6-1-2005

A New Color Representation for Non-White Illumination Conditions

Jae Byung Park

Avinash C. Kak

A NEW COLOR REPRESENTATION FOR NON-WHITE ILLUMINATION CONDITIONS

JAE BYUNG PARK
AVINASH C. KAK

TR-ECE – 05-06
JUNE 2005

PURDUE UNIVERSITY

SCHOOL OF ELECTRICAL
AND COMPUTER ENGINEERING
PURDUE UNIVERSITY
WEST LAFAYETTE, IN 47907-2035

A New Color Representation for Non-White Illumination Conditions

Jae Byung Park, *Student Member, IEEE*, and Avinash C. Kak*

Abstract

Color is an image attribute that has been used extensively in many areas of computer vision such as image segmentation, object recognition, and object tracking. Color values, however, heavily depend on both the illumination intensity and the color of the illuminant. Variations in illumination intensity create shading effects on object surfaces that need to be discounted. Moreover, with non-white illumination, object surfaces exhibit illumination-induced color changes. It therefore becomes important to use color descriptors that are maximally independent of not only the variations of illumination intensity but also the color content of the illumination. Although the *RGB* color space is one of the most commonly used color representations, it does not provide illumination-invariance. Other color spaces such as the *HSI* and the normalized *RGB* provide simple but ineffective mechanisms to cope with variations of illumination. To the best of our knowledge, a color descriptor that is completely free from illumination effects has not yet been reported in the literature.

In this paper, we propose a transformation technique that adapts the color space to the color of the illuminant and leads to a color representation that is more independent of illumination than any existing approaches. This color space transformation extends the well-known *RGB*-to-*HSI* transformation to the case of non-while illumination in such a way that the saturation is measured as radial distance from the illumination vector and the hue as the polar angle around the same vector. When a color space is constructed in this manner, it becomes possible to characterize object color with *illumination-adapted hue* and *illumination-adapted saturation*. Another benefit of the new color space is that the dichromatic plane now acquires a single-parameter characterization. Our experimental results on color constancy, color image segmentation, and color object tracking establish the usefulness of the new approach to color representation.

*The authors are with the School of Electrical & Computer Engineering, Purdue University, West Lafayette, IN. 47907
E-mail:{jbpark, kak}@purdue.edu

1. Introduction

Many computer vision algorithms are based on the assumption that the color of an object surface is a constant property of the object. However, when images are recorded with non-white illumination, the perceived image of an object is affected by the color of the illuminant. Also, if the intensity of the illuminant varies over time or if there exist shadows due to occlusions and/or mutual reflection effects in a scene, the color data recorded for an object surface will be compromised. In order to cope with such variations in illumination, computer vision algorithms try to use color descriptors that are approximately invariant to illumination changes – at least to small changes in the intensity of local illumination. For example, Funt and Finlayson [1] and Nayar and Bolle [2] have used what they refer to as illumination-independent color ratios as measures of local color. These color descriptors are designed for object recognition tasks where the usage of the color descriptors is limited to the comparisons between a pair of images. These approaches run shorts when one of images is contaminated by shading or highlights [3].

Using the dichromatic reflection model, Gever and Smeulders [3] demonstrated that normalized RGB and the H (Hue) and the S (Saturation) color descriptors can be substantially invariant to changes in camera view point, object pose, and the direction and the intensity of the incident light. However, these color descriptors, although simple to use, are usually not robust under non-white illumination conditions. The reason for this limitation are not surprising since these descriptors were designed primarily with white illumination in mind. We believe that if our goal is to measure object surface colors under a wide variation in illumination colors, the color descriptors used must explicitly take into account the color of the illuminant itself. In other words, the color descriptors used must adapt to the color of the illuminant if they are to perform equally well under a wide range of illumination colors. The main goal of this paper is to propose a new color representation that permits such adaptation.

For a quick demonstration of why a commonly used color descriptor – normalized RGB – that is meant to achieve a modicum of illumination invariance does not work under non-white illumination, shown plotted in Figure 1 (a) are two sets of data in an RGB cube, both taken for a yellow ball but with two different illuminations. The red points correspond to the color values in the image that was recorded under normal fluorescent illumination and the blue points correspond to the case when the

image was recorded with mostly blue illumination. Shown in Figure 1 (b) are the normalized versions of the red and the green components of the color values shown in (a). The very strong illumination dependency of the normalized values is obvious from these figures. We also want to use Figure 1 (a) to point out that the color values coming off a scene are strongly aligned with the color of the illumination.[1] Under perfect white illumination, these values would be strongly aligned with the gray axis shown in Figure 1 (a). To the extent the red points shown in (a) depart away from the gray axis, we can say the color of illumination produced by fluorescent lighting is not perfect white. Figure 1 (a) also shows the strong alignment of the data collected with blue illumination nearly aligned with the color vector corresponding to that illumination. In the rest of this paper, the vector in the RGB cube that predominantly defines the color of the illumination will be referred to as the *illumination axis*. The distribution of the color values for the yellow ball under the blue lighting (shown in Figure 1 (a)) will fall mostly along the illumination axis in the RGB cube.

In order to overcome the illumination dependency of the existing color representations and the color descriptors derived from such representations, we will propose a new color space transformation technique in this paper. By estimating the color of the illuminant, our new color representation uses for one of its axes the illumination axis. The new color representation scheme we present rotates the conventional RGB color cube in such a way that the illumination axis itself becomes the vertical axis (Z-axis) in a Cartesian coordinate frame. As to the other two axes, their directions are determined primarily by an RGB to *HSI-like* transformation we carry out prior to aligning the illuminant axis with the Z-axis.

Our new color space lends itself well to a reduced dimensionality representation of color when so needed for computational purposes. As a result, it gives us a more efficient way to analyze color images of dielectric objects under non-white illumination conditions. A series of experiments we present in Section 5 demonstrate the feasibility of the proposed color representation and its applicability to computer vision, especially when images are recorded under non-white illumination.

This paper is structured as follows: In **Section 2**, we briefly discuss well-known color spaces and

[1] This obviously assumes that the object surfaces return a combination of diffuse and specular light. While diffuse component will correspond to the true color of the surfaces, the specular component will correspond to the illumination. As predicted by the dichromatic reflection model, the diffuse components will reside at the foot of the plots shown in Figure 1 (a), the rest being the specular components.

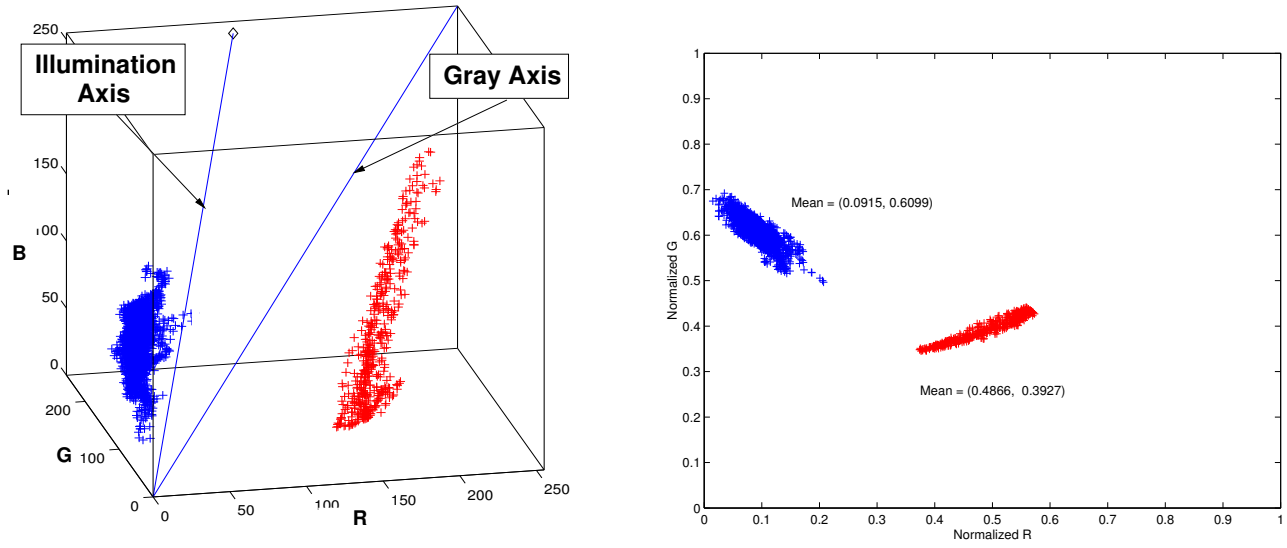


Figure 1: Illumination axis VS. Gray axis

previous approaches to modifying these color spaces to overcome the non-constancy of color with respect to illumination changes. This section also describes how dimensionality reduction is carried out in these color spaces. **Section 3** discusses about how to estimate color of the illuminant from images captured by a sensing device. **Section 4**, then describes our new color space. In this section, we also explain how to transform the *RGB* color space into our new color space to compensate for the illumination-induced color variations. In **Section 5**, we present experimental results using this color space transformation approach for the following computer vision applications: 1) Color Constancy; 2) Segmentation using the k-means clustering technique; and 3) Color histogram-based object tracking. Finally **Section 6** concludes this paper and suggests future works.

2. Previous Works on Color Spaces

Many color spaces have come into existence over the years. In their most general form, the color spaces tend to be three-dimensional – a property related to the fact that the human visual system uses three different types of cones in the retina for color perception and that the different types of cones are sensitive to different bands of the color spectrum. Of the various color spaces, the *RGB* (Red, Green, and Blue) color space is the one that is most commonly used for image processing work and for

designing phosphorescent displays. On the other hand, hard-copy displays and printers are designed using the *CMY* (Cyan, Magenta and Yellow) color space.

There are two color spaces – *YUV* and the *YIQ* – that mainly serve as color video standards. In both these representations, the intensity, denoted by Y, is separated from the chromaticity information which makes them attractive for TV broadcasting since the intensity values can be used all by themselves for monochrome reception. The *YUV* color space is used by the following composite color video standards: PAL (Phase Alternation Line) and SECAM (Sequential Color with Memory). The symbols U and V denote the chromaticity information in the *YUV* color space. The *YIQ* color space is further derived from the the *YUV* color space and is used by NTSC (National Television System Committee). The chromaticity information for the case of *YIQ* is captured by the inphase (I) and the quadrature (Q) components of the chromaticity vector.

If the goal is colorimetry, the color space to use is the *XYZ* space promulgated by *CIE* (*Commission internationale de I'Eclairage*). The tristimulus X, Y, and Z values are always positive for all real color stimuli.[2] As variations on the *XYZ* space, there are two other spaces, *LAB* and *LUV*, also formulated by *CIE*, that are more tuned to the properties of the human color perception. They are both thought of as perceptually uniform spaces, meaning that a small perturbation in a component value is approximately equally perceptible across the entire range of values for that component [4]. In both *LAB* and *LUV*, the L-axis stands for lightness, which is the same thing as brightness, and ranges from 0 for black to 100 for white. For *LAB*, the other two coordinates, A and B, represent redness-greenness and yellowness-blueness, respectively. For *LUV*, on the other hand, there is no analogous relationship between U and V and the *RGB* components as there is for the A and B planes in the *CIE LAB* color space. The non-linear relationship for L is intended to mimic the logarithmic response of the human eyes.

The color space that has been used the most frequently to match the properties of the human color perception is the *HSI* space. It has an intuitive appeal, in the sense that H stands for hue, the property that humans generally think of as the essence of a color, S that stands for saturation which to a human indicates the richness of a hue, and I that accounts for intensity – which is the same as brightness or luminosity – of a color.

[2]The coefficients for X, Y, and Z were chosen such that the Y value can be directly proportional to the luminance of the additive mixture. The linear transformation from the *RGB* space to the *XYZ* space is as follows: $X = 0.607R + 0.174G + 0.2B$, $Y = 0.299R + 0.587G + 0.114B$, and $Z = 0.066G + 1.116B$.

Color Space	Viewing Direction	Surface Orientation	Illumination Direction	Illumination Intensity	Color of Illuminant	Specular Highlight	Body Ref. Component	Surface Ref. Component
RGB	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	L	L
CMY	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	L	L
YUV	$\times$	$\times$	$\times$	$\bigcirc$	$\times$	$\times$	C	C
YIQ	$\times$	$\times$	$\times$	$\bigcirc$	$\times$	$\times$	C	C
XYZ	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	L	L
LAB	$\times$	$\times$	$\times$	$\times$	$\triangle$	$\times$	C	C
LUV	$\bigcirc$	$\bigcirc$	$\bigcirc$	$\bigcirc$	$\triangle$	$\times$	C	C
HSI	$\bigcirc$	$\bigcirc$	$\bigcirc$	$\bigcirc$	$\times$	$\bigcirc$	C	C
$l_1l_2l_3$	$\bigcirc$	$\bigcirc$	$\bigcirc$	$\bigcirc$	$\times$	$\bigcirc$	N/A	N/A
$h_1h_2h_3$	$\times$	$\times$	$\times$	$\times$	$\bigcirc$	$\times$	N/A	N/A

Table 1: Characteristics of three-dimensional color spaces
('$\times$' means "very sensitive to", '$\triangle$' means "relatively sensitive to", '$\bigcirc$' means "invariant to", 'L' means "shown as a Line", 'P' means "shown as a point", 'C' means "shown as a curve", and 'N/A' means "not applicable" or "not defined".)

Two other color spaces of note are $h_1h_2h_3$ and $l_1l_2l_3$. The former is based on channel differences that can be defined as follows: $h_1 = R - G$, $h_2 = G - B$, and $h_3 = B - R$ [3]. The differencing operations carried out on the primary colors R, G, and B could make this space potentially useful for ignoring specular highlights in images. However, the degree to which this space is effective for specularity suppression has not been established yet. The significance of the latter space, $l_1l_2l_3$, rests primarily on its geometric property that the three components will lie within the intersection of the chromaticity plane with the positive quadrant of the RGB cube.

Table 1 summarizes the characteristics of the color spaces mentioned above. It describes how each color representation scheme responds to the following illumination and imaging conditions: 1) Viewing Direction (column 1); 2) Surface Orientation (column 2); 3) Illumination Direction (column 3); 4) Illumination Intensity (column 4); 5) Color of Illuminant (column 5); Specular Highlight (column 6); 7) Shape of Body Reflectance Component (column 7); and 8) Shape of Surface Reflectance Component (column 8).

