Intrinsic Images for Shadow Removal

Jordan Frank
School of Computer Science
McGill University
Montreal, Quebec, Canada
jordan.frank@cs.mcgill.ca

***Abstract*— The purpose of this report is to summarize the work done by a number of researchers over the course of a number of years related to creating illumination invariant images for the purpose of shadow removal. We will also discuss the experiments that were implemented in order to test the algorithm on consumer-grade digital cameras, and the results of these experiments.**

I. Introduction

Over the last eight years or so, a large amount of research has been undertaken by a Graham D. Finlayson and Steven D. Hordley from the University of East Anglia in Norwich, and Mark S. Drew and Cheng Lu from Simon Fraser University in Vancouver, British Columbia, to study methods for creating invariant images from a single digital image. An invariant image is one that is independent of lighting, and therefore has shading removed. These invariant images are also referred to as intrinsic images in the literature, as they capture the intrinsic reflectivity of the objects in the image, independent of illumination conditions. This work is embodied in a number of papers that have been published by the aforementioned authors, and we will be focusing on three main papers, [2], [3], and [1].

In Section 2 of this report, we discuss the theoretical underpinning of the method for creating the invariant images. The procedure for finding the parameters of the model in practice will be discussed in Section 3. In Section 4 we will discuss the experiments that we performed in order to test the methods and the results, and in Section 5 we will conclude the report with a brief discussion and summary of work.

II. Background

A. Color Constancy

Color constancy has been studied for decades by many researchers, and plays an extremely important role in many computer vision applications. An image is made up of a set of pixels, and each pixel reflects the light that has been received at a sensor in a digital camera or other optical sensor. The light that hits the sensor is made up of two main components, the illumination component that reflects the colour and intensity of the illuminant, and the reflectance component which captures the surface reflectance properties, namely the color of the object reflecting the light. It is these two components that define the properties of the light that hits the sensor. The sensor then converts the light into a digital signal, adding in a certain amount of noise. Color constancy is concerned with finding an intrinsic measurement of the color of the surface at a pixel in such a way that the measure will be the same for that same surface independent of any illumination effects, and is useful for recognizing specific colors across images, or even within the same image if the lighting conditions vary within the image. Simply put, we wish to find a way of specifying that a red ball has a specific color value that it will retain in any image of the ball under any lighting condition. Color constancy then can be used for object recognition tasks, or for more specialized tasks such as shadow removal, which is what we will be concerning ourselves with.

Color is often analyzed by considering three separate channels, red, green and blue, and for the purposes of this report we will use this RGB color model. The intensity, then, of each of these three color channels can be described by

$$R_k = \int E(\lambda)S(\lambda)Q_k(\lambda)\mathrm{d}\lambda, k = R, G, B, \quad (1)$$

with the illumination spectral power distribution $E(\lambda)$, the surface spectral reflectance function $S(\lambda)$, and the camera sensor sensitivity functions $Q_k(\lambda)$. The integral is usually taken over the visible wavelengths of light, and we then form the RGB color $\mathbf{R} = \{R_R, R_G, R_B\}$ at a pixel.

In order to simplify this equation and remove the integral, the authors note that the camera sensor $Q_k(\lambda)$ behaves similar to a Direc delta function $Q_k(\lambda) = q_k\delta(\lambda - \lambda_k)$, where q_k represents the sensor strength $q_k = Q_k(\lambda_k)$. The authors also note that the illumination function E_k can be approximated by Planck's law

$$E(\lambda, T) = Ic_1\lambda^{-5}(e^{\frac{c2}{T\lambda}} - 1)^{-1} \quad (2)$$

where c_1 and c_2 are constancts, I controls the intensity of the light, and T is the temperature of the light. For the typical temperature ranges of most light sources, Wein's approximation [4] allows us to further reduce the illumination equation to

$$E(\lambda, T) = Ic_1\lambda^{-5}e^{-\frac{c2}{T\lambda}} \quad (3)$$

Now plugging the Dirac delta estimate as well as equation 3 back into equation 1, we get the narrow-band sensor response equation

$$R_k = Ic_1\lambda_k^{-5}e^{-\frac{c2}{T\lambda}}S(\lambda_k)q_k. \quad (4)$$

The most important contribution of [2] is contained in following observation. If we now divide any two color channels and form the band-ratio 2-vector chromaticities

