelength.

According to this model the *Nrgb* (Normalized *RGB*) color space is perfect for image segmentation. In practice the *Nrgb* color space doesn't yield good results. Some of the reasons for this are found within the scene itself: specularity, reflectance, several light sources with different colors and inconsistent object's albedo. However, other important reasons lie within the camera itself, these reasons can be divided into camera inaccuracy and color enhancement done by the camera. The focus of this work is modelling the sensors inaccuracy. unlike the rest of the above reasons, we show that this distortion from the simplified model is easy to model, yet once modelled yields results that are significantly better than those of standard color representations.

Camera sensor inaccuracy can be divided into three main types: a) In low intensities the sensors tend to be nonlinear. b) In high intensities the sensors reach saturation. c) The camera samples the world in a noisy manner.
Despite these inaccuracies, modern cameras have an almost linear response to light in a wide range of intensities. In this range, pixels belonging to a region with homogeneous color will align roughly to the same line in the *RGB* color space although this line doesn't necessarily pass through the origin. The reason for this is the sensors' response to light, which can't be modelled using a linear function [5]. Two different phenomena break the linear model: saturation and cutoff. As a rule of thumb, the better the camera, the larger its linear response range and the better it fits this model.
A typical sensor response is shown in figure 2. We would like to mention that by cutoff we refer to the phenomenon described in that figure and in equation 2. An immediate

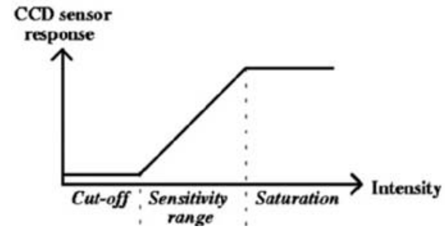


Figure 2. A typical CCD sensor response.

result of this phenomenon is the fact that we can't model the color of a pixel using equation 1 and we should change the equations as follows:

$$\begin{aligned} R &= min(max(\int (E(\lambda) - C_R) S_R(\lambda) d\lambda, 0), Sat_R) \\ G &= min(max(\int (E(\lambda) - C_G) S_G(\lambda) d\lambda, 0), Sat_G) \\ B &= min(max(\int (E(\lambda) - C_B) S_B(\lambda) d\lambda, 0), Sat_B) \end{aligned} \tag{2}$$

Where C_R, C_G, C_B are the cutoff values of the R, G and B sensors and Sat_R, Sat_G, Sat_B are their saturation values. According to the above equations, the color lines should all intersect in one point ($[-C_R, -C_G, -C_B]$). Due to noise and other color distortions the lines don't intersect and in practice we don't require their intersection. Although this modelling is only a rough approximation of the real color distortion (or color enhancement) done by the camera it is more accurate than the commonly used models and gives better results for computer vision and image processing tasks as will be shown later on.

Figure 3 shows two images of the same scene (objects and lighting) taken by two different cameras and the lines best describing their histograms. We see that not only do the lines not intersect the origin, but different lines are produced by different cameras, depending on the cameras sensors and the scene. Even two images from the same camera with different lighting conditions qualitatively differ (mainly due to white balance and other color enhancements done by the camera). As a result, images taken by such cameras don't preserve the laws of colorimetry [4]. We therefore can't create a single representation of color that is accurate for every image or even a color representation that will be accurate for all images taken by the same camera. Instead we should