When a color space is invariant to one of the first six variables listed above, that fact is indicated with the '$\bigcirc$' symbol in the table. Otherwise, the symbols '$\triangle$' and '$\times$' indicate levels of sensitivity to the variables, the former denoting "relatively sensitive" and the latter "very sensitive".

To explain the last two columns, according to physics-based reflection models, such as Shafer's dichromatic reflection model [5] (described in **Section 4.3**) or Phong's shading model [6], there are

Chromaticity Space	Viewing Direction	Surface Orientation	Illumination Direction	Illumination Intensity	Color of Illuminant	Specular Highlight	Body Ref. Component	Surface Ref. Component
rg	◯	◯	◯	◯	△	×	P	L
HS	◯	◯	◯	◯	×	◯	C	C
xy	×	×	×	◯	△	×	P	P
uv (Yuv)	×	×	×	◯	△	×	N/A	N/A
IQ	×	×	×	△	△	×	N/A	N/A
uv (Luv)	◯	◯	◯	◯	△	×	N/A	N/A

Table 2: Characteristics of two-dimensional color subspaces
('×' means "very sensitive to", '△' means "relatively sensitive to", '◯' means "invariant to", 'L' means "shown as a Line", 'P' means "shown as a point", 'C' means "shown as a curve", and 'N/A' means "not applicable" or "not defined".)

two key components of the light reflected from an object surface. These are body reflection and surface reflection components and appear as two coplanar color clusters, each forming a linear blob in the RGB color space. As the RGB color space is transformed into another color space, the shapes of these two color clusters will change in general. The last two columns of the table show how the shapes of the two clusters change with the transformation specific to that space from the RGB space.

As shown in this table, also as discussed by Gever and Smeulders [3], none of these color spaces can be considered to produce color values that are entirely independent of both illumination intensity changes and illumination color. Thus it makes sense to seek color transformation methods that minimize such illumination effects by a re-mapping of the color representation.

2.1 Dimensionality Reduction of Color Spaces

It has been shown over the years that a reduced dimensionality representation of color may exhibit greater color constancy properties than the full three-dimensional representations we just mentioned. For example, with the RGB representation, if the color vector is normalized and its degrees of freedom reduced by one, we obtain what is sometimes referred to as the rg-chromaticity space. The measured color values in this reduced-dimensionality space seem to depend less on small variations in the color of the illuminant and in the brightness of that color [7, 8, 9]. Such advantages of reduced-dimensionality representation also appear to extend to some if not all of the 3-D color spaces mentioned previously. Shown below is a listing of these reduced-dimensionality representations and how they are derived from the corresponding 3-D representation:

- Derivation from the *RGB* color space: The reduced-dimensionality representation in this case uses only two normalized color values r and g, where $r = R/(R+G+B)$ and $g = G/(R+G+B)$, by assuming that the third normalized value if needed can be obtained from $r+g+b = 1$ [7, 8, 9].

- Derivation from the *XYZ* color space: The reduced-dimensionality representation in this case consists of two values, x and y obtained by $x = X/(X+Y+Z)$ and $y = Y/(X+Y+Z)$ and $x+y+z = 1$ [10]. The space spanned by x and y is also referred to as the *xy*-chromaticity space.

- Derivation from the *HSI* color space: The reduced-dimensionality representation in this case consists of retaining only the H (Hue) and the S (Saturation) values [11, 12, 13, 14].

- Derivation from the *YUV* color space: The reduced-dimensionality here is based on the following normalizations that give us two values, u and v: $u = U/Y$ and $v = V/Y$. If the initial color measurements are in the form of R, G, and B values, these formulas can be expressed directly using those by $Y = 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B$, $U = -0.147 \cdot R - 0.289 \cdot G + 0.437 \cdot B$ and $V = 0.615 \cdot R - 0.515 \cdot G - 0.1 \cdot B$ [15].

- Derivation from the *YIQ* color space: The reduced-dimensionality is based on the following normalizations that give us two values, i and q: $i = I/Y$ and $q = Q/Y$. If the initial color measurements are in the form of R, G, and B values, these formulas can be expressed directly using those by $Y = 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B$, $U = 0.596 \cdot R - 0.275 \cdot G - 0.321 \cdot B$ and $V = 0.212 \cdot R - 0.523 \cdot G + 0.311 \cdot B$ [15].

- Derivation from the *LUV* color space: The reduced-dimensionality is based on the following normalizations that give us two values, u^* and v^*: $u^* = 13L(U - U_n)$ and $v^* = 13L(V - V_n)$ where U_n and V_n are intermediate quantities derived from the reference white in the XYZ color space [14].

Table 2 summarizes the characteristics of these two-dimensional color subspaces with respect to the same set of variables as shown earlier in Table 1.

It is also possible to further reduce the dimensionality of a color space to a single scalar value. Smith and Chang [16] introduced a one-dimensional color representation in which the R, G, B values at a

pixel are reduced to a single number by $m = r + N_r g + N_r N b_g$ where N_r and N_g equal the number of bins used for R and G, respectively. The primary advantage of this transformation is the compactness of the color representation.

We should also mention that Finlayson *et al.* [17] has proposed an iterative color constancy approach to normalize out both illumination intensity and color changes. The first normalization step is similar to that of the normalized RGB as described above. The second normalization step transforms each pixel according to the global mean value in each color band. They then repeat the two processes until they reach a stable state when each normalization is idempotent. Assuming that the camera sensors have a very narrow bandwidth, their approach successfully works in the context of object recognition or image indexing in which normalization requires a pair-wise comparison between the corresponding pixels in two images.

2.2 Limitations of Reduced-Dimensionality Color Spaces

In general, these reduced-dimensionality color representations discard the intensity (*i.e.*, luminance) information and allow for better chromatic discrimination under varying illumination conditions. However, there are several limitations associated with these reduced-dimensionality color spaces:

- As mentioned already, one of the main goals in using reduced-dimensionality representations is to achieve color constancy and, up to a point, this can be achieved by making color descriptors invariant to the intensity. However, the two-dimensional color subspaces are invariant to the changes of illumination intensity only when the illumination spectrum matches with pure white and when it is spatially constant – which on account of self and other occlusions often does not hold true in real cases. The color discrimination achieved in the reduced-dimensionality spaces degrades significantly as the color of the illuminant becomes more and more dissimilar from the white illuminant. Readers will witness this phenomenon in **Section 4.2** where we present the segmentation results in these reduced-dimensionality color spaces.

- Another disadvantage of using these reduced-dimensionality color spaces (*e.g.*, *rg*-chromaticity space or the xy-chromaticity diagram) is that there is no way to differentiate between objects of similar hue but different intensity.

- The previous section talked about a reduced-dimensionality representation for each of the better known of the three-dimensional color representations. In practice, because cameras and other optical sensors record color in the form of RGB values, all reduced-dimensionality representations must start from RGB. The transformation from RGB to a given reduced-dimensional representation is often non-linear and non-reversible. The extent of non-linearity and non-reversibility introduced by these transformation determines the loss of information with regard to the true color. Greater the non-linearity in any of the transformation steps and larger the extent of non-reversibility, the greater the loss of true color. We are making a distinction between linearity and reversibility because it is possible to construct a linear transformation (say from RGB to XYZ) that may still be non-reversible because of the positivity constraints on the RGB values and lack of such constraints on the XYZ values.

To overcome the shortcomings of the existing color spaces, both three-dimensional and their associated two-dimensional color subspaces, we believe it is essential to establish a new color representation scheme that might lead us to a lower-dimensional color subspace with a greater ability to discount the illumination-induced changes in the values of the color descriptors. In the following section, we propose such a color space. Our new color space involves a transformation that adapts the color space to the color of the illuminant and leads to a color representation that is more independent of the illumination color than any of the existing approaches.

3. ESTIMATION OF THE COLOR OF THE ILLUMINANT

Our goal is to create a color representation that is "fine tuned" with respect to the color of the non-white illuminant so that it can efficiently describe the dichromatic plane in a three-dimensional color space. We wish to achieve the additivity of the RGB color space and the intuitiveness of the HSI (Hue, Saturation and Intensity) color space in our new color representation. To that end, we first estimate the color of the illuminant, as described in the rest of this section, and then apply linear transformations to the RGB cube with respect to the direction of the illuminant to create the new color space.

In much work reported in the past, reflectance spectrophotometers and colorimeters have been commonly used for measuring the color of the illuminant. A reflectance spectrophotometer scans different

wavelengths of light reflected from a surface of interest and records the intensity of that reflected light relative to a white standard. On the other hand, a colorimeter measures the tristimulus values more directly by using three broad-band filters. A colorimeter does not provide spectral reflectance data but the measured tristimulus values. The tristimulus values as recorded by a colorimeter are used for the derivation of the *CIE XYZ* color space. In the work reported in this paper, since our focus is primarily on narrow-band illumination, we will show that it suffices to use standard CCD cameras for illuminant estimation. An earlier contribution, by Pattanaik *et al.* [18], has also used a CCD camera for the estimation of the illuminant color. That work, not necessarily limited to narrow-band illuminants, consists of recording the output of a CCD camera through a set of narrow-band optical filters. The cameras are calibrated for the gamma correction in each channel of the *CIE XYZ* tristimulus response. The outputs of the narrow-band optical filters are then combined after the gamma correction to form the *CIE XYZ* values. One thus obtains the colorimetric values directly from the CCD output. Our approach to illuminant estimation, being limited to narrow-band illuminants, differs from the work reported in [18] in the sense that we do not need the tristimulus values and therefore we do not need to calibrate the cameras for the gamma correction.

As mentioned earlier, our attention is on non-white illumination conditions where the spectrum of the illuminant is limited to a narrow range compared to that of natural light[3]. We propose a linear interpolation approach to estimating the color of the illuminant which can be summarized as follows: We first repeatedly measure the color values of the reflected light from a planar surface covered with a barium sulfate (or a similar white material) placed at different distances from the light source. For each measurement, we calculate the average values for R, G and B channels. In the RGB color space, we then use linear least squares fitting to estimate the straight line that corresponds to the illumination axis. In the remainder of this section, we will elaborate the process of the illuminant color estimation.

For each different illuminant, a training set of 40 images is recorded in the following manner: Conforming to the 10^o *CIE* standard observer, a digital camera is installed approximately $50cm$ away from the white reference plate. For each of 20 different locations of the camera (for example, see Figure 2 (a)), two images are captured for two viewing directions (approximately 0^o and 45^o to the surface normal of the reference plane). For each such sampled image (see Figure 2 (b)), the mean (μ_R, μ_G, μ_B)

[3]Estimation of natural light illuminants under daylight conditions is addressed in [19, 20].

(a) (b) (c)

Figure 2: Estimation of the illuminant color (an example scene of a parking structure with low-pressure sodium lighting)

(a) Example scene of the Illumination geometry ("⊗" marks indicate the locations of the white reference plate); (b) The 40 square regions, each of size 100x100 pixels, extracted from the 40 different images that are recorded for an experiment; and (c) The Illumination axis estimated from the recorded RGB values is marked with a blue dotted line.

and the standard deviation $(\sigma_R, \sigma_G, \sigma_B)$ in the R, G, and B color channels are calculated. If the mean is close to the max for a color channel, that indicates saturation in that channel. The patch for which this is true is discarded from further calculations. A patch may also be discarded if the calculated mean for the patch is too close to zero for any of the three color channels[4]. Figure 2 (c) plots the average color values calculated from the selected patches for low-pressure sodium lighting. Linear least squares fitting is applied to estimate the best fitting straight line that defines the *illumination axis* in the RGB cube. The main idea here is that, assuming that illuminant is sufficiently narrow-band, the direction of the color of the illuminant in the RGB color cube can be described by a straight line starting from the origin of the RGB cube. As has been observed by various researchers in the past [20, 21, 17], this illuminant color direction corresponds to the surface reflection vector defined in the dichromatic reflection model (see **Section 4.3**).

The illuminant estimation results for nine different scenes, each with a different illuminant, are tabulated in Table 3. The main purpose of the table is to show the direction of the illumination axis in the RGB cube. As shown in the fifth column, the direction of the illumination axis (denoted as $\hat{L}_{RGB}$) is presented by a unit vector, these being the projections of the illumination axis on the R, G, and B

[4]The following thresholds have worked well for us for detecting saturation and lack of adequate illumination: 1) over-saturation threshold: $max\,(\mu_R, \mu_G, \mu_B) \geq 252$; and 2) inadequate illumination threshold: $min\,(\mu_R, \mu_G, \mu_B) \leq 2$.

axes. The last column of the table also shows the rotation axis by a unit vector for a certain rotation of the RGB cube that will be explained in the next section. We should also state that in all of the experiments used for the construction of this table, white balancing and automatic gain control in the cameras were made inactive.

Our discussion in this section has focused entirely on estimating all three direction components of the illumination axis in the RGB color cube. We should also mention that one could talk about estimating the direction of the illumination vector in a reduced-dimensional chromaticity space, but, as the reader will infer from the transformations we present in the next section, that will not serve our goal. Nonetheless, for the sake of completeness, we should mention that [22, 23] have estimating the illuminant vector in chromaticity space using specular highlights. Tan *et al.* [24] suggested a unique way to estimate the illumination chromaticity using inverse-intensity chromaticity space.

4. AN ILLUMINATION ADAPTED COLOR SPACE

As stated in the introduction, our overall goal is the development of a more principled approach to the representation of color so that we can obtain a truer measurement of the color of an object surface even under non-white illumination. This obviously requires that the color of the illuminant itself be taken into account in the measurement of the color of an object surface. We believe that this is best done by employing a color space that adapts to the ambient illumination. As we will show in the rest of this article, such adaptation can be carried out in a manner that makes the color descriptors measured for object surfaces substantially independent of the illumination itself.