$$c_k = R_k/R_p \quad (5)$$

where p is fixed to one color channel, and k indexes over the other two channels, then the effects of the illumination intensity, I, is removed since it is a constant value at each pixel for all three color channels. For example, we could choose to use the green channel as the divisor, giving us $\mathbf{c} = \{R_R/R_G, R_B/R_G\}^1$. Now if we take the log of these c_k values, with $s_k \equiv c_1\lambda_k^{-5}S(\lambda_k)q_k$ and $e_k \equiv -k_2/\lambda_k$, we obtain

$$\rho_k \equiv \log(c_k) = \log(s_k/s_p) + (e_k - e_p)/T. \quad (6)$$

Looking closely at equation 6, we see that it is the equation for a straight line parameterized by T. The ratios s_k/s_p are dependent of the surface, without any illumination effects, and the 2-vector direction $(e_k - e_p)$ is independent of the surface. This means that the offset for the line is based on the surface, but the direction is independent of the surface. Therefore, if we project the 2D logs of chromaticity, our ρ_ks into the direction $\mathbf{e}^\perp$ orthogonal to the vector $\mathbf{e} \equiv (e_k - e_p)$, then we will get a single scalar value that represents the intrinsic color of the surface captured by the pixel, completely removing the effects of illumination. Since shadows are created by differences in intensity and color (temperature T) of the lighting between various regions in an image, they are effectively removed through this projection.

In [1], a variation on this method is introduced which creates a 2D invariant chromaticity image rather than just a 1D invariant image. The authors make the observation that the quality of the 1D invariant image is dependent on the color channel that is chosen as the divisor. Instead of picking a single color channel to divide by, the authors instead amend the definition of equations 5 and 6 as follows:

$$c_k = R_k/\left(\prod_{i=R,G,B} R_i\right)^{1/3} = R_k/R_M \quad (7)$$

and the log version

$$\rho_k = \log(c_k) = \log(s_k/s_M) + (e_k - e_M)/T \quad (8)$$

for $k = R, G, B$, with s_k, s_M, e_k, and e_M updated accordingly. Therefore, for each pixel we get a coordinate in 3-space, $\rho = \{\rho_R, \rho_G, \rho_B\}$. However, since we are dividing by a constant, R_M, all of the values of ρ will lie on a plane orthogonal to $\mathbf{u} = 1/\sqrt{3}(1, 1, 1)^T$, and so we can easily project these points onto a 2D space, which is refered to in the paper as the Geometric Mean 2D Chromaticity Space. Then, from the 2D projection, each point is given by two coordinates, $\{\chi_1, \chi_2\}$. Then instead of finding the vector $\mathbf{e}$ onto which we project the points, we can consider rotating the points in

[1]We have chosen to divide each channel by the green channel, this decision was made arbitrarily, and any channel can be chosen as the divisor.

2D space by an amount θ, and then projecting the points onto the horizontal axis to form a 1D greyscale image

$$\mathcal{I} = \chi_1 \cos\theta + \chi_2 \sin\theta \tag{9}$$

If we can find the proper value of θ by which to rotate the data, we can tranform the data into a 2D space where the horizontal axis corresponds to the surface reflectance properties, and the vertical axis accounts for the illumination. Therefore, projecting down onto the horizontal axis gives a 1D measure that is independent of any illumination effects.

B. Camera Calibration

In order to create our invariant image, we need to determine the correct value of θ, or the vector $\mathbf{e}$ on which to project the data such that the effects of illumination are removed. Through experimentation, the authors determined that this parameter is dependent on the camera being used, but once it is found, it can be used to create invariant images for any image generated by the camera. The simplest way to calibrate a camera then is to take a number of pictures of the same surface under different lighting conditions. [2] describes the algorithm for computing the required parameters from these images, but for the sake of conserving space, and since we will not be using this method, we will leave the interested reader to locate the details in the original paper.