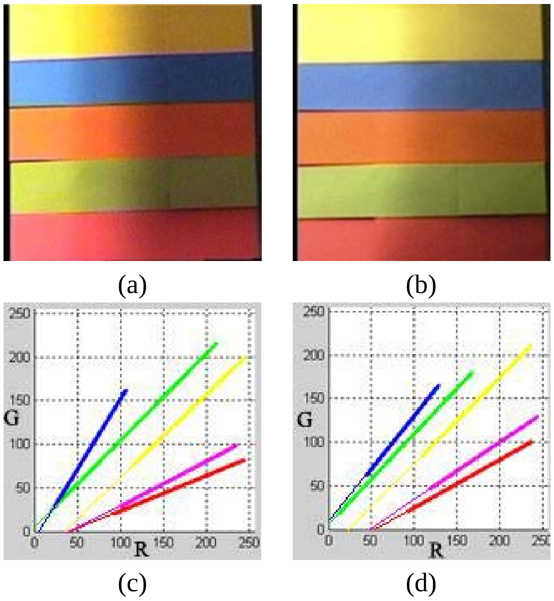


Figure 3. Figure (a) is an image taken with a Sony TRV10E. (b) Is an image of the same scene taken with a Canon Optura. Figures (c) & (d) show 2D views of the best lines describing the histogram of the above images.

analyse each individual image in order to find the color representation that fits it best.

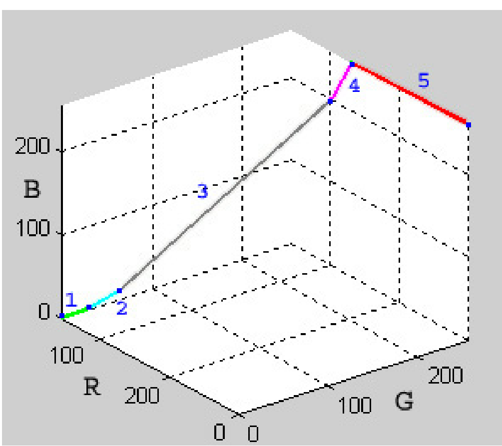


Figure 4. Color line consisting of 5 color segments, 1) only the green sensor response. 2) both green and blue sensors response. 3) all three sensors response, general affine line. 4) the green sensor becomes saturated. 5) both blue and green sensors are saturated.