In the subsection to follow, we will describe the geometrical transformations that take us from the RGB space in which the camera measurements of the color are made to an illumination-adapted color space. In **Section 4.2**, we will then explain how to extract 3-D and 2-D color descriptors from the $X_L Y_L Z_L$ color space. We will show, in **Section 4.3**, that a single scalar number which is the rotational angle of the projection of the color vectors into the $X_L Y_L$-plane is sufficient for illumination-independent characterization of the dichromatic plane. The last subsection briefly provides an intuitive explanation for why, under non-white illumination, the color descriptors in the new space are less likely to be affected by, say, the intensity of the illumination compared to using the RGB directly. It will also list

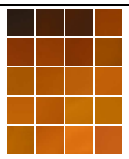
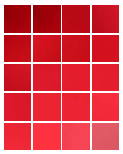
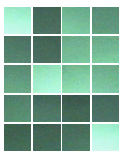
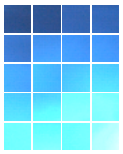
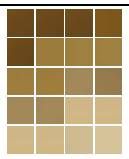
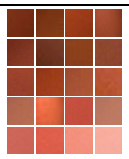
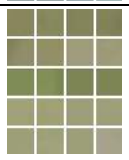
(i) Light Sources	(ii) Lighting geometry	(iii) White References	(iv) Illuminants (Watts/Units) (W_{max})	(v) Illumination Axis in *RGB* ($\hat{L}_{RGB}$=(L_R,L_G,L_B)	(vi) Illumination Axis in *XYZ* ($\hat{L}_{XYZ}$=(L_X,L_Y,L_Z))	(vii) Rotation Axis in *XYZ* ($\hat{U}$) (U_{XYX}=(u,v,w))
(1) Low-pressure Sodium Light			Low-pressure Sodium Light Bulbs (100Watts X 20)	$L_R = 0.7930$ $L_G = 0.5679$ $L_B = 0.2205$	$L_X = 0.3418$ $L_Y = 0.2579$ $L_Z = 0.9037$	$u = 0.6023$ $v = -0.7983$ $w = 0$
(2) High-pressure Sodium Light			High-pressure Sodium Light Bulbs (100Watts X 14)	$L_R = 0.8992$ $L_G = 0.4270$ $L_B = 0.0956$	$L_X = 0.5415$ $L_Y = 0.2436$ $L_Z = 0.8046$	$u = 0.4103$ $v = -0.9128$ $w = 0$
(3) Incandescent Light (red)			Sylvania Flood Par38 Red: *Peak* 625nm (100Watts X 5)	$L_R = 0.9834$ $L_G = 0.0810$ $L_B = 0.1624$	$L_X = 0.7240$ $L_Y = -0.0592$ $L_Z = 0.6872$	$u = -0.0815$ $v = -0.9967$ $w = 0$
(4) Incandescent Light (green)			Sylvania Flood Par38 Green: *Peak* 535nm (100Watts X 5)	$L_R = 0.4351$ $L_G = 0.6888$ $L_B = 0.5799$	$L_X = -0.1722$ $L_Y = 0.0815$ $L_Z = 0.9817$	$u = 0.4278$ $v = 0.9039$ $w = 0$
(5) Incandescent Light (blue)			Sylvania Flood Par38 Blue: *Peak* 425nm (100Watts X 5)	$L_R = 0.2278$ $L_G = 0.5112$ $L_B = 0.8287$	$L_X = -0.3787$ $L_Y = -0.2355$ $L_Z = 0.8951$	$u = -0.5281$ $v = 0.8492$ $w = 0$
(6) Incandescent Light			*GE* Flood Par38 Yellow: *Peak* 570nm (200Watts X 3)	$L_R = 0.7556$ $L_G = 0.5914$ $L_B = 0.2816$	$L_X = 0.2744$ $L_Y = 0.2307$ $L_Z = 0.9335$	$u = 0.6435$ $v = -0.7654$ $w = 0$
(7) Tungsten Light Bulbs (pink)			GE Orange: *Peak* 600nm (100Watts X 6)	$L_R = 0.8606$ $L_G = 0.4455$ $L_B = 0.2467$	$L_X = 0.4402$ $L_Y = 0.1473$ $L_Z = 0.8857$	$u = 0.3173$ $v = -0.9483$ $w = 0$
(8) Halogen Light Bulbs			Halogen Light Bulbs: (500 Watts X 1)	$L_R = 0.5180$ $L_G = 0.6283$ $L_B = 0.5804$	$L_X = -0.0747$ $L_Y = 0.0360$ $L_Z = 0.996$	$u = 0.4341$ $v = 0.9008$ $w = 0$
(9) Florescent Light			GE StsrCoat T8: (32Watts X 12)	$L_R = 0.5975$ $L_G = 0.5834$ $L_B = 0.5501$	$L_X = 0.0266$ $L_Y = 0.0249$ $L_Z = 0.0993$	$u = 0.6834$ $v = -0.7301$ $w = 0$

Table 3: Estimation of illuminant color for nine different scenes, each with a different illuminant. (i) Light source used for illumination; (ii) The scene, (iii) 20 of the 40 image patches recorded in each experiment; (iv) Information about the illumination source, including the wavelength (W_{max}) that corresponds to peak illumination intensity; (v) The normalized unit vector ($\hat{L}_{RGB} = (L_R, L_G, L_B)$ for the direction of the illumination axis in the RGB cube; (vi) The normalized unit vector ($\hat{L}_{XYZ} = (L_X, L_Y, L_Z)$) for the direction of the illumination axis in the XYZ space; and (vii) The rotation axis $\hat{U}_{XYZ} = (u, v, w)$ in the XYZ space that is perpendicular to both the illumination axis and the z-axis. This will be used as the rotation axis for the second rotation to bring the illumination axis into alignment with the vertical axis (Z-axis). (see **Section 4**)

some of the unique advantages of our new approach to color representation.

4.1 The $X_L\, Y_L\, Z_L$ Color Space

The illumination adaptation that we will present in this subsection consists of reorienting the RGB cube so that the illuminant axis itself becomes one of the cardinal axes. This problem of reorienting the RGB cube is a little more complex than merely rotating the cube through the illumination axis angle measured in **Section 3**. As we discuss here, that angle does not provide us with sufficient information to fully constrain the rotation needed. The new color space is obtained by transforming the RGB cube in which the illuminant color is estimated in the following manner:

- As a precursor to a soon-to-be-described rotation of the RGB cube, label the R-axis as X, the G-axis as Y, and the B-axis as Z.

- Now rotate the RGB cube in the XYZ Cartesian coordinate frame such that the gray axis of the RGB cube is aligned with the vertical, meaning the "Z", axis. This is illustrated in Figure 3 (a). Note that the gray axis is the diagonal line that joins the origin with the maximally distant point in the cube, the point where R, G, and B all take their maximal values.

- Further rotate the RGB cube in the XYZ space until the illumination axis coincides with the Z-axis. The color space thus obtained will be denoted by $X_LY_LZ_L$. This is illustrated in Figure 3 (b).

These transformations and the logic on which they are based will now be explained in greater detail. It is important to understand the reason why the overall transformation that yields the new color space needs to be broken down into the two steps. The direction of the illuminant, specified by its three direction cosines shown in Table 3, has only two degrees of freedom. On the other hand, one must constrain three degrees of freedom to fully specify a rotation of one coordinate frame with respect to another coordinate frame that is centered at the same origin. Of the three degrees of freedom, two will specify the direction around which the coordinate frame is to be rotated and the third the degree of rotation around the axis. Since the illuminant axis in the RGB frame fixes only two degrees of freedom, we are at liberty to constrain the third in any way we like.

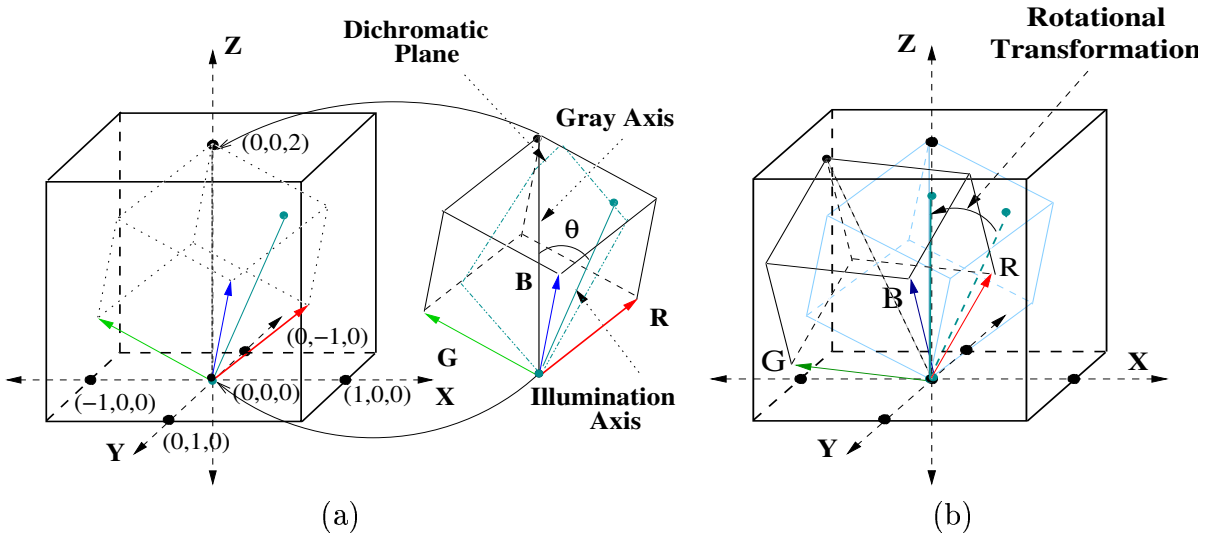


Figure 3: Color space transformations
(a) The first (linear) transformation; and (b) the second (rotational) transformation.

We have chosen to constrain the third degree of freedom of the new color space in such a way that it retains the intuitiveness that people associate with the *HSI* space. This explains the first of the two transformations mentioned above. As the reader will recall, the vertical axis of the *HSI* space stands for "Intensity", which corresponds to the diagonal axis in the *RGB* space. Since our first transformation aligns this axis with the vertical – a traditional step in the *RGB* to *HSI* transformation matrix – the vertical axis after the transformation stands for intensity with respect to white light illumination. Our second transformation then brings the illuminant axis into coincidence with the vertical. The new vertical then represents the same intensity effects as the *HSI* vertical, except that now the intensities in our space as represented by the vertical are with respect to the color of the illuminant.

The first transformation is represented by T^{RGB}_{XYZ}. Borrowing from the well-known transformations between *RGB* and *HSI*, the various elements of this matrix are shown below (see also Figure 3):

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = T^{RGB}_{XYZ} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix} = \frac{1}{255} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ \frac{2}{3} & \frac{2}{3} & \frac{2}{3} \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{1}$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \left[T^{RGB}_{XYZ}\right]^{-1} \cdot \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = 255 \begin{bmatrix} \frac{2}{3} & 0 & \frac{1}{2} \\ -\frac{1}{3} & \frac{1}{\sqrt{3}} & \frac{1}{2} \\ -\frac{1}{3} & -\frac{1}{\sqrt{3}} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{2}$$

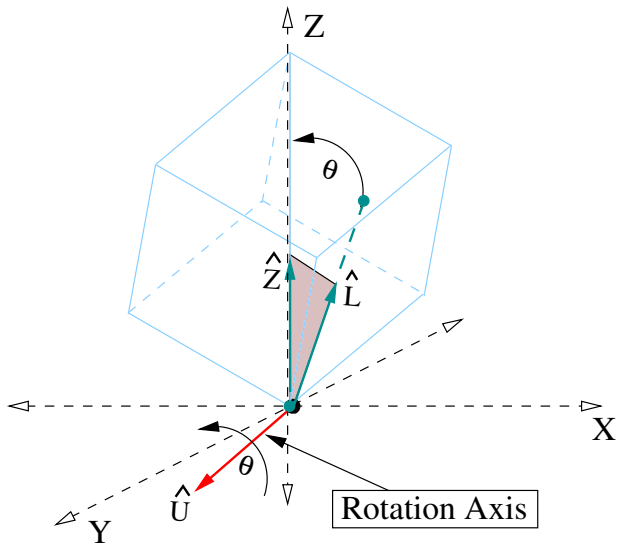


Figure 4: The rotational transformation $R^{XYZ}_{X_L Y_L Z_L}$

The second transformation rotates *RGB* cube in the *XYZ* space by θ around the rotation axis U. Here the rotation axis U can be derived from the cross product of the illumination axis and the gray axis.

The matrix transformation shown is based on the assumption that the *RGB* color space has its color values ranging from 0 to 255 in each channel. The scaling factors shown are based on the assumption that we want the X-axis and the Y-axis to vary between -1 and 1 and the Z-axis to vary between 0 and 2. The choice of the scaling factors is dictated by our desire to see an approximation to the *HSI* space in the *XYZ* space. While it is true that the vertical axis – the intensity axis – ranges from -1 to +1 in the genuine *HSI* space, for computer processing the same axis is best represented as varying from 0 to 2. The above transformation also assumes that the R-axis remains aligned with the X-axis. That is because "Red" is the starting hue with respect to the angular variations in the *HSI* space. The *RGB* cube stands on its black corner after the transformation, as shown in Figure 3 (a), and the values $\sqrt{X^2 + Y^2}$ and $arctan2(Y/X)$ correspond approximately to the saturation (S) and the hue (H) in the *HSI* space, respectively.