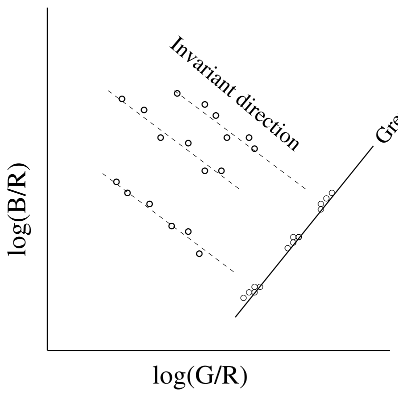

Fig. 1. Intuition for finding best direction by minimizing entropy

We would like, then, a method for determining the parameters from a single image with an uncalibrated device, and this is the main contribution of [1]. In this paper, the authors describe a method that uses entropy minimization to find the optimal projection vector. Figure 1 is from the paper, and gives us the intuition behind the method, that the entropy of our resulting projected image will give a useful measure for how good the direction on which we project is. What is not mentioned in the paper, but plays a vital role as far as the method is applicable in practice is that this method really works best when the number of different invariant colors in an image is low.

Equation 9 gives us a greyscale image $\mathcal{I}$ representing the invariant image, but we cannot simply calculate the entropy of a 1D vector. In order to calculate the entropy, we create a normalized histogram of the values in $\mathcal{I}$, using Scott's Rule

$$\texttt{bin\_width} = 3.49\texttt{std}(\mathcal{I})N^{1/3} \tag{10}$$

where $\texttt{std}(\mathcal{I})$ is the standard deviation of the components of $\mathcal{I}$, and N is the size of the invariant image data for the current angle. The reason for the N term is that the authors exclude the top and bottom 5% of the range values in order to eliminate noise, and so a different number of values are excluded for each angle θ.

C. Removing Shadows

Once we have an invariant image, we would like to reintegrate this shadowless image back into the original image in such a way that we preserve the original colors and contrasts, but we remove the shadows and other effects of illumination[2]. The general idea presented in [3] is to create two edge maps, one from the original image, and one for the invariant image. Then, edges that appear in the original image edge map, but not in the edge map for the invariant image can be assumed to be due to illumination effects, hence they are shadow edges. The authors then create a gradient map for each of the color channels for the original image, and then alter the map to zero out the gradients at the shadow edges. By then reintegrating this gradient map, taking into consideration the boundary conditions by essentially using the values from the original image, the resulting image will have the same colors and contrast as the original image, but

[2]Though it is not addressed in any of the cited papers, it would seem that this technique would effectively remove specularities, and other illumination effects other than just shadows.

with the shadows removed. In [1], the method is further refined to use the invariant chromaticity image that is developed in the paper, as well as using a few other tricks such as taking the edge map of a Mean-Shift processed original image, and using a form of in-filling to grow edges into the shadow-edge regions, rather than simply zeroing the gradient at the shadow edges.

For the purpose of this report, we were more interested in investigating the process of creating the invariant image, and so we did not go much further with the shadow removal process other than to understand the methodology.

III. Algorithm

As stated in the previous section, we were concerned mostly with the process of creating invariant images, not the process of removing shadows, and so we will only go into detail on the algorithm for generating the invariant images through entropy minimization.

The main idea behind the process for finding the invariant image is summarized in the following algorithm.

```
Form a 2D log-chromaticity representation of the image.
for θ = 1..180 do
    Form greyscale image I: the projection onto 1D direction.
    Calculate top and bottom 5th percentile of range of values and remove.
    Calculate bin width using Scott's Rule.
    Form histogram for I and calculate entropy.
    Keep track of minimum entropy.
end for
Min-entropy direction is correct projection.
```

The algorithm is fairly simple, but there are a number of details that are hidden by using pseudocode, which we will discuss in the next section.

IV. Experiment and Results

In [1], the authors claim that the method described worked on all of the cameras that they tested, and so the purpose of our experiments were to test if the claim would hold even for consumer-grade digital cameras. The cameras that the authors used in their experiments were very expensive, professional cameras. The goal then was to see if we could replicate the results using this author's Pentax Optio© 6.0 megapixel digital camera.

We set out on a sunny day and took a number of pictures of scenes that had predominant shadows. It turned out that the most important property of the shadows in the photographs was that they were not too dark. This is due to the fact that if the pixels in the shaded areas are too dark, it is nearly impossible to differentiate the pixels from black, and so getting any color information out of these pixels becomes difficult, and the algorithm fails to produce even somewhat reasonable results.