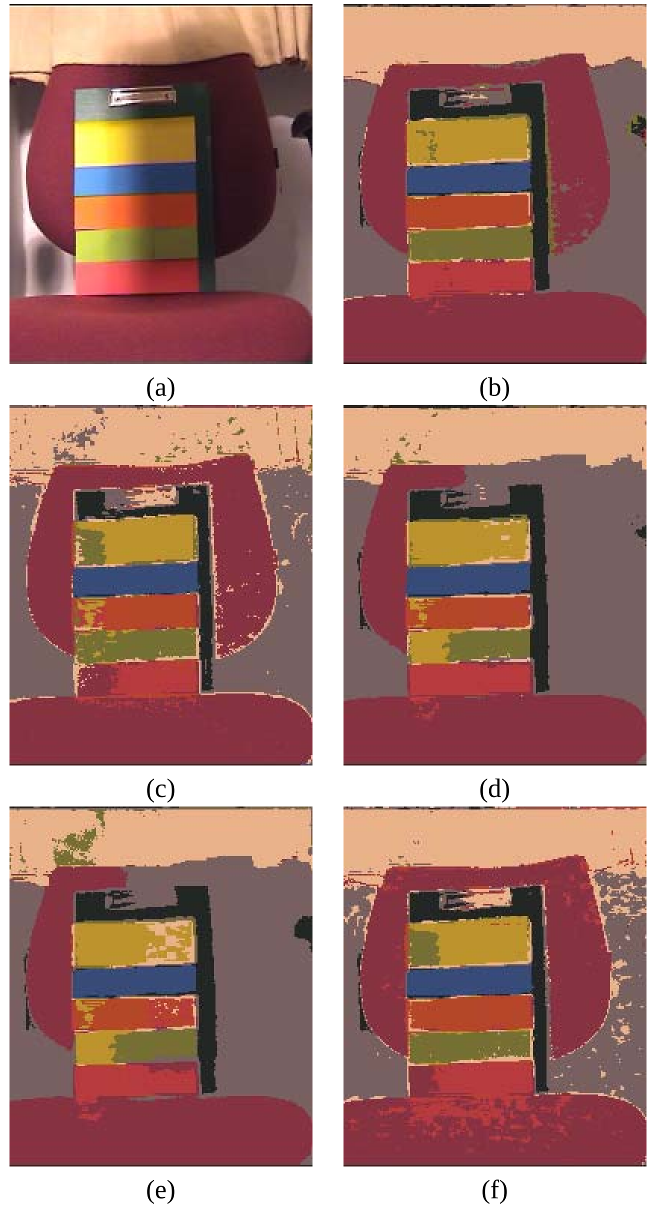


Figure 5. (a) Original image; (b-e) Segmentation according to (b) our color lines model, (c) HSV model, (d) CIE-LAB model, (e) $I_1 I_2 I_3$ model, (f)Nrgb model.

1.2. Modelling Color

According to the physical camera model, color should be modelled using a general affine line in 3D rather than using a line through the origin. Although the actual color distortion in the image can be complicated due to a variety of reasons, we show that the affine color segments model yields results that are signifficatly better than those achivied by standard color models. Moreover standard color models do not consider color distortion and therefore fail in those places our method does. In cases of images with saturated colors, we link two and even more color segments together in order to create a single model that describes the saturated, non saturated and low intensity pixels of the same object which has a single color in the world. We call a group of one or more linked color segments a *color line*. Up to two color segments can be used in order to describe the saturated object's color (once the third sensor becomes saturated, we loose all color information and can no longer recover any color information from the histogram). Theoretically, two more color segments can be used in order to describe the intensities below cutoff values, in practice though, these regions in the histogram are usually very dense and it is difficult to separate the different segments. In their work from 1990, Klinker et al [6] proposed the T model for modelling specularity. Their model introduced affine lines in the *RGB* histogram for the modelling of *specular* image regions. For non specular regions however, their model is linear, while ours is affine as well. Another similarity between the models is the fact they both handles saturated colors but Klinker's model refer to the phenomenon only in the context of specular image regions, while our model handles it in the general case and we suggest ways of overcoming that problem. Figure 4 shows a color line consisting of 5 color segments. We suggest that modelling color as color lines in *RGB* color space is a better color model than other linear or non-linear color models for images taken by digital cameras and that the model is useful for overcoming the digital cameras' inaccuracy.

In order to assess our claim, we show an experiment that compares the color modelling quality of different color representations. For the experiment, we manually segmented an image to its different regions. We then chose the best model for each region's color - in the $I_1 I_2 I_3$, HSV and CIE-LAB color spaces, the model is a point in a 2D color plane. For the *Nrgb* color space it is a line through the origin in 3D (or a point in 2D). For our model it is a color line. We then associated each pixel to its closest model according to the different color representations. In this way we tried to use color information alone for the segmentation task and neglected all other segmenter specific parameters. The results seen in figure 5 show that our color representation modelled the color better than the other color representations. The differences between the results in the experiment are only due to the qualitative differences in the modelling of color between the color representations and not due to any segmentation algorithmic differences. Although this color representation describes color better than the more common representations as shown above, it is still a very simple one. It is very easy to manipulate a pixel's color by manipulating the color line as shown in the following sections.

2. Implementation

Searching the *RGB* color space for the optimal line like clusters is a difficult problem (NPC). Approximations to that problem like the Hough transform and Ransac [8, 9] are also computationally expensive. The large amount of noise in the histogram domain does not make the search any easier. This noise is a result of various reasons, some of which are easy to handle. A good example of this is the areas along edges between different objects, where the camera interpolates the colors of the objects. As a result of this interpolation we have a thin membrane connecting the clusters, creating a planar cluster rather than two separated line like clusters. Either ignoring pixels along edges or cleaning the histogram using filtering can handle this. Specularity is yet another problem that changes the object's color, which can also be handled through special techniques [11]. Another difficulty in finding the best lines describing the histogram is due to the fact that in most images the colors tend to be very grayish and the histogram points are grouped in a small region along the (0,0,0) - (255,255,255) line.

