Efficient Color Object Segmentation Using the Dichromatic Reflection Model

Vladimir Kravtchenko, James J. Little

The University of British Columbia
Department of Computer Science
201-2366 Main Mall, Vancouver
Canada, V6T 1Z4

E-mail: vk@cs.ubc.ca, little@cs.ubc.ca

Abstract:

The goal of our work is efficient color object segmentation from a sequence of live images for use in real-time applications including object tracking, system navigation, and material inspection. A novel, compact, look-up table color representation of a dielectric object that models the behavior of a color cluster, yields real-time performance, accounts for non-white illumination, shadows, highlights, and variable viewing and camera operating conditions is proposed.

1. Problem

Color has been widely used in machine-based vision systems for tasks such as image segmentation, object recognition and tracking. It offers several significant advantages over geometric cues and gray scale intensity such as computational simplicity, robustness under partial occlusion, rotation in depth, scale changes and resolution changes. Although color based object segmentation methods proved to be efficient in a variety of vision applications, there are several problems associated with these methods, of which color constancy is one of the most important. A few factors that contribute to this problem include illumination changes, shadows and highlights, inter-reflection with other objects, and camera characteristics. Also, speed of segmentation may become an issue in real-time applications where use of computationally expensive operations during the segmentation process may be restrictive. The problem therefore is how to represent the object color robustly and efficiently, and provide means for a high speed segmentation.

2. Color Models

In computers, color pixels usually contain Red, Green and Blue values each measured in 8 bits. A typical color object segmentation would involve a conversion of these values to some color model parameters, then, comparison of these parameters to the assumed object model invariants. The most popular color models used for image segmentation are RGB, HSV, HLS, HSI and NCC.

In addition to the above, the HDI (*Hue, Distance, Intensity*) color model is suggested in this paper and is used for most illustrations herein. The rationale for the suggestion is to preserve the geometrical distances in units used to measure R, G and B values and avoid a problem inherent to some of the above color models in describing saturated pixels. The subspace within which the model is defined is the RGB cube as shown in Figure 1.

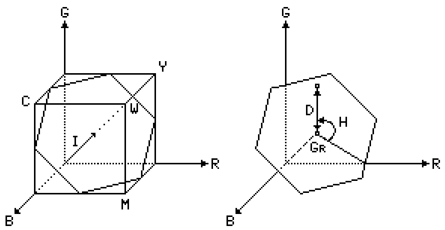

Figure 1. The HDI color model.

Hue is the same as in the HSV model. Intensity is defined as $I = (R + G + B)/\sqrt{3}$. This is the true unscaled distance along the gray diagonal from the black corner of the cube to the plane in which the RGB point lies. This plane of constant intensity is perpendicular to the gray diagonal. Distance is the distance from the RGB point to the gray diagonal. Distance is not a ratio. Distance and Intensity are measured in the same units as are R, G, and B.

3. Pixel Distribution in Real Settings

To demonstrate how image pixels of a dielectric object are distributed in color space, several experiments in real settings were performed. The following equipment was utilized: a Sony EVI-D30 CCD color camera providing an NTSC signal via S-Video port, a Matrox Meteor NTSC frame grabber, and a PC with an Intel Pentium II - 233 MHz processor running under Linux. Lighting was of two types: indoor fluorescent illumination and an incandescent desk lamp.

In our first experiment a red ball made of a soft plastic was exposed to the indoor fluorescent illumination. The camera operating parameters (iris, shutter and gain) were automatically adjusted by the camera. The HDI distribution of pixels from the red ball is shown in Figure 2.

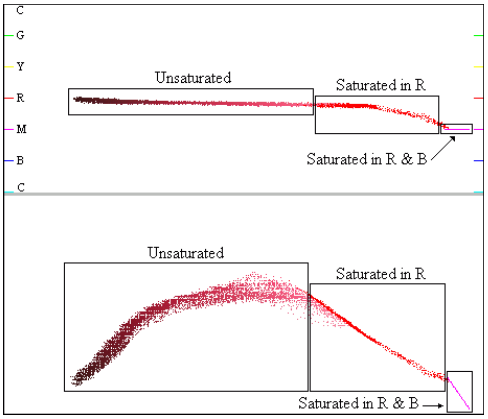

Figure 2. Red ball pixel distribution at automatic exposure.

The top plot is H vs I. The bottom plot is D vs I.