Our next transformation converts this *HSI-like* representation into what turns out to be a useful color representation scheme for non-white illuminations. This transformation further rotates the *RGB* cube until the illumination axis becomes the vertical axis of the XYZ coordinate frame (see Figure 3 (b)). This transformation is carried out by rotating the *RGB* cube by the minimum angle between the gray axis (the vertical axis after the first transformation) and the illumination axis. We refer to

this angle, which is always acute, as the *illumination axis angle* and denoted it by θ (see Figure 4). As shown in Figure 4, let us define two unit vectors, $\hat{L}$ and $\hat{Z}$ along the illumination axis and the Z-axis in the XYZ space, respectively. Then we can define another unit vector $\hat{U} = (u, v, w)$ that is perpendicular to both the illumination axis and the gray axis and is given by the cross-product of $\hat{L}$ and $\hat{Z}$. Notice that the unit vector $\hat{U}$ is always in the XY-plane, implying that $U_z = 0$ always. Our goal is to rotate the RGB cube around $\hat{U}$ through angle θ. The new color space obtained in this manner will be denoted $X_L Y_L Z_L$, as previously mentioned. For example, the new color space that corresponds to the Figure 3 (1) can be specified by $X_{0.7930} Y_{0.5679} Z_{0.2205}$. The θ-rotation of the RGB cube in the XYZ space will be denote by the notation $R^{XYZ}_{X_L Y_L Z_L}$. The matrix $R^{XYZ}_{X_L Y_L Z_L}$ is given by the following equations:

$$\begin{bmatrix} X_L \\ Y_L \\ Z_L \end{bmatrix} = R^{XYZ}_{X_L Y_L Z_L} \cdot \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{3}$$

$$= \begin{bmatrix} u^2+(v^2+w^2)\cdot cos\theta & uv(u-cos\theta)-w\cdot sin\theta & uw(1-cos\theta)+v\cdot sin\theta \\ uv(1-cos\theta)+w\cdot sin\theta & v^2+(w^2+u^2)\cdot cos\theta & uv(1-cos\theta)-u\cdot sin\theta \\ uw(1-cos\theta)-v\cdot sin\theta & vw(1-cos\theta)+u\cdot sin\theta & w^2+(u^2+v^2)\cdot cos\theta \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{4}$$

$$= \begin{bmatrix} u^2+v^2\cdot cos\,\theta & uv(1-cos\,\theta) & v\cdot sin\,\theta \\ uv(1-cos\,\theta) & v_2+u^2\cdot cos\,\theta & uv(1-cos\,\theta)-u\cdot sin\,\theta \\ -v\cdot sin\,\theta & u\cdot sin\,\theta & (u^2+v^2)\cdot cos\,\theta \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{5}$$

where u, v, and w are the projections of $\hat{U}$ on the X, Y, and Z axes, respectively. Eq. 4 is the generic form of the rotational transformation matrix when an object, in this case the RGB cube, is rotated around a designated axis through a specified angle in a fixed coordinate frame. As mentioned above, we know that the w value for the unit vector $\hat{U} = [u, v, w]$ is always 0. Therefore, Eq. 4 becomes much simpler and leads to Eq. 5. In addition, using the orthogonality between $\hat{U}$ and $\hat{L}$, the unit vector $\hat{U} = (u, v, w)$ can be expressed as

$$\hat{U} = \begin{bmatrix} u \\ v \\ w \end{bmatrix}^T = \frac{1}{\sqrt{L_X^2 + L_X^2}} \cdot \begin{bmatrix} L_Y \\ -L_X \\ 0 \end{bmatrix}^T \tag{6}$$

4.2 Extracting Color Descriptors from the $X_L Y_L Z_L$ Color Space

The previous subsection focused on how to carry out the geometrical transformations that take us from the *RGB* space in which the camera measurements of the color are made to an illumination-adapted color space. In this subsection, we address the issues of how to extract color descriptors from the $X_L Y_L Z_L$ representation for the purposes of illumination-invariant display of object color, image processing, and color image understanding.

When it comes to displaying colored objects on either terminal monitors or through hard-copy printing devices, we need all three components of color. For terminal monitors, these are most commonly the *RGB* components, and for hard-copy printouts, these will be the *CMY* components. What that implies is that in order to display on a terminal monitor a color represented in our $X_L Y_L Z_L$ color space, we must somehow convert the color vector back into the *RGB* space. How to do this will be addressed in **Section 4.2.1**. Suffice it to say here that this transformation *cannot* be the exact reverse of the *RGB* to $X_L Y_L Z_L$ transformation that we showed in **Section 4.1**, since an exact reverse will result in the same illumination-dependent color vector that we started out with.

Then there is the issue of what information to extract from the $X_L Y_L Z_L$ for image processing and image understanding work. Traditionally, reduced-dimensionality color descriptors have been used for such purposes. For example, frequently the normalized *RGB* that has only two degrees of freedom is used. Another possible descriptor is the 2-D representation in the chromaticity plane that we will explain in greater detail later in this paper. In **Section 4.2.2**, we will explain how our $X_L Y_L Z_L$ representation lends itself well to a two-dimensional color descriptor that given illumination invariant measure of the color of an object surface suitable for image processing and image understanding tasks.

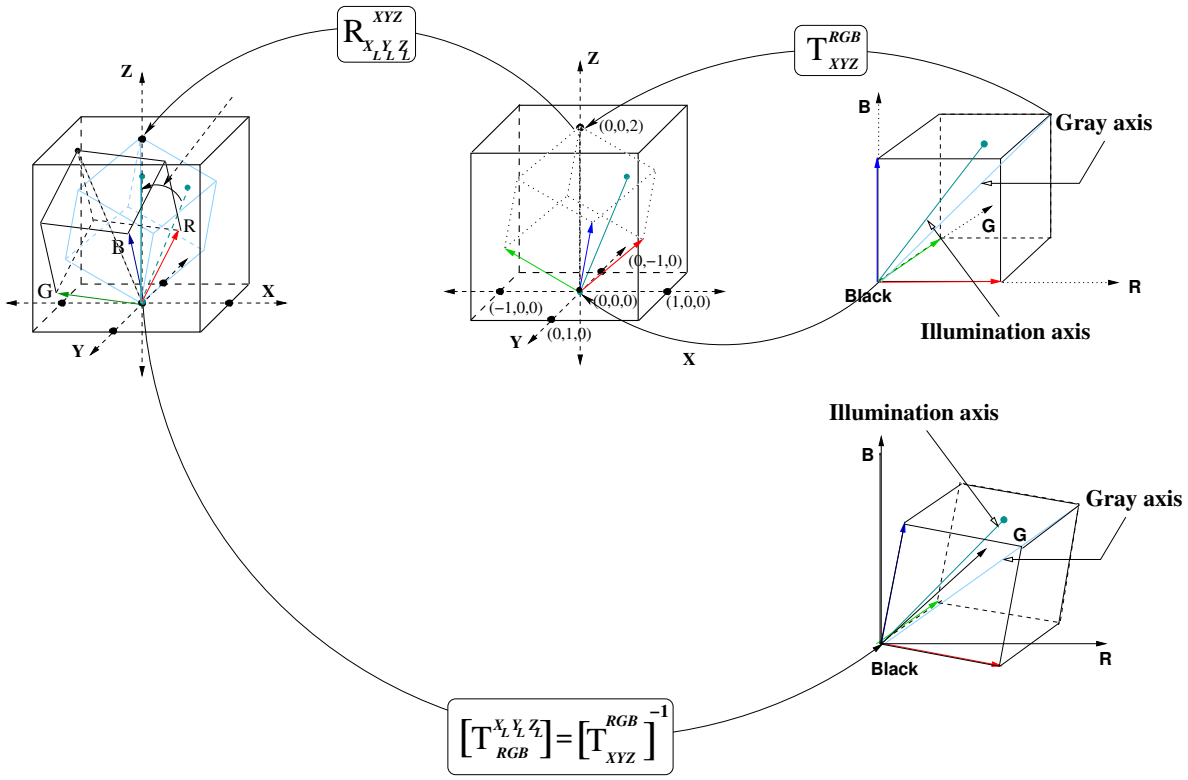


Figure 5: The reverse transformation $\left(T^{RGB}_{XYZ}\right)^{-1}$.

The upper set of transformations represent the forward process from the RGB space in which color is initially measured to the $X_LY_LZ_L$ space. The lower arc shows the reverse transformation from the $X_LY_LZ_L$ color representation to RGB values needed for illumination invariant display of a colored object on a terminal monitor.

4.2.1 Extraction of 3-D Descriptors for Illumination-invariant Display of the Object Color

As described in **Section 4.1**, there are two transformations from the RGB space in which color measurements are initially made to the $X_LY_LZ_L$ color space: a linear transformation T^{RGB}_{XYZ} followed by another linear transformation $R^{XYZ}_{X_LY_LZ_L}$. Recall that the former transformation rotates and scales the RGB cube so that it forms an *"HSI-like"* structure in the intermediate space, the XYZ space. And the second transformation matrix, $R^{XYZ}_{X_LY_LZ_L}$, carries out another rotation so that the illumination axis is aligned with the vertical axis where the gray axis was originally.

Now suppose our goal is to display on a terminal monitor a colored object that is represented in the $X_LY_LZ_L$ space. We obviously do not want to reverse the two transformations of the forward process. Applying both those transformation in reverse would negate the very purpose of the $X_LY_LZ_L$ space. We will now argue that an appropriate transformation for generating the RGB values for display – we will refer to these as the $R_DG_DB_D$ values – is just $\left[T^{RGB}_{XYZ}\right]^{-1}$. If our illumination was perfectly white,

this would indeed be the mathematically precise way to get from the *HSI*-like *XYZ* space to the *RGB* space. Basic to the $X_L Y_L Z_L$ space is the notion that we want it to have all of the properties of the white-illumination based *XYZ* space but with respect to the illumination axis. This requires that we treat the illumination axis in the $X_L Y_L Z_L$ space just as we would treat the Z axis – the gray axis – in the *XYZ* space. We claim that that when the displayable $R_D G_D B_D$ values are extracted by applying $\left[T_{XYZ}^{RGB}\right]^{-1}$ to a color vector in the $X_L Y_L Z_L$ space, the result is greater color constancy in the displayed output. Our claim carries no mathematical proof – since we do not believe it is possible to provide such a proof. We will, however, establish the veracity of our claim through the experimental results shown in **Section 5**. On intuitive grounds, we can justify our approach by saying that $R_D G_D B_D$ is to the illumination axis what *RGB* would be to the gray axis. In other words, $R_D G_D B_D$ values measure *color with respect to the illumination axis in just the same manner as RGB values measure color with respect to the gray axis.*

Therefore, the reverse transformation that takes us from the $X_L Y_L Z_L$ color space back to the *RGB* color coordinates for the purpose of rendering may be expressed as:

$$\begin{bmatrix} R_D \\ G_D \\ B_D \end{bmatrix} = \left[T_{XYZ}^{RGB}\right]^{-1} \cdot \begin{bmatrix} X_L \\ Y_L \\ Z_L \end{bmatrix} = 255 \begin{bmatrix} \frac{2}{3} & 0 & \frac{1}{2} \\ -\frac{1}{3} & \frac{1}{\sqrt{3}} & \frac{1}{2} \\ -\frac{1}{3} & -\frac{1}{\sqrt{3}} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} X_L \\ Y_L \\ Z_L \end{bmatrix} \tag{7}$$

The forward and the reverse transformations are depicted in Figure 5.

4.2.2 Extraction of 2-D Descriptors for Image Segmentation, *etc.*

As already mentioned, it is more convenient to use some sort of a reduced-dimensionality representation of color in image processing and image understanding work. Frequently, people use the normalized *RGB* representation that has two degrees of freedom and that has the advantage of providing a modicum of color constancy as the normalization step reduces the color variations caused by changes in the brightness of illumination. Again as already mentioned, other reduced-dimensionality representations consist of taking the projection of the color vector on some 2-D plane in the *RGB* cube – frequently, the distinguished plane on which the color vector is projected is the chromaticity plane. Now the question arises: What kinds of reduced-dimensionality representations can be obtained from the $X_L Y_L Z_L$

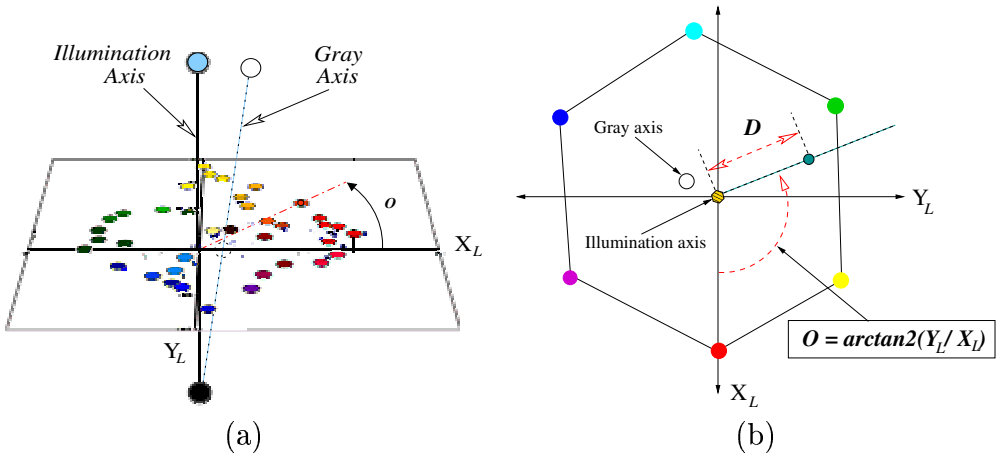


Figure 6: The X_LY_L-plane.
(a) The $X_LY_LZ_L$ color space projected onto the X_LY_L-plane; and (b) the 2-D color descriptors acquired from the projection onto the X_LY_L-plane.

representation of color?

A two-dimensional descriptor formed by projecting the $X_LY_LZ_L$ vector onto the X_LY_L-plane gives us a convenient reduced-dimensionality representation in the new color space. Projecting the color vector onto the X_LY_L-plane is similar to extracting a 2-dimensional representation of color by projecting the full color vector onto the *HS*-plane of the *HSI* color space. In fact, under ideal white illumination, the two would be equivalent. In both cases, the goal is to extract a reduced-dimensionality representation that makes the measured color immune to the brightness variations in the illumination. Ignoring the I component achieves that for the *HSI* case, provided the illumination is white, and ignoring Z_L in our new space does the same for a non-white illumination. This projection onto the X_LY_L-plane is illustrated in Figure 6 (a).

We represent the projection of the $X_LY_LZ_L$ vector onto the X_LY_L-plane by the parameter pair (D, O) where

$$O = arctan2(\frac{Y_L}{X_L}) \tag{8}$$

The function $arctan2$ calculates the inverse tangent and returns an angle in the correct quadrant in the range from $-\pi$ and π. The parameter D is the radial distance of the projected point from the origin, as shown in Figure 6 (b). In analogy with the *HSI* space, we can refer to O as the illumination-adapted hue and D as the illumination-adapted saturation at a pixel.

We will use the (D, O) color descriptions in **Section 5.2**, where we present image segmentation and video tracking results on actual color imagery. In the next section here, we will first present a physics-based rationale for the superiority of our new color space when dealing with non-white illumination, followed by a listing of some of the unique advantages of the $X_L Y_L Z_L$ color space.