In spite of the above problems, we are able to use natural image properties in order to make this search very feasible. One property that can be used in order to simplify our search, is the fact that we are actually not looking for a general line in the histogram 3D space, but we do have a strong knowledge of the line's orientation. This fact helps our search in two ways: a) it provides us with an easy tool for finding points along the color segments in the image histogram and b) it helps us to match these points to form the actual 3D color segments.

Finding the color model of a given image is performed in

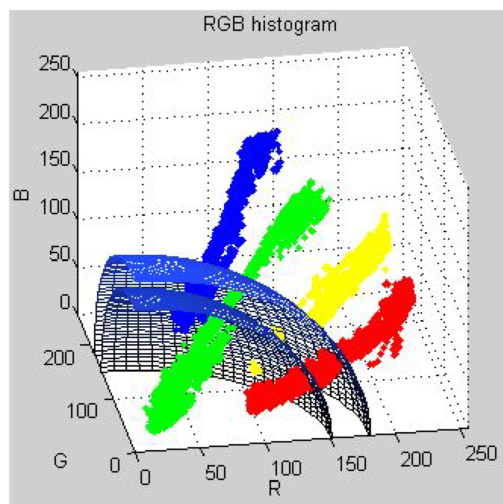


Figure 6. Histogram slicing.

three steps. The first step is creating histogram slices. This is done by first converting the histogram to color space that has three coordinates, one is a distance from the *RGB* origin (or intensity) and the two other represents spatial angle in the *RGB* color space (or color). we then construct each slice S_i as follows:

$$\forall p_{(x,y)} \in S_i \qquad p_{(x,y)} = \sum_{j=\alpha(i)}^{\beta(i)} P_{(j,x,y)}.$$

Where $p_{(x,y)}$ is the point (x, y) in the slice S_i, $P_{(j,x,y)}$ is the 3D histogram point having an intensity coordinate (norm) j and the color (angular) coordinates (x, y) and $\alpha(i)$ and $\beta(i)$ are the intensity summation range of the slice S_i.
The next step is finding the local maxima in these slices. Although our model permits general affine lines, we do expect the general direction of the lines to point towards the origin. as a result, we expect the color lines to intersect the histogram slices in a point (due to the affect of noise, color variety, reflection etc, it doesn't intersect in a point, but in a small region.) and not to be coplanar with these slices. This intersection point forms a local maxima in the histogram slice. We now have for each slice the points in which the color lines intersect the slice. The final step is matching these points to form the color lines. The matching is done in a very simple way. We store a list of color lines and for each new point (a point that hasn't been allocated to a line) we allocate it to the nearest line if it's distance from the line is below a certain threshold or create a new line if it is not close to any of the lines.
Although this approach does not utilize spatial information, it makes strong use of our prior knowledge of the world and in practice gives good results even for very complicated scenes. We would like to mention that in fact, considering the extreme case, when the entire *RGB* cube is considered as one slice, this approach would be identical to performing the segmentation in the 2D Nrgb color space. Nevertheless, there are two main advantages to slicing the histogram: a) it lets us use the affine color segments model which is stronger than the Nrgb model which permits only linear segments, and b) it is less sensitive to noise in the histogram domain since it makes a usage of locality in the histogram domain.
Figure 6 illustrates the histogram slicing used to recover the color model. Figure 7 shows the results of using the histogram slicing approach for recovering the color model of complicated scenes downloaded from the web.