We implemented the algorithm in the previous section using both the standard log-chromaticity space in equations 5 and 6, as well as the geometric mean 2D chromaticity space in equations 7 and 8, and it turned out that the differences were barely noticeable. At this point it is important to note that for an experiment such as this, we are trying to generate images in which the effects of illumination are eliminated, but this is not something that can be reasonably quantified. Therefore, the results are strictly qualitative, and we do not have empirical error measurements. If we were somehow able to photograph a scene with no illumination, which is impossible since photographs require capturing reflected light, then we could compare these images to our results, but because such a thing is not possible, we have to look at the results and give some subjective measure of how good we are doing.

After running the algorithm on all 28 of the photographs from our digital camera, the results were so poor that we figured there must be a bug in the algorithm. To test the algorithm, we tried running it on the images that were used in the papers. Unfortunately, these images were not publicly available, and so we resorted to taking low resolution screen-captures of the images from the PDF versions of the paper, which resulted in poor copies of the images. Surprisingly though, the algorithm was able to produce excellent results, not quite as good as in the papers, but quite close, and certainly much better than the results from our own photographs.

At this point it seemed clear that the problem was in the images produced by the camera. It was not a problem of detail or resolution, since even the low quality images taken from the PDF for the paper worked well, it was a problem of the colors in the image. We investigated further by looking at the photographs in more detail using the program Adobe Photoshop©. Photoshop© allowed us to look at the images in higher detail, and inspect statistics of the image, such as the color histogram. It turned out that in our images, the contrast in the shaded regions was much higher than in the bright regions, while in the image from the paper, the contrast did not vary nearly as much between the shaded and bright regions.

We discussed this problem with a fellow classmate, Boris Oreshkin, who informed us that this problem is actually a feature of most digital cameras. In order to produce visually appealing images, the cameras artificially boost the color contrast in darker regions. This explanation agreed with our observations, and provided a reasonable explanation for the poor results.

Next we attempted to adjust the algorithm in order to compensate for this problem. We looked at the various invariant images that were produced as we rotated the log-chromaticity image from 1 to 180 degrees. It turned out that some of the images that were produced that had much higher entropy than the minimum entropy image were actually much better and had many of the shadows removed. What this meant was that entropy was not in fact a good indicator of how good our projection was for our testing images. Unfortunately we were unable to come up with a better measure than entropy, and so we were unable to find an automatic way of determining how good the invariant image was, and had to resort to manual inspection.

The next logical step seemed to be to try to manually calibrate our camera, that is to take a number of pictures of uniformly colored patches under various lighting conditions, plot the mean log-chromaticities, and then visually inspect the result to determine the direction by which to project the data in order to remove the effects of the illumination. Figure 2 shows the log-chromaticities of 4 different colored patches, red, blue, white, and yellow, under 14 different illumination conditions. From the figure, it is apparent that there is in fact

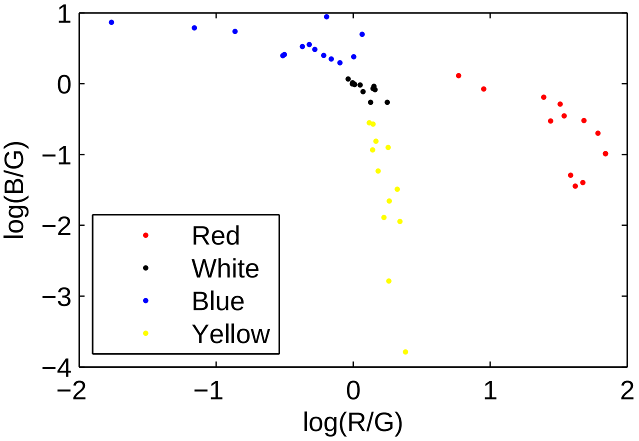

Fig. 2. Log-chromaticities of 4 different colored patches under 14 different illumination conditions

no predominant direction on which the data falls, no angle that we could rotate the data by in order to cause the data to be vertically aligned within each color, as there was in the papers. For each of the patches, the points do seem to vary somewhat linearly, but while the blue, red, and white points follow one direction, the yellow points seem to follow a completely different, nearly perpendicular direction.