4.3 Representing the Dichromatic Plane in the $X_L Y_L Z_L$ Color Space

Over the years, the dichromatic plane, first advanced by Shafer [5], has served as a convenient tool to understand the interplay between surface reflections, whose spectral composition often corresponds to that of the illumination, and body reflections that are more truly representative of the true color of an object. For dielectric non-homogeneous materials, the dichromatic reflection model describes the reflected light as an additive mixture of two components: 1) the body reflectance component, and 2) the surface reflectance component. For example, we can write the following expression for the color value at a pixel located at (x, y):

$$C(x,y) = m_b(i,e,g)\, C_b + \, m_s(i,e,g)\, C_s \, . \tag{9}$$

where $C_b = [R_b, G_b, B_b]$ and $C_s = [R_s, G_s, B_s]$ are the body reflection vector and the surface reflection vector, respectively. Here, i is the angle of incidence, e is the angle of reflection and g is the phase angle, and m_b and m_s are geometric scale factors of the body reflectance component and the surface reflectance component respectively. The plane spanned by the body reflection vector C_b and the surface reflection vector C_s is known as the dichromatic plane. To demonstrate the utility of the dichromatic plane, shown in Figure 7 (a) and (b) are the measurements of the reflected light at a single pixel as the dimmer on a source of illumination is varied, keeping everything else the same (Figure 7 (a) shows the dichromatic plane in the *RGB* color cube whereas Figure 7 (b) shows the same dichromatic plane vertically standing in the $X_L Y_L Z_L$ color space after our color space transformation). Note that pattern of measurements consists of two straight line segments, one corresponding to the body reflections and the other to the surface reflections. The size of these segments depends on the spectral richness of the illumination *vis-a-vis* the reflectivity properties of the object surface point that corresponds to the pixel at (x, y). When the illumination is narrow-band, the segment along the body reflection vector

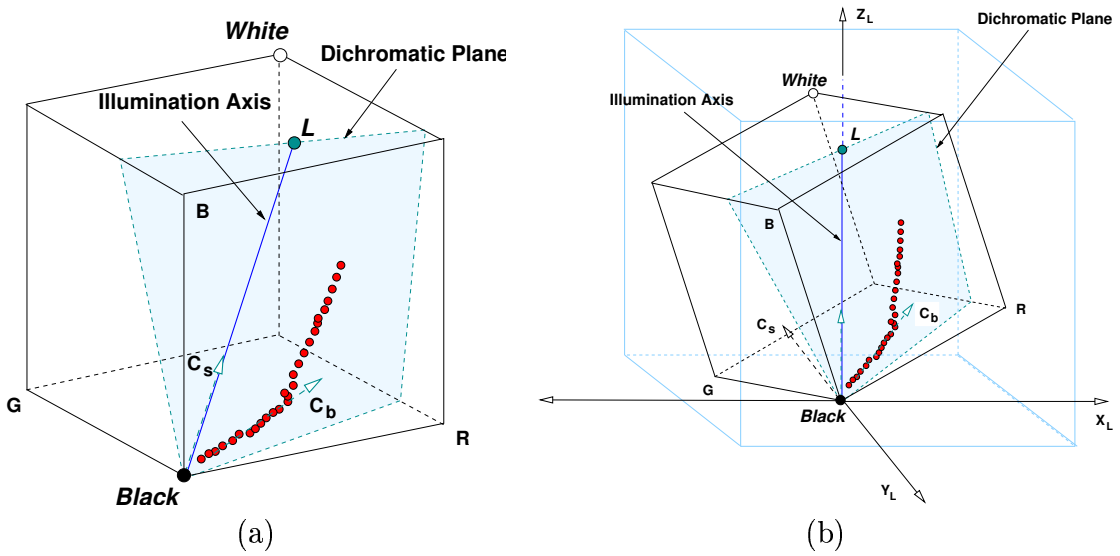


Figure 7: Measurements of the reflected light at a pixel (x,y) with varying the illumination intensity (a) Dichromatic plane in the *RGB* color cube; and (b) After the color transformation completed, the same dichromatic plane appears in the $X_L Y_L Z_L$ space.

C_b will be relatively short. The same thing happens when the object surface becomes more and more mirror-like. The important thing to bear in mind is that as the intensity of illumination is varied, the measurements will always be along the two-segment pattern shown in the figure. This figure also illustrates the experimental fact that under very low intensities of illumination, the measured color will be close to the true color of the object – assuming that the illumination is sufficiently broadband and assuming that the object surface is sufficiently diffuse.

Given the significance of the dichromatic plane, we need to address the issue of what happens to this plane in the $X_L Y_L Z_L$ color space. Fortunately, the dichromatic plane acquires a particularly simple representation in our new color space, simple in the sense that it acquires a one-parameter representation. This is a consequence of the fact that the illumination axis is always aligned with the Z_L-axis in the $X_L Y_L Z_L$ color space. Since the surface reflection vector C_s of the dichromatic plane is the same as the illumination axis [17, 20, 21], there is only one degree of freedom left for the dichromatic plane in the $X_L Y_L Z_L$ color space – that is regardless of the illumination and dependent only on the color of the object, as illustrated in Figure 8. That is, the degree of freedom, dependent only on the color of the object, is the angle of rotation of the dichromatic plane around the Z_L-axis.

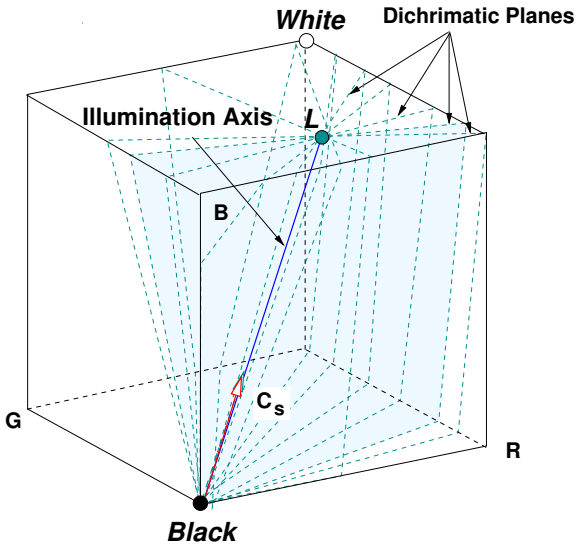


Figure 8: Dichromatic planes along the illumination axis

4.4 Advantages of the $X_L Y_L Z_L$ Color Space

With the help of a simple experiment, in this section we will first provide an intuitive explanation for why, under non-white illumination, the new color representation is to be preferred over other color descriptors. We will then list some of the unique advantages of the $X_L Y_L Z_L$ color representation.

Suppose there are two balls with two different colors, green and yellow. As demonstrated by the two images in Figure 9 (a), these two objects look very different and possess very different normalized color values under ambient white illumination. However, as the color of the illuminant changes into blue, the measured color of the objects varies dramatically. Shown in Figure 9 (b) is the image of the yellow ball under blue-colored illumination. Notice that the two recorded images of the yellow ball under blue-colored illumination (shown in Figure 9 (b)) and the green ball under ambient lighting (shown in Figure 9 (a)) are very alike in color content, implying that the normalized color values will be nearly the same for both cases. On the other hand, if we record the image of Figure 9 (b), that is the image of the yellow ball under the blue illumination, in the $X_L Y_L Z_L$ color representation and display it using the reverse transformation described in **Section 4.2.1** (see Eq. 7), the result as shown in Figure 9 (c) demonstrates that the illumination-induced color changes are immensely reduced.

This result helps us see experimentally why, under non-white illumination, the new color representation is to be preferred over other color descriptors. We will now summarize some of the unique

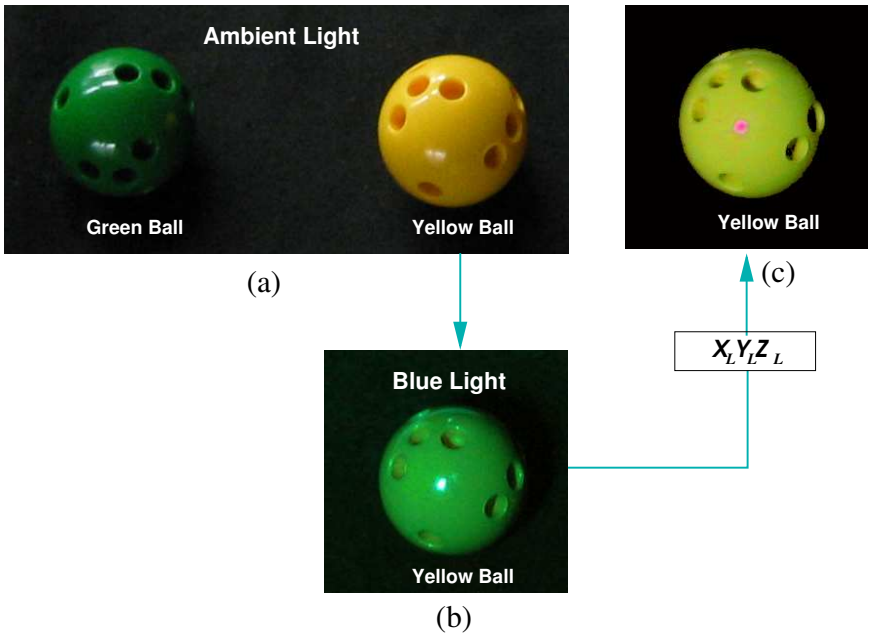


Figure 9: Discounting the Illumination-induced color changes.
(a) A green ball and a yellow ball under ambient lighting; (b) The yellow ball under blue-colored lighting; and (c) The image of (b) after processing in the $X_L Y_L Z_L$ space. The recovered image shown in (c) evidently looks more yellow, and therefore, more illumination independent than the measured image in (b).

advantages of the $X_L Y_L Z_L$ color representation in the rest of this section:

- **Discounting the illumination-induced color variations:** As we showed in the previous experiment (see Figure 9 (c)), the $X_L Y_L Z_L$ color representation along with the reverse transformation described in **Section 4.2.1** (see Eq. 7) is capable of discounting illumination-induced variations in the measurement of color. This amounts to achieving color constancy, a topic that we will address in greater detail in **Section 5.1**.

- **Compactness of the descriptions for the two reflectance components:** Another advantage of our new color representation is that the compactness of the description of the object color for computer vision tasks. As we pointed out in **Section 4.3**, the rotational angle of the projection of the color vectors into the $X_L Y_L$-plane is sufficient for illumination-independent characterization of the dichromatic plane.

- **Increased color separability:** The $X_L Y_L Z_L$ color representation provides better color separability, as demonstrated by the experimental results in **Section 5.2**. The better color separability

in the $X_L Y_L Z_L$ space is a consequence of the fact that the dichromatic planes in this space are all vertical and only differ with respect to their rotational position around the vertical axis, as explained previously with the help of Figure 8.

- **Easy adaptation to the illumination color changes:** It is very simple to switch an $X_L Y_L Z_L$ color representation into another $X_L Y_L Z_L$ color representation with respect to new illumination. This transformation can be carried out in the following manner: Suppose the color of the illuminant has changed from (R_1,G_1,B_1) to (R_2,G_2,B_2), with the corresponding change in the illumination axis from $L1$ to $L2$ in the underlying RGB space, then the following transformation takes a measured color vector $X_{L1}Y_{L1}Z_{L1}$in the space in which $L1$ is aligned with the vertical to its new representation $X_{L2}Y_{L2}Z_{L2}$ in the space in which $L2$ is aligned with the vertical:

$$\begin{bmatrix} X_{L2} \\ Y_{L2} \\ Z_{L2} \end{bmatrix} = R^{XYZ}_{X_{L2}Y_{L2}Z_{L2}} \cdot T^{RGB}_{XYZ} \cdot \left(T^{RGB}_{XYZ}\right)^{-1} \cdot \left(R^{XYZ}_{X_{L1}Y_{L1}Z_{L1}}\right)^{-1} \cdot \begin{bmatrix} X_{L1} \\ Y_{L1} \\ Z_{L1} \end{bmatrix} \tag{10}$$

As the transformation matrix T^{RGB}_{XYZ} and its reverse transformation $\left(T^{RGB}_{XYZ}\right)^{-1}$can be canceled out, we have

$$\begin{bmatrix} X_{L2} \\ Y_{L2} \\ Z_{L2} \end{bmatrix} = R^{XYZ}_{X_{L2}Y_{L2}Z_{L2}} \cdot \left(R^{XYZ}_{X_{L1}Y_{L1}Z_{L1}}\right)^{-1} \cdot \begin{bmatrix} X_{L1} \\ Y_{L1} \\ Z_{L1} \end{bmatrix} \tag{11}$$

$$\begin{bmatrix} X_{L2} \\ Y_{L2} \\ Z_{L2} \end{bmatrix} = R^{X_{L1}Y_{L1}Z_{L1}}_{X_{L2}Y_{L2}Z_{L2}} \cdot \begin{bmatrix} X_{L1} \\ Y_{L1} \\ Z_{L1} \end{bmatrix} \tag{12}$$

where $R^{X_{L1}Y_{L1}Z_{L1}}_{X_{L2}Y_{L2}Z_{L2}}$ is the desired $3x3$transformation matrix. This transformation matrix needs to be computed only once for a pair of illumination vectors $L1$ and $L2$. Subsequently, it can be applied to all the pixels in the image recorded under $L1$ in order to obtain the image corresponding to $L2$.

5. EXPERIMENTAL RESULTS

We will now present our experimental results that demonstrate the superiority of the $X_L Y_L Z_L$ color space in three areas of computer vision: color constancy, color image segmentation, and color object tracking. For image segmentation and object tracking, we will present comparative results to establish the superiority of the $X_L Y_L Z_L$ color space over the normalized *RGB* and the *HSI* representations. For color constancy, we will depend on the reader's subjective assessment of our results.

5.1. Color Constancy by Transforming the *RGB* Color Space into the $X_L Y_L Z_L$ Color Space

In **Section 4.2.1** we talked about the mathematical transformation that must be applied to a color vector in the $X_L Y_L Z_L$ space in order to recover the *RGB* values for the purpose of rendering an object on a terminal monitor. In that section, we also claimed that this transformation was meant to exhibit color constancy, in the sense that the displayed colors would be close to the white-illumination true color of an object surface and substantially independent of the illumination color. We will now demonstrate this property of the $X_L Y_L Z_L$ space with experimental results.

As shown in the top row of the Figure 10, a multi-colored plastic toy was photographed four times subject to four different illumination conditions: 1) florescent light; 2) blue colored incandescent light; 3) green colored incandescent light; and 4) tungsten light. Specifications of these lighting conditions are presented column (iv) of Table 3. The corresponding output images, as obtained from the $X_L Y_L Z_L$ representation of color values at each of the pixels and subsequent display of these transformed values via the reverse transformation described in **Section 4.2.1** (see Eq. 7), are presented in the second row of the same figure. As readers can see, the images in the second row exhibit much greater similarity to each other compared to the originally recorded images in the first row.

We also conducted an outdoor experiment. The "before" and "after" images shown in Figure 11 (a) and (b) respectively. In other words, the input image depicted in Figure 11 (a) was recorded under low-pressure sodium lighting (shown in the first row of Table 3). After the reverse transformation described in **Section 4.2.1**, is completed, the resulting output for color constancy is shown in Figure 11 (b).