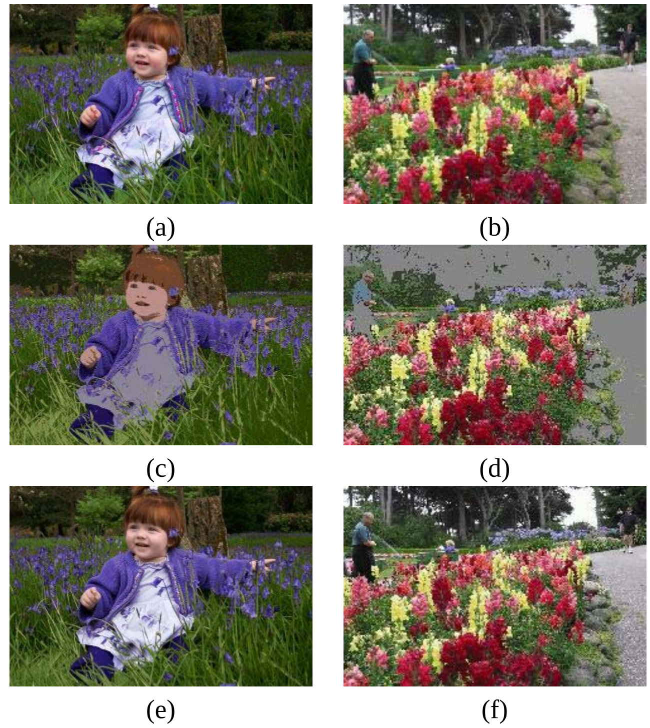


Figure 7. (a) original image downloaded from the Kodak web-site. (b) original image downloaded from the Canon web-site. (c)&(d) the image, segmented to the color segments found, each pixel colored to the middle of the color segment describing its color, the total number of colors is 26 for (c) and 47 for (d). (e)&(f) are the same as in (c)&(d), only this time, the color of each pixel has been assigned to its projection upon the color segment describing its color.

3. Applications

This new proposed color representation approach has several advantages over other common ones: Representing the color as color line gives a model that is strong enough to define color on one hand but is very compact and easy to manipulate on the other. Many applications can utilize this representation for segmentation or various types of color manipulations such as: color correction, fixing saturated color components and even replacing colors of entire objects. We implemented a few examples to demonstrate the strength of this model. We must mention that the implementation of these application is very basic and only comes to demonstrate the utility of our color model.

3.1. Segmentation

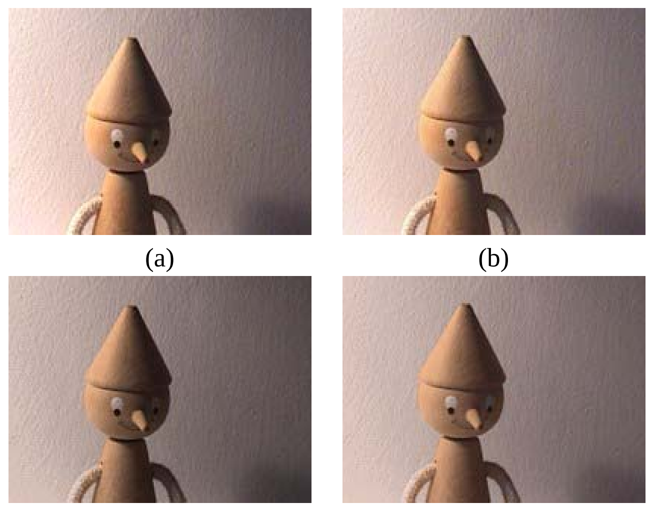
Color representation is a crucial factor for color image segmentation and many attempts have been made to find the best color space for the task, yet no one model has proven to be always superior [12, 7]. We implemented a simple image segmenter that utilizes our color representation. Our segmenter just recovers the color model as described in the implementation section, assign each pixel to the color line closest to it and then remove color lines with less than one percent of the pixels and reassign their pixels. The segmenter does not use any spatial information and is not claimed to be generally superior to any top of the line image segmenter. It only comes to demonstrate the advantages of our color representation and we show it gives very good results using color information alone. We compared the number of color found by our segmenter and the visual segmentation correctness to those of a mean shift image segmenter working in the LUV color space downloaded from [3]. Our segmenter was found to be much less sensitive to luminance changes and yet did not group differently colored pixels. As a result, while our segmenter usually segmented the image into a smaller number of segments than the mean-shift segmenter (in spite the fact that we used the mean-shift segmenter in under-segmentation mode), the segments produced by our segmenter are generally superior as can be seen in figures 11 - 14 (last page).