The cluster in this figure is formed from three distinct parts: unsaturated pixels that lie inside the RGB cube, pixels saturated in one color band that lie on the side of the RGB cube, and pixels saturated in two color bands that lie on the edge of the RGB cube. The saturated pixels came mostly from the area of a bright specular reflection on the ball. One intuitively sees that the cluster lies more or less in the plane. The top plot would correspond to the edge view of this plane, and the bottom plot would correspond to the view perpendicular to this plane.

4. The Dichromatic Reflection Model

The use of the *Dichromatic Reflection Model* suggested by S. Shafer [6] helps solve a problem associated with the illumination changes, namely, shadows and highlights, and also takes into account the color (spectral distribution) of the illuminant. Although problems with object inter-reflections and spectral changes of the illuminant still have to be addressed, the implemented segmentation system using this model showed more robustness and higher speed compared to either the popular color model or to pixel clustering approaches.

The dichromatic reflection model describes the light which is reflected from a point on a dielectric, nonuniform material as a mixture of the light reflected at the material surface and the light reflected from the material body. The first, the surface reflection component, generally has the same spectral power distribution as the illumination and appears as a highlight on the object [2]. The second, the body reflection component, provides the characteristic object color and exhibits the properties of object shading. Thus, according to the dichromatic reflection model, all pixels belonging to a color object may be described as a linear combination of two vectors: the *surface reflection vector* $C_S(\lambda)$ and the *body reflection vector* $C_B(\lambda)$, as seen in Figure 3.

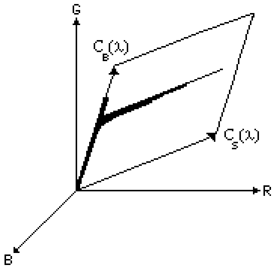

Figure 3. A dichromatic plane in RGB space.

Highlight pixels form a *highlight line* in the direction of the surface reflection vector. Matte pixels form a *matte line* in the direction of the body reflection vector.

With a three-dimensional RGB color space, the body reflection and surface reflection vectors span a *dichromatic plane* containing a parallelogram defined by these vectors [5]. The colors of all rays reflected from the dielectric object then form a planar *color cluster* that lies within the parallelogram. This is shown in Figure 3. The clusters are shaped as a skewed "T" or a skewed "L". The geometry of the clusters may vary and is defined by the illumination and viewing conditions, as well as by the characteristics of the camera [1].

To verify the dichromatic reflection model, and to determine how the distribution of pixels changes with the illumination of various colors, a second experiment was undertaken. In this experiment a desk lamp was used to which red, green, and blue Kodak color filters were applied. These filters had the same transparency (magnitude) but different colors. The distribution of pixels of the red ball at fixed camera exposure and using various filters is shown in Figure 4.

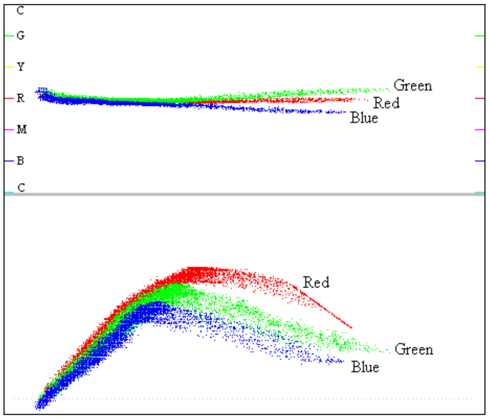

Figure 4. Filter response of the red ball.

The red ball illuminated with red light does not show any changes in Hue. In this case, the body reflection color vector and the surface reflection color vector are co-planar with the gray diagonal of the RGB cube, resulting in the horizontal Hue line. With the green color filter, the tendency of Hue to become green when Intensity increases can be observed. With the blue color filter, the tendency of Hue to become blue when Intensity increases can be observed. The Hue curves for green and blue filters are inclined respectively. The bottom plot of Figure 4 demonstrates the decreasing spectral responsitivety of the camera to short wavelengths so that red is sensed brighter than blue.