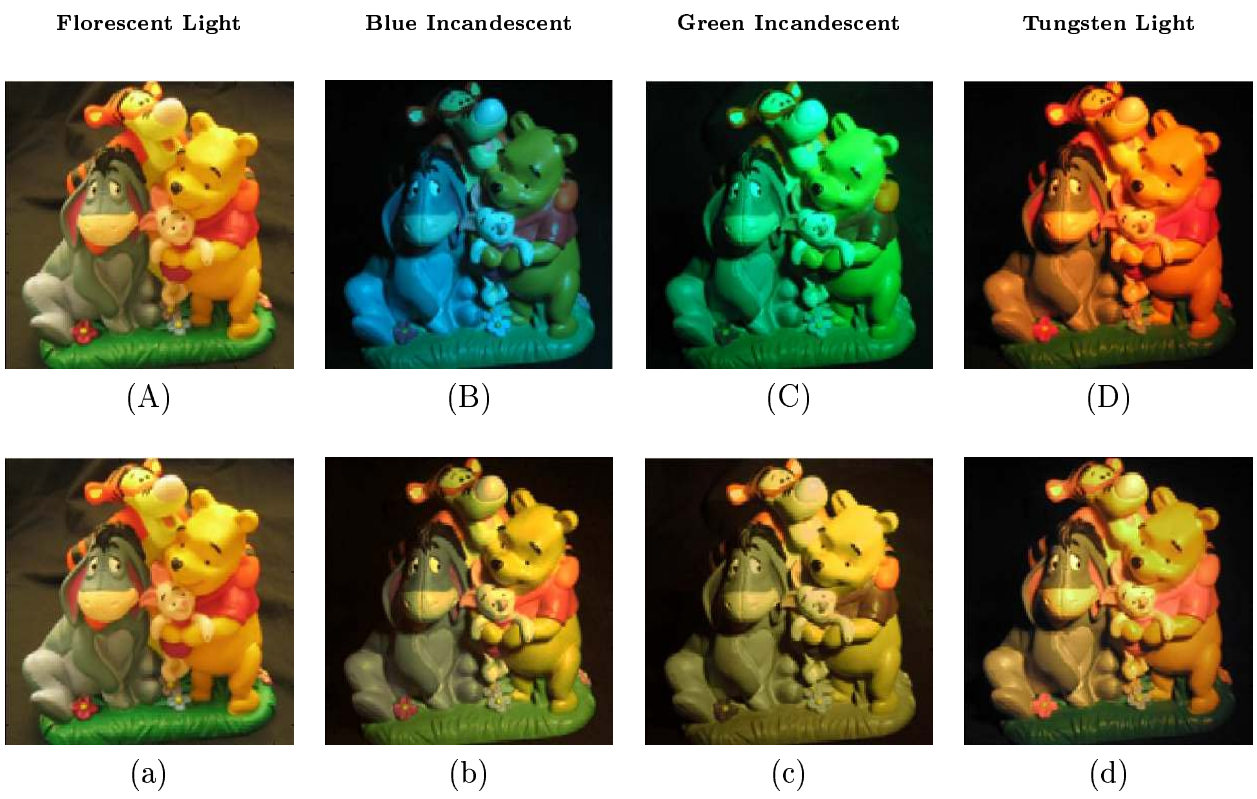


Figure 10: Experimental results of Illumination color adaptation.
(A) Input image (RGB) taken under *Florescent Lighting* (Table 3 (9)); (B) Input image (RGB) taken under *Blue Incandescent Lighting* (Table 3 (5)); (C) Input image (RGB) taken under *Green Incandescent Lighting* (Table 3 (4)); (D) Input image (RGB) taken under *Tungsten Lighting* (Table 3 (7)); and (a), (b), (c) and (d) are the compensated results of the input images (A), (B), (C) and (D) respectively, for the illumination-induced color changes using the $X_L Y_L Z_L$ color representation (see the reverse transformation described in **Section 4.2.1**).

We are also providing at the RVL web site[5] an interactive tool that allows a user to upload his/her image taken under non-white illumination for testing our color constancy claims. The remote user specifies the illumination color by clicking on a palette or by directly entering the RGB values of the dominant color component of the illuminant.

5.2. Image Segmentation in the $X_L Y_L Z_L$ Color Space

The success of many computer vision algorithms for scene analysis depends critically on the performance of the image segmentation step. If a vision algorithm is to work under different illumination conditions and if the scenes include objects of varying color, these colors must show sufficient illumination-invariant separability in the color space used so that the segmentation algorithm can distinguish between different

[5]http://rvl1.ecn.purdue.edu/RVL/newcolorspace

(a) (b)

Figure 11: Color constancy result (outdoor)
(a) The input image (RGB) taken under *Low-pressure Sodium Lighting* (at the parking garage as shown in Table 3 (1)); and (b) Color constancy obtained using the $X_L Y_L Z_L$ color representation (see the reverse transformation described in Section 4.2.1). Note that the pixels that are close to being black in the input image have been marked as "sky blue". The pixels are outside the range of values that can be accepted by the transformation.

object surfaces (or differently colored regions on the same surface) regardless of the illuminant color. If a subspace of the color space is used for color discrimination, adequate separability between the object colors must exist in that subspace. In the rest of this section, we will compare the segmentation results obtained using three different representations for color. For the sake of fair comparisons, all segmentations will be shown via clustering in an appropriate two-dimensional color subspace. The last subsection will discuss the segmentation of specular highlights in the $X_L Y_L Z_L$ color space.

5.2.1 A Comparison of Segmentations Achieved by Three Different Color Spaces

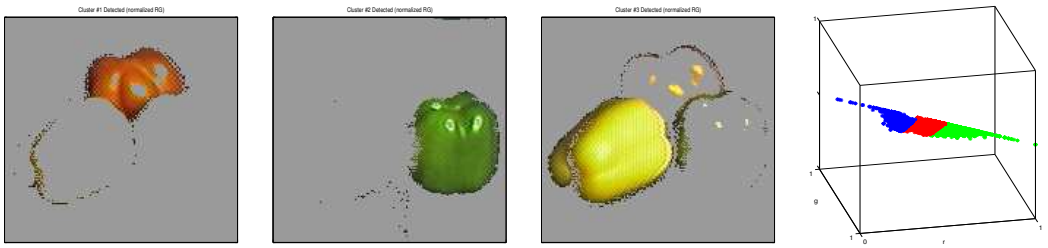
Using a test image, we will now present our evaluation of image segmentation using three different color representations – the normalized RGB and HSI color spaces and our new $X_L Y_L Z_L$ color representation. The segmentation results in all three cases will be shown on the input image of Figure 12 that was recorded under green incandescent illumination[6] whose parameters are described in the forth row of Table 3. In all three cases, we will use the same clustering algorithm – the venerated k-means algorithm [25] – to identify the pixels that belong together on the basis of similarity of color. To the extent possible, the various independently specifiable parameters of the algorithm will be kept the same in all three cases so as not to introduce any bias in our results.

The segmentation output using the normalized RGB representation for color is shown in Figure

[6]We should also mention that the RGB components of the illumination axis for the green incandescent illumination used are given by $L_R = 0.4351$, $L_G = 0.6888$, and $L_B = 0.5779$.

Figure 12: The input image

The recorded image of three colored peppers was taken with green incandescent lighting. The segmentation results shown in this section are for this input image.



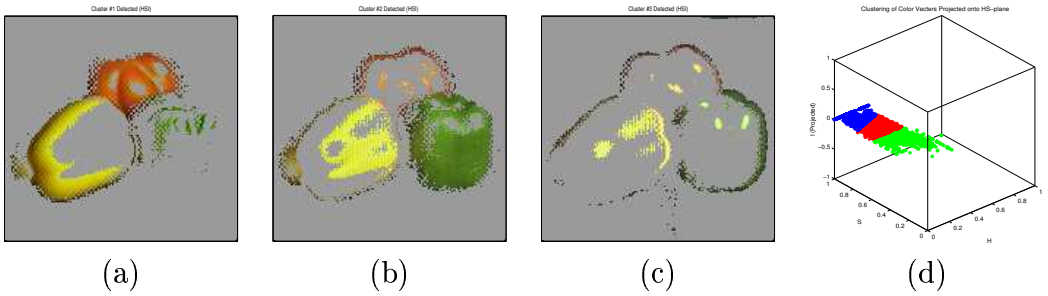
(a) (b) (c) (d)

Figure 13: Segmentation results using the normalized R and G

(a), (b), and (c) show the three output regions obtained from the segmentation when the color is represented using the normalized RGB representation. Shown in (d) are the three clusters identified in the rg-chromaticity plane. For the purpose of visual identification, the three different clusters in the rg-chromaticity plane have been arbitrarily assigned the red, blue, and the green colors.

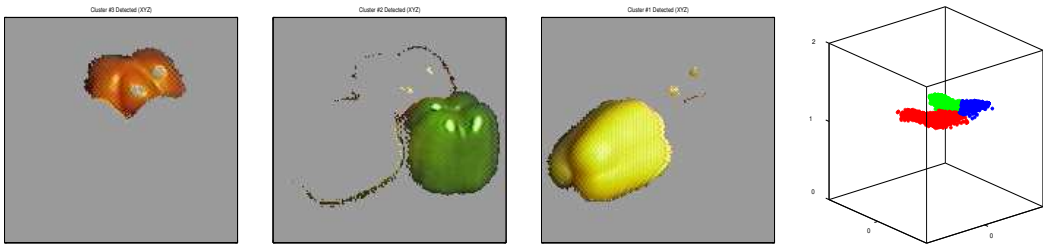
13. This segmentation was achieved by applying the clustering algorithm to the projection of the color vectors onto the rg-chromaticity plane. Since the normalized RGB representation has only two degrees of freedom, it makes sense to perform clustering of the data in two dimensions only and the rg-chromaticity plane is the plane of choice in much color computer vision. In its most straightforward implementation, the k-means algorithm requires the number of clusters – the value of k – to be specified in advance. We set k equal to three. Figures 13 (d) shows a color-coded 3-D map of the extracted clusters in the rg-chromaticity plane. Note that the colors used for the coding the aggregation of points in the rg-chromaticity plane have no connection with the colors in the input image. That is, the colors shown in (d) are merely for the visual identification of the clusters.

Figure 14 is the HSI version of Figure 13, meaning that with all other parameters held the same, the color in the input image was represented in the HSI color space and the clustering carried out in the HS color subspace, that is the HS-plane. The three segmented regions are shown in (a), (b), and



(a) (b) (c) (d)

Figure 14: Segmentation results using H (Hue) and S (Saturation)
The same as in Figure 13 except that now the color is represented in the *HSI* color space and the clustering carried out in the *HS* color subspace (the *HS*-plane).



(a) (b) (c) (d)

Figure 15: Segmentation results using X_L and Y_L in the $X_L Y_L Z_L$ color space
The same as in Figure 13 except that now the color is represented in the $X_L Y_L Z_L$ color space and the clustering carried in the $X_L Y_L$-plane.

(c), whereas, as before, (d) shows the three color-coded clusters projected onto the *HS*-plane.

Figure 15 shows the corresponding segmentation results using the $X_L Y_L Z_L$ color representation for color. In this case, the clustering was carried out in the $X_L Y_L$-plane. The three color-coded clusters are again shown in (d).

To summarize, the segmentation results achieved in both the *rg*-chromaticity space and the $X_L Y_L$-plane are superior compared to those obtained in the *HS* color subspace. As can be seen by the presence of yellow patches in the area of the red pepper, the inter-reflection between the peppers cannot be completely eliminated for the case clustering in both the *rg*-chromaticity space and the $X_L Y_L$-plane.

For quantitative evaluation of color separability in different color spaces, we also measured the following two parameters that are commonly used in linear discriminant analysis: 1) within-class variances (S_w), and 2) between-class variances (S_b). A global measure of the within-class variances for all the clusters can be calculated by [26]

$$S_w = \sum_{i=1}^{L} P_i E\left[(X - M_i)(X - M_i)^T | \omega_i\right] = \sum_{i=1}^{L} P_i \Sigma_I \tag{13}$$

and a global measure of the between-class variance, again for all the clusters, by

$$S_b = \sum_{i=1}^{L} P_i \left[(M_i - M_0)(M_i - M_0)^T\right] \tag{14}$$

where $M_0 = E[X] = \sum_{i=1}^{L} P_i M_i$ represents the expected mean of the mixture distribution. In other words, M_0 is the global mean for the color vectors in the subspace being used for clustering. We will use these formulas on a pairwise basis, that is, we will use them to measure the within-class and the between-class variances for each pair of clusters. We will therefore set $L = 2$. Also, typically, for the case of pairwise computations, the value of $P_i = 1/2$ is used. Since both Eqs. 13 and 14 produce matrices, it is more convenient to use the following scalar value as a measure of inter-class separability for the case of pairwise computations [26]:

$$J(C_i, C_j) = \frac{trace(S_b)}{trace(S_w)} \tag{15}$$

where $S_b(C_i, C_j)$ is the between-class variance (*i.e.*, the between-class scatter matrix) of two classes C_i and C_j and the $S_w(C_i, C_j)$ is the within-class variance for the mixture of two classes C_i and C_j. By definition, here, the larger $J(C_i, C_j)$ values, the better inter-class separability [26].

Table 4 shows the numerical values of the separability criterion in Eq. 15 obtained for each pair of color clusters in the three color subspaces used in this section for the input image of Figure 12. In this table, three classes, C_1, C_2, and C_3 correspond roughly to the clusters for the red, green and yellow peppers, respectively.

For the purpose of comparison with "ground truth", we performed the same clustering based segmentation on an image of the same scene that was recorded under white fluorescent illumination. The same inter-class separability measures $J(C_i, C_j)$ were computed for the three clusters extracted for each of the three color subspaces. These numbers are shown in red in Table 4.

The next subsection continues our discussion on color image segmentation by demonstrating how the $X_L Y_L Z_L$ color space facilitates the extraction of specular highlights in images.

		rg-Chromaticity Space	***HS*** Color Subspace	$X_L Y_L$ Color Subspace
Measurement of Inter-class separability $J(C_i, C_j)$	$\frac{S_b(C_1,C_2)}{S_w(C_1,C_2)}$:	$6.431e^{-4}$	$2.947e^{-4}$	$19.217e^{-4}$
		$(11.625e^{-4})$	$(4.126e^{-4})$	$(21.606e^{-4})$
	$\frac{S_b(C_1,C_3)}{S_w(C_1,C_3)}$:	$1.531e^{-4}$	$0.549e^{-4}$	$3.681e^{-4}$
		$(17.243e^{-4})$	$(4.271e^{-4})$	$(2.456e^{-4})$
	$\frac{S_b(C_2,C_3)}{S_w(C_2\,C_3)}$:	$5.251e^{-4}$	$5.639e^{-4}$	$5.754e^{-4}$
		$(4.792e^{-4})$	$(6.957e^{-4})$	$(4.823e^{-4})$

Table 4: Measurement of inter-class separability

The three classes C_1, C_2, and C_3 stand for the three clusters that correspond roughly to the red, green and yellow peppers, respectively. The table shows the inter-class separability measure $J(C_i, C_j)$ for each pair of clusters for the three color subspaces used in the segmentation experiments. The numbers in red are the corresponding values when the input image is recorded under white fluorescent illumination.