(a) (b) (c)

Figure 8. (a) Original image. (b) Segmentation according to our color line segments model. (c) Segmentation according to our model, this time, saturated pixels were colored according to their saturation color

3.2. Saturated color correction

Another application of this method is correcting the color of saturated image pixels. The dynamic range of a typical natural or artificial scene is usually larger than the dynamic range that the camera's sensors can capture. As a result, in many pictures some of the pixels have at least one saturated color component (people are often not sensitive to this). In the histogram domain, this phenomenon appears in the form of a knee in the color cluster's line, the line looks as if it has been projected upon the RGB bounding box. By modelling the color as color lines, it is easy to classify saturated pixels and non saturated ones as belonging to the same object as shown by figure 8. This not only achieves a better segmentation by classifying saturated pixels to the correct color line, but also allows us to correct the color of these pixels. We correct the saturated component by substituting one of the non saturated color components in the line equation of the non saturated line segment and retrieving the saturated component. (we can even use one non saturated component to correct the other two). The color correction results is shown in figures 9. This method works when we model the color as lines through the origin as well (or Normalized RGB segmentation) but the results will only be as good as the segmentation and currently the Normalized RGB color model is not an accurate model for images taken by digital cameras. By correcting the saturated pixels we in fact increase the dynamic range of the image, therefore making it unsuitable for direct presentation with typical display devices. In order to readjust the dynamic range we can use gamma correction or other methods for high dynamic range compression [10]. Simply rescaling the color will usually create an image that is significantly darker than the original image and therefore yields poor results.



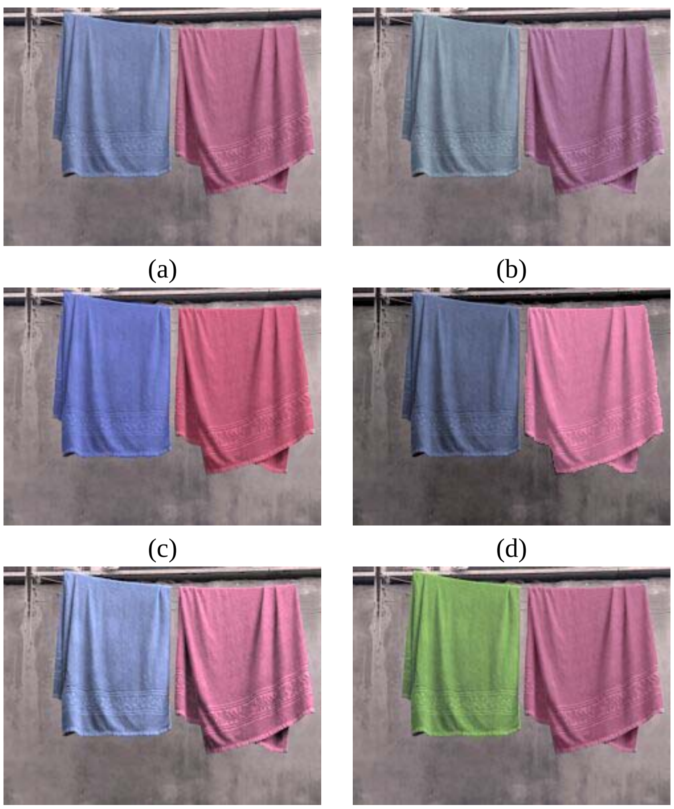
(a) (b)

(c) (d)

Figure 9. (a) Saturated image. (b) Using gamma correction. (c) Correcting saturated pixels and rescaling the colors. (d) Correcting saturated pixels and using gamma correction. It is possible to see that in figures (c) and (d) the saturated (yellowish) pixels in the left part of Pinocchio are corrected but the intensity range has increased from 255 to 305 and the image in (c) is too dark. The intensity in image (d) has been corrected using gamma correction.