5. Camera Limitations

The limitations inherent in many digital cameras introduce distortion to the linearity of the dichromatic reflection model. A brief review of these limitations and their effect on the color cluster follows. (1) A *limited dynamic range* causes the color cluster to bend along the sides and edges of the RGB cube. This process is also referred to as *color clipping*. (2) *Blooming* causes some pixels of the color cluster to shift from their true locations in one or more distinct directions. (3) *Gamma-Correction* transforms the dichromatic plane into a non-planar dichromatic surface [2] and causes a non-linear deformation of the color cluster. (4) *Noise*, a common problem inherent to electronic devices, affects pixel values and causes them to fluctuate within a certain range. According to the specification, the color camera we used had a S/N ratio of 48 dB. The frame grabber had on the average 1.5 bad bits out of 8 with the least significant bit reflecting more noise than data. The fluctuations are not always visible to a human eye, but we have to take them into account in the context of the considered dichromatic reflection model. In theory, noise introduces thickness of the dichromatic plane, and, in practice, it introduces thickness of the dichromatic surface. (5) Other limitations that distort pixel distribution in the color space include but are not limited to: *Sensitivity to the Infrared Light*, varied *Spectral Responsitivety*, *Spatial Averaging* and *Chromatic Aberration* [3,4].

6. Camera Operating Parameters

Automatic exposure is a common feature in many modern cameras. It automatically controls *iris* (aperture), *shutter* (exposure) and *gain* to optimally adjust brightness of the picture depending on the illumination conditions and brightness of the seen objects. The camera was tested to see how these adjustments affect the distribution of pixels in the color space. In the third set of experiments the red ball was exposed to indoor fluorescent illumination. Firstly, the shutter was manually fixed at 60 Hz, gain at 9 dB, and the iris was changed from F2.0 to F4.8 (from close to open). Secondly, the iris was manually fixed at F1.8, gain at 6 dB, and the shutter was changed from 60 to 500 Hz. In the third experiment the iris was manually fixed at F2, shutter at 60 Hz, and the gain was changed from -3 to 12 dB. In all these experiments the red ball pixel distributions complied to the dichromatic reflection model. It was observed that the color cluster of the red ball fluctuated within the dichromatic surface.

7. Approximating the Dichromatic Surface

In previous sections the factors that influence the distribution of pixels in the color space were discussed. From this discussion and the experiments performed it is known that a dichromatic plane is transformed into a non-planar dichromatic surface. The problem is how to approximate this surface. One way that may be suggested is statistical. This involves sampling the color clusters formed by an object viewed at various illumination conditions (same spectra but different intensity), and viewing conditions and accumulating the sampled data in an array. Alternatively, a principal components analysis might be applied to the color clusters of the same object at various illuminations and various viewing conditions to get eigenvectors. The cross product of the two biggest eigenvectors will determine the normal of the plane approximating the dichromatic surface. A third, more simple approximating technique of fitting two planar slices to the dichromatic surface is suggested here and tried to see if satisfactory results could be achieved.

The color cluster shown in Figure 5 is typical for the red ball used in our experiments as viewed by the camera with automatic exposure at the indoor fluorescent illumination.

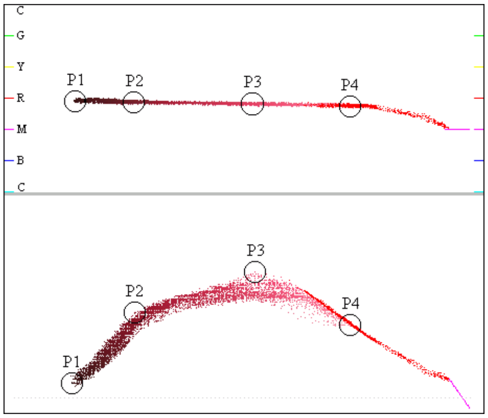

Figure 5. Reference points on the color cluster.

Initially three points were selected from the unsaturated pixels of the color cluster. In the HDI model two points correspond to the following: P1 is the point with minimum Intensity, P4 is the point with maximum Intensity. P3 is the furthest point from the P1-P4 line. In a rough approximation, P1-P3 corresponds to the body reflection vector, and P3-P4 corresponds to the surface reflection vector. A plane was fitted using points P1-P3-P4. However, many pixels in the area between points P1 and P3 turned out to be off the plane. Then point P2 was introduced as the furthest unsaturated pixel from the P1-P3 line. The second plane was fitted using points P1-P2-P3. This considerably increased the recognition of the object pixels. In fact, often a specular spot occupies a relatively small area of the object surface, and fitting a separate plane to the planar gamma-curve that corresponds to the matte line will cover the matte pixel majority with higher accuracy.