5.2.2 Segmentation of Specular Highlights in the $X_L Y_L Z_L$ Color Space

In general there are two types of specular pixels: 1) Unsaturated specular pixels and 2) Saturated specular pixels. Unsaturated specular pixels correspond to the surface reflectance component as defined in Eq. 9, whereas the saturated specular pixels result from clipping in one or more of the color channels. We can thus say that the unsaturated specular pixels follow the dichromatic reflection model, whereas the saturated specular pixels do not. Yet another way of saying the same thing is that a pixel may become specular either because the illumination intensity is high enough to cause clipping, or because the pixel is on the surface reflection "leg" of the color values that could possibly be reflected back from a point on the surface, or because of both these reasons.

As we mentioned previously in **Section 4.3**, the pixels produced by a scene in which all the object surfaces possess the same optical properties will lie in a dichromatic plane. In particular, the pixels will follow a two-legged distribution like the one shown in Figure 7 (a). The specular pixels in this distribution correspond to the surface leg and the clipped portions, as marked in the figure. In general, this figure applies to the *RGB* representation of the color. When the same figure is constructed in the $X_L Y_L Z_L$ space, the leg formed by the color clusters of the surface reflectance component becomes predominantly vertical, as shown in Figure 7 (b).

In what follows, we will now explain in greater detail the algorithmic steps for identifying the unsaturated color pixels – that is, the pixels that lie on the vertical portion of the two-legged curve in the dichromatic plane in the $X_L Y_L Z_L$ space. However, before doing so we must mention that Klinker

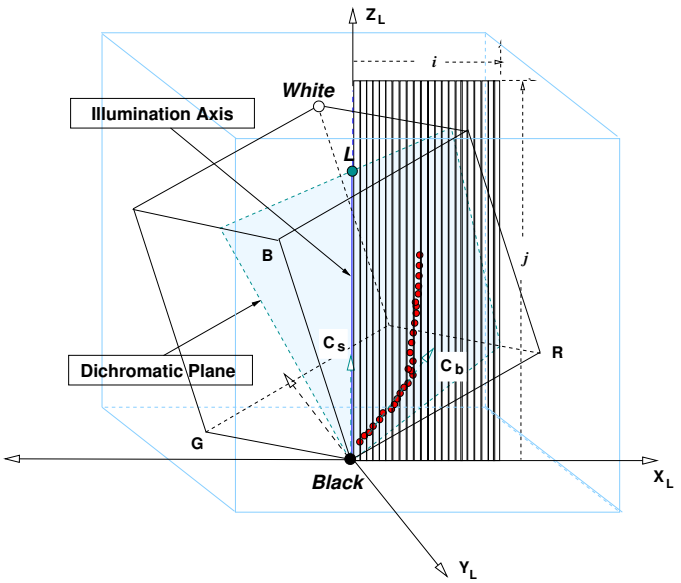


Figure 16: Detection of specular highlights from the dichromatic plane vertically located in the $X_L Y_L Z_L$ color space

[27] has used recursive line splitting for the same purpose. The recursive split tries to identify the point of maximum bend between the straight line fits to the two approximately linear segments of pixel values in the RGB space. However, the identification of this point becomes difficult if the slope difference between any two adjacent linear segments is not significant – a condition that is typically caused by the color of the illuminant being nearly the same as the color of the object surface.

We will now explain the algorithm we use to identify the unsaturated specular pixels in the $X_L Y_L Z_L$ color space. Our algorithm is based on dividing the dichromatic plane in the $X_L Y_L Z_L$ space into stripes parallel to the vertical and computing a weighted sum of the pixel counts in each of the stripes. The weighting factor assigned to each pixel equals its Z_L value. This weighting factor is based on the rationale that the higher the Z_L content of a color value, the greater the likelihood that it is a specular pixel.

To explain the algorithmic steps, let $C_s = [R_s, G_s, B_s]$, as defined in Eq. 9 and as shown in Figure 16, be the surface reflection vector, meaning that it corresponds to the surface reflections in the dichromatic model. The unsaturated specular pixels are on a line segment parallel to this vector. Again as stated previously, C_S is parallel to the illumination axis. Therefore, C_S will be along the vertical in the $X_L Y_L Z_L$ space.

To separate the specular reflection component from the body reflection component in the dichro-

matic plane, we first divide the dichromatic plane into n equal-spaced rectangular strips (*i.e.*, vertical bins) and for each strip compute the following sum

$$S(i) = \sum_{j=0}^{2} k \cdot Z_L(i,j) \tag{16}$$

where k is the number of pixels whose values correspond to the cell (i,j) in the dichromatic plane. In addition, for each strip i, we record the distance between the maximum and the minimum Z_L values, that is, $d_i = max_i(Z(i,j)) - min_i(Z(i,j))$. The unsaturated specular pixels can then be found the in the bin(s) that meet the following two criteria: 1) the d_i value that exceeds a given threshold[7] and 2) the largest sum $argmax_i\ S(i)$. The former criterion indicates if specular components exist. The specular pixels are then extracted by constructing a circular region of radius r around the bin in the X_LY_L plane that yields the maximum count and separating out all the pixels that fall in this circular region. The value of r depends on the the pixel value noise σ_0 which can be estimated from the camera noise. Following Klinker [27], we set r equal to $4\sigma_0$.

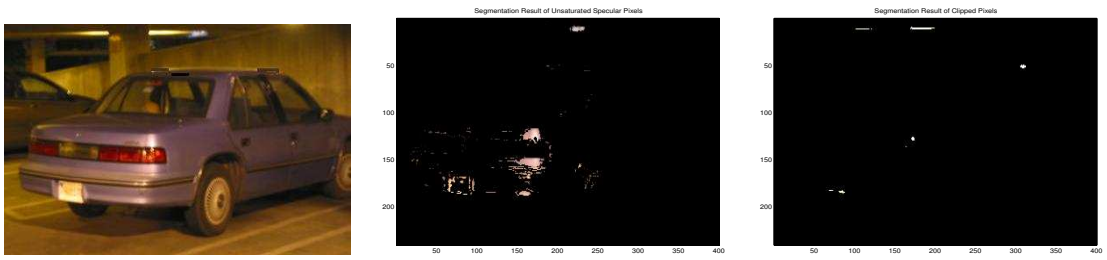
The previous discussion applied to the extraction of unsaturated specular highlights in images. Extraction of saturated specular highlights is straightforward since all we have to do is to apply a sufficiently high threshold to all the pixels to separate out those that are clipped.

Shown in Figure 17 (a) is the image of a blue car taken under low-pressure sodium lighting. Figure 17 (b) shows the regions of the image that contain unsaturated specular pixels; these regions were extracted by the algorithm just explained. We also show in (c) of the same figure the clipped pixels that correspond to the saturated specular highlights; these were extracted by applying a threshold value of 253 equally in all of the three color channels.

5.3. Color Object Tracking using Three Different Color Representations

We will now present a comparative study of object tracking using just the color content of the object to be tracked. The study was carried out using the following three color spaces: 1) the *rg*-chromaticity space; 2) the *HS* color subspace; and 3) the X_LY_L color subspace. For the *rg*-chromaticity space, we used the normalized R and the normalized G values. The H (Hue) ans S (Saturation) values were

[7]For our experiments, we set $d_i = 0.2$, that is a tenth of the maximum variation in the direction of of the Z_L-axis.



(a) (b) (c)

Figure 17: Segmentation results.
(a) The input image (RGB) taken under *Low-pressure Sodium Lighting* (at the parking garage as shown in Table 3 (1)); (b) Unsaturated specular highlights detected; and (c) Clipped (saturated) specular pixels.

used in the HS color subspace. And, for the case of the $X_L Y_L$ color subspace, we used the values $D = \sqrt{X_L^2 + Y_L^2}$ and $O = arctan2(Y_L/X_L)$, which as mentioned in Section 4.2, are the illumination-adapted saturation and the illumination-adapted hue, respectively.

Since the focus here is on just the color cues, we used a color histogram-based tracking algorithm in all three cases. This algorithm constructs a histogram of the color values coming off the object to be tracked in a scene and then basically updates and follows the center of mass of this histogram during tracking. The histogram is "learned" in the first frame of a video sequence with the help of a human delineating a rectangular region containing the object of interest. The center of mass of this rectangular region becomes the initial location of the object in the image. Subsequently, as the object moves, a search for a new object region is carried in a portion of the image that is usually twice the size of the initially delineated region and that surrounds the initial region. This search yields a new object region whose histogram is most similar to what was learned in the first frame. Further details on this approach can be found in [13].

Our data consisted of three image sequences, each for a different object to be tracked and each consisting of 60 images recorded at the rate of 10 frames per second with the blue incandescent lighting described in Table 3 (5). Data was collected with a progressive scan CCD camera (Jai CV-M70) without automatic gain control and without white balancing. The objects used in the three videos are: a yellow toy car and an orange box.[8] For quantitative evaluation, we use a set of *human-provided* ground-truth constructed by classifying all pixels in each image as either foreground or background. The human

[8]Additional results of these tracking experiments using various objects and illuminations are available at the web site of Purdue University Robot vision Laboratory ($http://shay.ecn.purdue.edu/jbpark/demo1.htm$).

marking of the ground truth is facilitated by using the *GIMP* (the Gnu Image Manipulation Program) image editing tool. In each image of a video sequence, the human marks out the region that best corresponding to the object region used for learning the color distribution in the first frame.

Tracking in these experiments is demonstrated by displaying a *tracking rectangle* that for successful tracking should move with the object while keeping the initially chosen object region enclosed within itself. Shown in the first column of the top row of Figure 18 is the human-delineated rectangle on a yellow toy car. After the color distribution in this rectangle is learned, the tracking algorithm tries to follow this rectangle as the object moves. Shown in Figure 18 are four other frames where the tracking rectangle is color-encoded separately for each of the three color spaces. The other columns in Figure 18 display the search area, which as already mentioned, is twice as wide and twice as high for the frames shown and the object pixels detected inside each search region. A pixel within the search rectangle is declared to be an object pixel with the help of a lookup table constructed during the first learning frame. The second column shows the object pixels detected when tracking is carried out using the *rg*-chromaticity space for representing the color values, the third column the same for tracking with the *HS* color subspace, the last for the case of tracking with the $X_L Y_L$ color subspace. Shown below each frame are a pair of parenthesized numbers. The first indicates the number of correctly detected pixels (True Positives) that belong to the target object – based on the human-provided ground-truth – and the second the number of pixels that actually belong to the background but are declared to be object pixels (*i.e.*, foreground pixels) by the algorithm (False Positives).

In a sequence of 60 frames of the yellow toy car (Figure 18), the average number of pixels per frame correctly detected (True Positives) by our tracking algorithm running in the *rg*-chromaticity space is 1123.7. For the same sequence, *the HS color subspace* produces the average value of 981.5. The average for the $X_L Y_L$ subspace is 2598.2. Comparing true positives only, the $X_L Y_L$ subspace produces a detection rate that is roughly 29.88% higher than what we get with the *rg*-chromaticity space, which in turn is higher by 34.59% compared to the result obtained with the *HS* color subspace. Comparing the False Positives, the $X_L Y_L$ yields the value of 117.2 pixels per frame, whereas the *rg*-chromaticity space yields an average value of 60.7, and *HS* color subspace the average of 64.6. The false positive rate for the $X_L Y_L$ color subspace is 1.23% lower than that for the *rg*-chromaticity, which itself is lower by 1.19% compared to the *HS* color subspaces. False Positives are a result of any color similarities, as

Color Spaces	***Normalized RGB***	***HSI***	$X_L Y_L Z_L$
Parameters Used	***rg***	***HS***	*DO*
Yellow Toy Car	DR: 37.95 %	DR: 33.24 %	DR: 67.83 %
	FPR: 6.29 %	FPR: 6.25 %	FPR: 5.06 %
Orange Box	DR: 23.93 %	DR: 26.10 %	DR: 44.54 %
	FPR: 7.72 %	FPR: 18.94 %	FPR: 1.05 %

Table 5: Detection rates (DR) and False Positive rates (FPR)

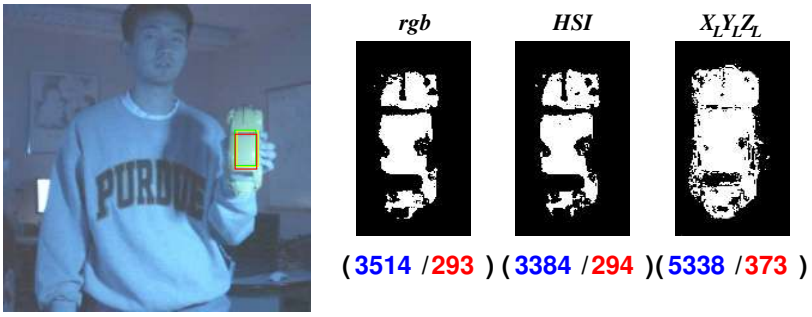
Detection Rate ($DR(\%)=\sum_{i=1}^{n} TP_I \cdot 100/\sum_{i=1}^{n}(TP_I + FN_i)$) and False Positive Rate ($FPR(\%)=\sum_{i=1}^{n} FP_i \cdot 100/\sum_{i=1}^{n}(FP_i + TP_i)$), where TP_i and FP_i are the numbers of pixels that are ***correctly*** and ***incorrectly*** detected as foreground pixels in the image i, respectively; and FN_i is the number of pixels that are ***incorrectly*** declared as background pixels in the image i, for the three tracking experiments (for each experiment, the number of image frames, $n = 60$) under the blue incandescent lighting.

measured in the different color spaces, between the foreground and the background pixels. For example, as shown in the right-most image of Figure 18, the pixels the belong to the fingers of the person holding the toy car were counted as the false alarm. Note that False Negatives (FN) and True Negatives (TN) are of no interest here in this tracking experiments, especially as the system loose track in the many frames [28].