3.3. Color editing

Creating a color representation using indices to color lines and norms enables us to manipulate color very efficiently and in a very intuitive way, as can be seen in figure 10. We can increase or decrease the color saturation of an object in an image by moving the object's color line from or towards the central line ([0,0,0] - [255,255,255]). We can increase or decrease object's intensities by moving their colors up or down along their representative color line. We can increase an object's contrast by stretching its color line and we can change an object's color completely and yet preserve all its intensities, and therefore apply changes to big regions in the image at a very low computational cost by moving the whole line.



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

Figure 10. (a) Original image (segmented into 3 color lines). (b) Decreasing the color saturation. (c) Increasing the color saturation. (d) Shifting the intensities making one object brighter and the others darker. (e) Stretching the lines. (f) Changing colors.

4. Summary and Conclusions

We proposed a new way of representing color using a simple physical model of digital image capture. This model is simple, yet rich enough to yield good results in several applications. We use this representation of color for image segmentation and in order to enhance an image's color.

References

[1] *CIE Web site.* http://www.cie.co.at/cie/index.html.

[2] *Color FAQ.* http://www.inforamp.net/∼poynton/ColorFAQ.html.

[3] *Mean Shift color image segmenter.* http://www.caip.rutgers.edu/ meer/RIUL/uploads.html.

[4] M. Chapron. A new chromatic edge detector used for color image segmentation. *IEEE International Conference on Pattern Recognition*, pages 311–314, 1992.

[5] S. Chen and R. Ginosar. Adaptive sensitivity ccd image sensor. *SPIE: CCD and Solid Sate Optical Sensors.*, 2415, 1995.

[6] S. S. G.J. Klinker and T. Kanade. A physical approach to color image understanding. *International Journal of Computer Vision*, 4:7–38, 1990.

[7] Y. S. H.D. Cheng, X.H. Jiang and J. Wang. Color image segmentation: advances and prospects. *Pattern Recognition*, 34:2259–2281, 2001.

[8] P. Hough. Machine analysis of bubble chamber pictures. *International Conference on High Energy Accelerators and Instrumentation*, 1959.

[9] R. C. B. M. A. Fischler. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. of the ACM*, 24:381–395, 1981.

[10] D. L. R. Fattal and M. Werman. Gradient domain high dynamic range compression. *ACM Transactions on Graphics (Proc. ACM SIGGRAPH 2002)*, 2002.

[11] K. Schluns and A. Koschan. Global and local highlight analysis in color images. *CGIP*, 2000.

[12] W. Skarbek and A. Koschan. Colour image segmentation - a survay. Technical report, Technical University of Berlin, Oct. 1994.

[13] T. K. Y, Ohta and T. Sakai. Color information for region segmentation. *Computer Graphics and Image Processing*, 13:222–241, 1980.

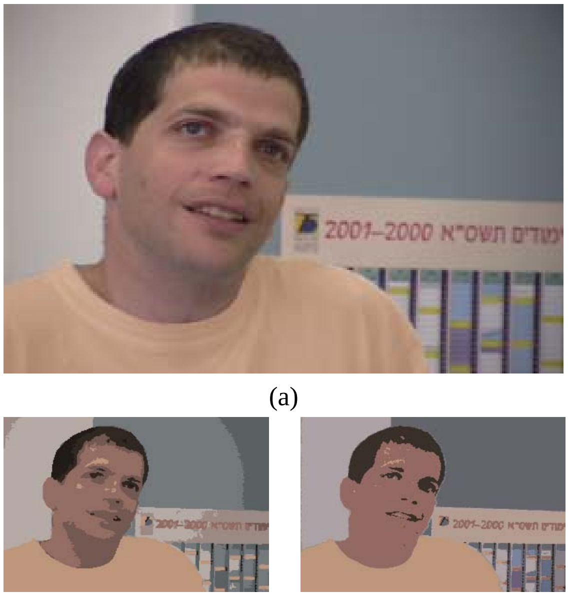


Figure 11. (a) An image captured using a digital camera. (b) Using a mean shift segmenter working in the LUV color space - 9 segments found. (c) Using our color lines segmenter - 10 segments found.