The thickness (Th) of the fitted planes was set according to the average deviation (Ea) of unsaturated pixels from the assigned regions, P1-P2-P3 for the body reflection plane, and P1-P3-P4 for the surface reflection plane respectively. The thickness was chosen to be Th = Ea * 2. The thickness is in the same measurement units as R, G, B in the RGB model and D, I in the HDI model. In our experiment the thicknesses were 6.4 and 6.8 for P1-P2-P3 and P1-P2-P3 planes respectively. The angle between planes was 13.3°.

In theory, the color cluster lies in a restricted area, to be exact, within the parallelogram defined by the body and surface reflection vectors. In practice, the area within the dichromatic surface must be restricted. For that purpose directions of P1-P2 and P3-P4 lines as shown in Figure 6 were used.

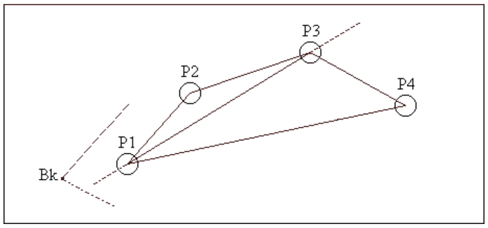

Figure 6. Area restriction of the dichromatic surface.

For plane P1-P2-P3 we calculated two perpendicular planes, the first running through P1-P3, the second running through the black point of the RGB cube parallel to P1-P2 line. A slice of the P1-P2-P3 plane restricted by the two perpendicular planes and the surface of the RGB cube was used for further consideration. By analogy, for the P1-P3-P4 plane two perpendicular planes were calculated, the first running through P1-P3, the second running through the black point of the RGB cube parallel to P3-P4 line. A slice of the P1-P3-P4 plane restricted by the two perpendicular planes and the surface of the RGB cube was used for further consideration.

It is hard to restrict the side farthest from the black point because the varying camera operating parameters may force the border to be pushed to the surface of the RGB cube. For this reason we let this surface represent the border.

8. A Look-Up Table Representation of the Object Color

As mentioned before, in color digital images, pixel values usually contain Red, Green and Blue values each measured in 8 bits. A typical color object segmentation would involve a conversion of these values to some color model parameters, then comparison of these parameters to the assumed object model invariants, with Hue being one of the most common. The disadvantages of this approach are: a slowdown due to an intensive use of computationally expensive operations, such as division, square root and trigonometric functions for every analyzed pixel, and, in general, an imprecise representation of the object color at various viewing conditions and at non-white illumination. An ideal way of segmenting a color object in the image would be to interpret a pixel value as a 24-bit (or less, depending on the desired color resolution) number, and to use this number as an index in a *Look-Up Table (LUT)*, the entries of which tell whether or not the analyzed pixel belongs to the object. Such a check would be extremely fast, yielding real-time performance. Further, a method for creating such an LUT using the dichromatic reflection model is suggested.

To create a Look-Up Table for fast color object segmentation the RGB color model was used. Consider a cube consisting of zeros and ones, where ones represent pixels that belong to the object and lie on the dichromatic surface, or, in this case, on the planar slices approximating the surface. Since color resolution of disaturated colors near the gray diagonal of the RGB cube is low, a safe volume determined by a cylinder with the axis being the gray diagonal is used. Pixels inside a cylinder volume are disregarded. The diameter of the cylinder was chosen to be 10. With these analytical descriptions in mind - the two planar slices, and the safe cylinder - ones are turned where the dichromatic surface runs, and zeros are left in the rest volume of the RGB cube.

With 8 *bits per color (bpc)* and three colors (RGB) one may obtain 16,777,216 various colors. Camera noise was measured by subtracting neighboring frames of the same image with the result that the lower 2 bits of the RGB byte values are in constant flux. When auto focus and auto exposure features were activated, noise increased. To represent color resolution, 6 bpc was selected. With 6 bpc, 262,144 various colors may be obtained. To represent this number of colors, 262,144 entries are needed in the LUT. While 262,144 bytes may be used, this however would be a waste of space since only one bit of a byte entry will actually be used. The LUT may be compressed into 262,144 bits (32,768 bytes), thus obtaining a very small LUT that may be used in applications where the amount of memory is a concern.