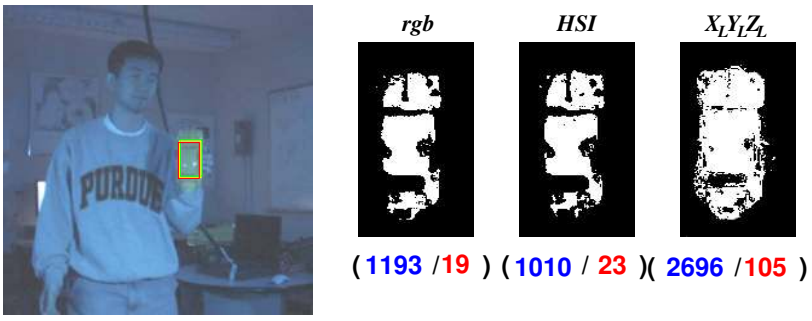
Table 5 summarizes both *detection rates* (DR) and *False Positive rates* (FPR) recorded in the two tracking experiments with yellow toy car, and orange box under blue incandescent lighting.

6. CONCLUSIONS

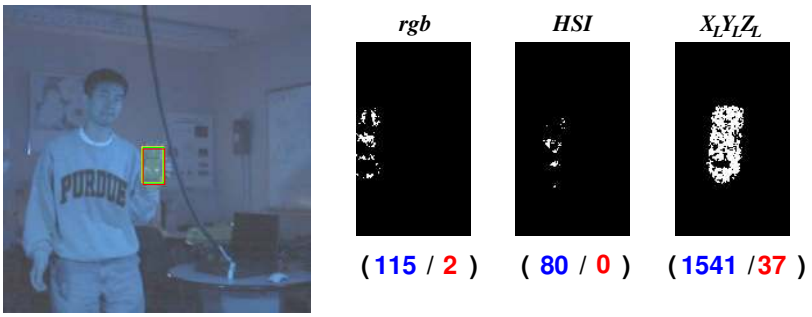
In this paper, we have presented a new color representation scheme that adapts to the color of the illuminant. The transformation required by this representation aligns the vertical axis of the color space with the color of the illuminant that is estimated in the RGB color cube. The other degrees of freedom of the new color space are determined by an *RGB to HSI-like* transformation, *but with respect to the color of the illuminant.* A combination of how we choose the vertical axis and then how the color space is rotated around the vertical allows us to introduce the notions of illumination-adapted hue and illumination-adapted saturation for the characterization of object color. Another practical benefit of these transformations is that dichromatic plane now acquires a single-parameter characterization, which is the rotational angle around the vertical. This paper also presented a simple procedure for estimating the color of the illuminant from a set of image patches recorded for small tiles placed at



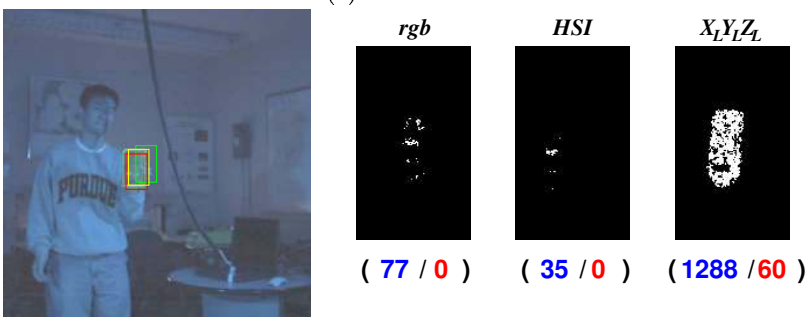
(a) Frame 001



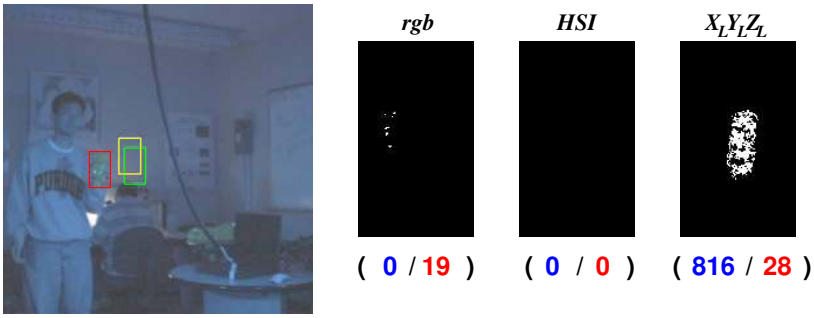
(b) Frame 024



(c) Frame 038

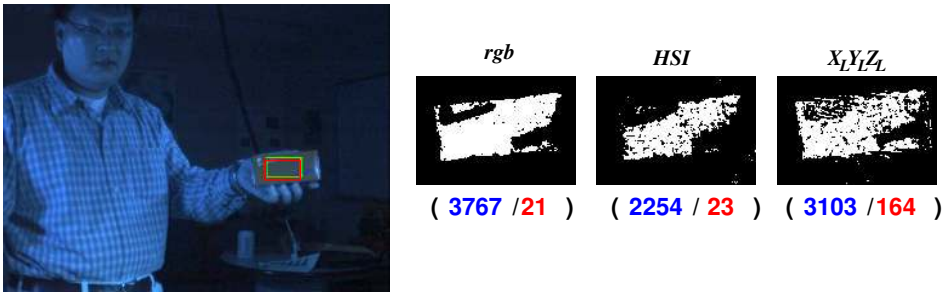


(d) Frame 045

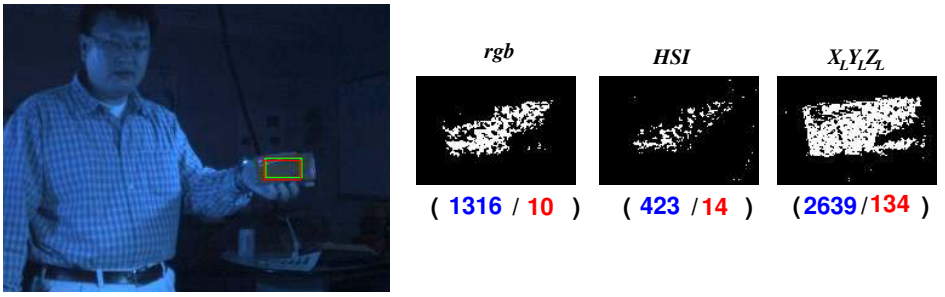


(e) Frame 059

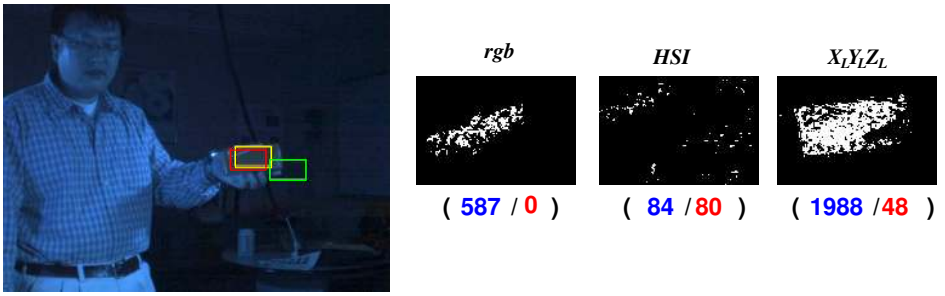
Figure 18: Comparisons of the tracking results (Yellow Toy Car)



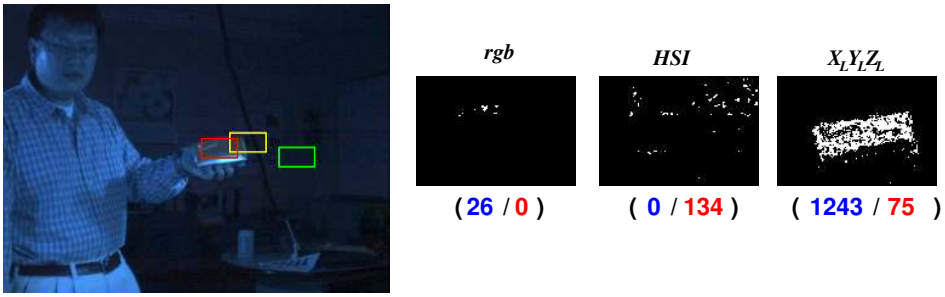
(a) Frame 004



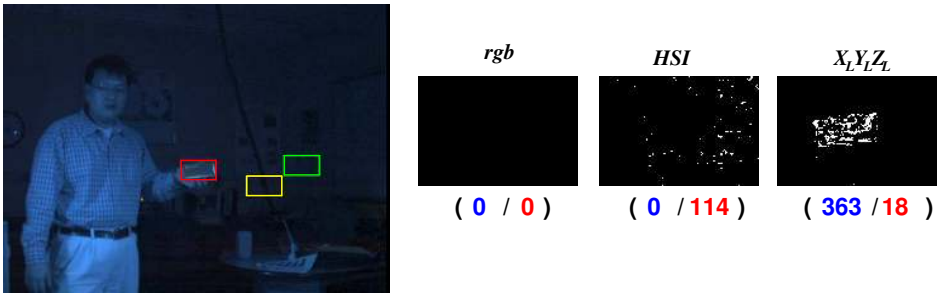
(b) Frame 007



(c) Frame 012



(d) Frame 018



(e) Frame 039

Figure 19: Comparisons of the tracking results (Orange Box)

different locations with respect to the source of the illumination. The experimental results shown on color-constancy, color image segmentation, and color object tracking demonstrate the usefulness of the new color representation.

ACKNOWLEDGEMENT

The authors are grateful to the Staff of the Advanced Manufacturing Technology Department, Ford Motor Company, for proving continuous supports and feedback on this study. Special thanks to Guilherme Nelson DeSouza and Youngrock Yoon for their help with the object tracking experiments.

References

[1] B. V. Funt and G. D. Finlayson, "Color constant color indexing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, pp. 522–529, May 1995.

[2] S. K. Nayar and R. M. Bolle, "Reflectance based object recognition," *International Journal of Computer Vision*, vol. 17, no. 3, pp. 219–240, 1996.

[3] T. Gevers and A. Smeulders, "Color-based object recognition," *Pattern Recognition*, vol. 32, no. 3, pp. 453–464, 1999.

[4] B. V. Funt and M. S. Drew, *Color Space Analysis of Mutual Illumination.* Technical Report CSS/LCCR TR 91-93, Simon Fraser University, Center for Systems Science LCCR, Burnaby, B. C., Canada, Apr., 1991.

[5] S. Shafer, "Using color to separate reflection components," *Color Res. Appl.*, vol. 10, pp. 210–218, 1985.

[6] T. B. Phong, "Illumination for computer generated images," *Communications of the AMC*, vol. 18, no. 6, pp. 311–317, 1975.

[7] J. Yang and A. Waibel, "A real-time face tracker," in *Proceedings of IEEE Workshop on Applications of Computer Vision*, pp. 142–147, 1996.

[8] K. Schwerdt and J. L. Crowley, "Robust face tracking using color," in *Proceedings of the 4th IEEE International Conference on Automatic and Face Gesture Recognition*, Mar. 2000.

[9] Y. Wu and T. S. Huang, "Color tracking by transductive learning," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. I, pp. 133–138, 2000.

[10] H. W. Qian Chen and M. Yachida, "Face detection by fuzzy matching," in *Proceedings of the 5th IEEE International Conference of Computer Vision*, pp. 591–596, 1995.

[11] K. Sobottka and I. Pitas, "Segmentation and tracking of faces in color images," in *Proceedings of Automatic Face and Gesture Recognition*, pp. 236–241, 1996.

[12] S. Feyrer and A. Zell, "Detection, tracking, and pursuit of humans with an autonomous mobile robot," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 864–869, 1999.

[13] G. R. Bradski, "Computer vision face tracking for use in perceptual user interface," *Intel Technology Journal*, vol. 2, no. 2, pp. 12–21, 1998.

[14] B. J. Y. Y. B. Lee and S. W. Lee, "A real-time color-based object tracking robust to irregular illumination variations," in *Proceedings of the IEEE International Conference on Robotics and Automation*, May 2001.

[15] T. J. D. C. R. Wren, A. Azarbayejani and A. P. Pentland, "Pfinder: Real-time tracking of the human body," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 780–785, Jul. 1997.

[16] J. R. Smith and S. F. Chang, "Local color and texture extraction and spatial query system," in *Proceedings of the IEEE International Conference on Image Processing*, Sep. 1989.

[17] B. S. G. D. Finlayson and J. L. Crowley, "Comprehensive color image normalization," in *5th European Conference on Computer Vision*, pp. 475–490, 1998.

[18] K. E. T. S. N. Pattanaik, J. A. Ferwerda and D. P. Greenberg, "Validation of global illumination simulations through ccd camera measurements," in *Proceedings of the 5th Color Imaging Conference, Society for Imaging Science and Technology*, pp. 250–253, Nov. 1997.

[19] S. D. Buluswar and B. A. Draper, "Color models for outdoor machine vision," *Computer Vision and Image Understanding*, vol. 85, pp. 71–99, 2002.

[20] S. E. Shoij Tominaga and B. A. Wandell, "Scene illuminant classification: Brighter is better," *Journal of Optical Society of America, A*, vol. 18, pp. 55–64, Jan. 2001.

[21] S. Tominaga, "Surface identification using the dichromatic reflection model," *PAMI*, vol. 13, pp. 658–670, Jul. 1991.

[22] H.-C. Lee, "Method for computing the scene-illuminant from specular highlights," *Journal of Optics Society of America*, vol. A(3), no. 10, pp. 1694–1699, 1986.

[23] T. M. Lehmann and C. Palm, "Color line search for illuminant estimation in real-world scene," *Journal of the Optical Society of America*, vol. 18, no. 11, pp. 2679–2691, 2001.

[24] K. N. R. T. Tan and K. Ikeuchi, "Illumination chromaticity estimation using inverse-intensity chromaticity space," in *CVPR*, pp. 673–680, 2003.

[25] J. McQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the 5th Berkeley Symposium on Math. Stat. and Pro.*, vol. 1, pp. 281–296, 1967.

[26] K. Fukunaga, *Intorduction to Statistical Pattern Recognition.* 2nd Edition, Academic Press, New York, 1990.

[27] G. J. Klinker, *A Physical Approach to Color Image Understanding.* AK Peters, LTD., Wellesley, MA., 1990.

[28] M. T. B. Georis, F. Bremond and B. Macq, "Use of an evaluation and diagnosis method to improve tracking performances," in *Proceedings of VIIP'03 - the 3rd IASTED International Conference on Visualization, Imaging and Image Processing*, pp. 827–832, Sep., 2003.