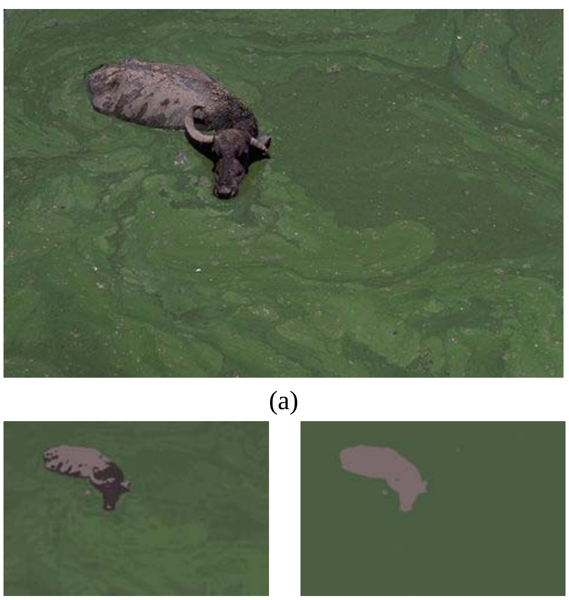


Figure 12. (a) An image downloaded from Berkeley's segmentation dataset. (b) Using the mean shift segmenter - 5 segments found. (c) Using our color lines segmenter - 2 segments found.

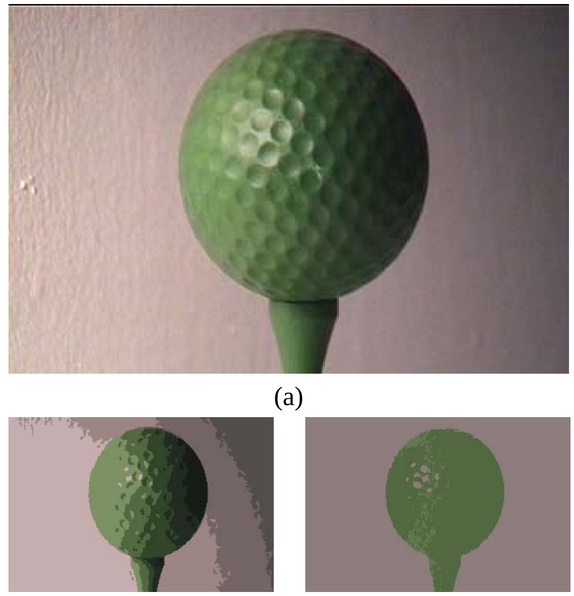


Figure 13. (a) An image captured using a digital camera. (b) Using the mean shift segmenter - 8 segments found. (c) Using our color lines segmenter - 3 segments found (only a minor over segmentation).

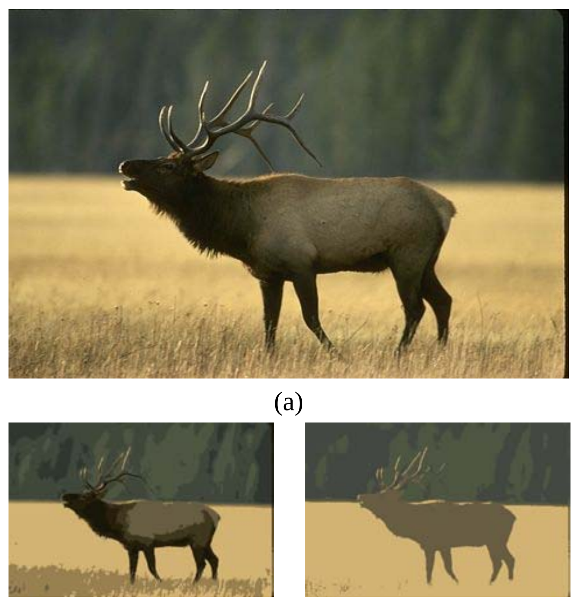


Figure 14. (a) Another image downloaded from Berkeley's segmentation dataset. (b) Using the mean shift segmenter - 9 segments found. (c) Using our color lines segmenter - 4 segments found