From this previous experiment, it is clear that the intensity of pixels in images produced by this camera either do not conform to the theoretical assumption in equation 1, the camera sensors are not narrow-band enough for this method to work, or there is simply too much noise in the images due to the quality of the sensors in the camera. Therefore, the results of our experiments were quite poor. It seems most likely that the largest problem is that the sensors are not narrow-band enough, and respond quite differently to different wavelengths of light.

Figures 3, 4, 5, and 6 show the results for one of the test images. The algorithm detected that a rotation angle of 20 degrees produced the minimum entropy invariant image. It is clear from figure 6 that the minimum-entropy invariant image still contains the shadows of the car, meaning that the result is quite poor. From the log-chromaticity plot, figure 4, it is quite evident that a large number of the pixel seem to lie in a somewhat horizontal direction, but the pixels on the right side of the plot which correspond to the bright red regions

Fig. 3. The original test image, blurred slightly with a Guassian kernel

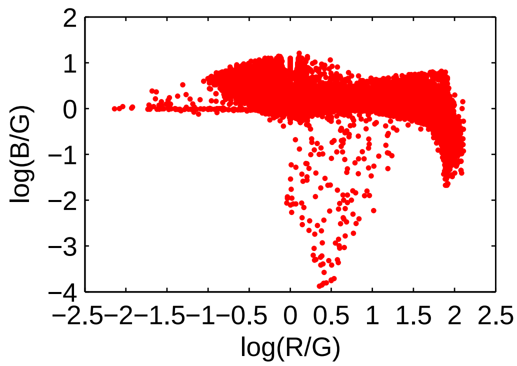

Fig. 4. Log-Chromticity plot of the original image

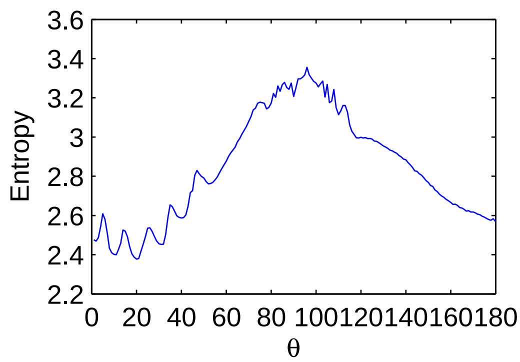

Fig. 5. Entropy of the invariant image as we rotate from 1 to 180 degrees

Fig. 6. Optimal invariant image as chosen by our algorithm

in the image seem to to be spread out vertically. As in figure 2, we see that different colors seem to be spread across very different directions, which makes the construction of an illumination invariant image impossible using the methods we've looked at.

V. Conclusions and Future Work

In this report we implemented an algorithm that has been developed and refined over the course of a number of years by a number of researchers for creating an illumination invariant image from a single image using an uncalibrated camera. The authors of the paper were able to get excellent results using pictures taken with a high-end professional camera. In the papers, the authors claim that their method works on all of the cameras that they tested, and so we attempted to test whether the method would work on a consumer-grade camera. It turns out that there are properties of lower end digital cameras that prevent the method from producing useful results. Colors and contrast are distorted in lighter and darker regions of images, and these two problems cause the algorithm to find a suboptimal result.

It would be interesting to find a different measure other than entropy of the invariant image for determining how good the image is. The log-chromaticities of the different colors in the image were found to translate linearly as the properties of the illumination changed, the problem was that the linear translation had a different direction for different colors. Somehow taking this into consideration, and possibly rotating various bands of colors in different directions could produce better results, but we did not have a chance to test this.

References

[1] G.D. Finlayson, M.S. Drew, and C. Lu. Intrinsic images by entropy minimization. *Proc. 8th European Conf. on Computer Vision, Praque*, pages 582–595, 2004.

[2] G.D. Finlayson and S.D. Hordley. Color constancy at a pixel. *Journal of the Optical Society of America A*, 18(2):253–264, 2001.

[3] G.D. Finlayson, S.D. Hordley, and M.S. Drew. Removing shadows from images. *ECCV 2002: European Conference on Computer Vision*, page 4, 2002.

[4] G. Wyszecki and W.S. Stiles. *Color science*. Wiley New York, 1982.