To reduce color resolution a simple algorithm shown in Figure 7 is suggested. In the initial pixel, three out of four bytes represent RGB values (the fourth upper byte is often zero and unused). The reason for using four bytes per pixel instead of three is that the packs of four bytes are processed by a CPU faster than the packs of three. We split the four byte integer representing a pixel into three integers containing only R, G, and B values. Color reduction is performed as a simple register bit shift operation. The remaining R, G, and B values are then aligned and combined back into a single integer. The resulting integer may be used to access a particular entry in the LUT.

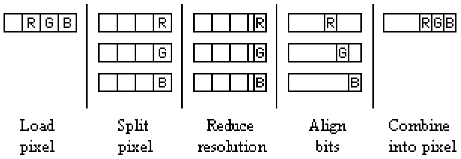

Figure 7. Reduction of color resolution.

To access the bit entry in the LUT that defines whether or not a pixel belongs to the object another simple algorithm is suggested. The pixel value, after reducing color resolution, is divided by 8 (by a bit shift to the right). The integer part of the division tells which byte of the LUT to check. The modula part of the division (obtained by a bit mask) tells which bit of the byte to check.

The main advantages of using the compressed LUT approach are simplicity, speed and the small amount of memory required. All operations used, except memory access, are simple one CPU clock cycle operations. There are no divisions, square roots, trigonometric functions or other computationally expensive operations.

9. Results

The results of segmenting a color object from a sequence of live images using the suggested approach are very satisfactory. With an LUT occupying only 32 Kb of memory (6 bpc) and an RGB image of 320 x 240 pixels an extremely high speed segmentation was achieved. It takes 8 ms to segment the image using the suggested approach compared to 80 ms segmentation based on hue. Calculating hue for every pixel involves at least one integer division which significantly slows down the segmentation process. In NTSC a new frame comes every 33 ms (30 frames per second) and to stay real-time, without dropping frames, one should complete segmentation within a 33 ms limit. This has been achieved using the suggested approach.

The segmentation also accounts for a non-white illuminant, shadows and highlights. The achieved quality of segmentation is also higher compared to the hue-based segmentation which assumes object color to be of a uniform hue. A non-white illuminant and a specular surface cause a hue range to increase to accommodate areas of color space that represent the highlight. However, the side effect of such an increase is inclusion of areas of the color space that may belong to other objects, thus introducing additional noise and causing error in object recognition.

Satisfactory results were also obtained when the suggested approach was applied to model the color of human skin. Multiple objects of skin color were segmented out with acceptable quality in real time.

10. For Further Investigation

Although the results are very satisfactory, several things can be improved while a few problems need to be addressed. (1) The dichromatic reflection model contains several assumptions that are both restricting and simplifying. These include: a single spectrum of body reflection, a single spectrum of surface reflection, and light sources of a single spectra. Ambient light and object inter-reflections were not taken into consideration. However, for many applications, the dichromatic reflection model still provides a reasonable and useful description of the physics of light reflection. (2) Approximation of the dichromatic surface can be improved in several ways.
(3) Additional cues as intensity, texture, or geometry may improve accuracy of object recognition.

11. Conclusion

In our work, an effort was made to develop an approach to efficient color object segmentation from a sequence of live images for use in real-time applications. A novel, compact, look-up table color representation of a dielectric object that modeled the behavior of a color cluster, yielded real-time performance and accounted for non-white illuminant, shadows, variable viewing conditions and camera operating parameters was proposed. Further development based on this approach may result in more efficient color representation of various materials and multi-colored objects.

References:

[1] G.J Klinker, A physical approach to Color Image Understanding, PhD dissertation, Carnegie Mellon University, Pittsburgh, Pa, 1988

[2] G.J. Klinker, S.A. Shafer, T. Kanade, The Measurements of Highlights in Color Images, Int. J. Comp. Vision 2(1), pp. 7-32, 1988

[3] C.L Novak and S.A. Shafer, Color Vision. In Encyclopedia of Artificial Intelligence, pp. 192-202, 1992

[4] C.L Novak, S.A. Shafer, R.G Willson, Obtaining Accurate Color Images for Machine Vision Research, Proc. SPIE 1250, pp. 54-68, 1990

[5] S.A. Shafer, Describing light mixtures through linear algebra, J. Opt. Soc. Amer. 72(2), pp. 299-300, 1982

[6] S.A. Shafer, Using Color to Separate Reflection Components, Color Res. App. 10(4), pp. 210-218, 1985