

Lighting Invariance through Joint Color Change Models

Erik G. Miller,* Kinh Tieu, and Eric Grimson
Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139

Abstract

In [9], we introduced a linear statistical model of *joint color changes* in images due to variation in lighting and certain non-geometric camera parameters. We did this by measuring the mappings of colors in one image of a scene to colors in another image of the same scene under different lighting conditions.

In this paper, we extend our model in several ways and examine its applicability to several important problems in machine vision. The extensions to our model include incorporating a model of image noise and a prior on the color flows used to explain a particular image difference. In addition, we increase the flexibility of our model by allowing color flow coefficients to vary according to a low order polynomial over the image. This allows us to better fit smoothly varying lighting conditions as well as curved surfaces without endowing our model with too much capacity. The problems we explore include shadow removal and detection as well as inference of scene geometry.

1. Introduction

In previous work [9], we introduced *color flows*, a linear statistical model of *joint color changes* in images due to variation in lighting. Color flows are meant to model not only common global lighting changes (which have an approximately linear effect on scene radiance), but also various non-linear camera effects such as aperture settings, transducer dynamics, and gain correction.

A color flow is a vector field in color space; each vector in the vector field starts at one color and ends at another. Since a vector field can be considered an element of a vector space, an ensemble of color flows can be studied using common statistical methods such as clustering and principal components analysis. We showed that the principal components of the ensemble of color flows (the basis fields

that represented the directions of greatest variation in the ensemble) corresponded to common modes of color variation in images due to lighting changes, such as “brightness”, “warmth”, and “contrast”. We then demonstrated how color flows can be used to create a simple model for the variability in images of a new object using only a single example of that object.

In Sections 2 and 3, we define the color flow model and how it is obtained from observations in a particular environment. Then in Section 4, we discuss image matching with color flows, incorporating an image noise model and a prior on color flows. We examine the applicability of our model to several important problems in machine vision in Section 5. The problems we explore include shadow removal and detection as well as inference of scene geometry. We also describe how we increase the flexibility of our model by allowing color flow coefficients to vary according to a low order polynomial over the image. This allows us to better fit smoothly varying lighting conditions as well as curved surfaces without endowing our model with too much capacity. We discuss the advantages and disadvantages of our model with respect to these problems and relationships to other lighting invariance models such as [2, 1, 12].

2. Color Flows

In the following, let $\mathcal{C} = \{(r, g, b)^T \in \mathbb{R}^3 : 0 \leq r \leq 255, 0 \leq g \leq 255, 0 \leq b \leq 255\}$ be the set of all possible observable image color 3-vectors. Let the vector-valued color of an image pixel p be denoted by $\mathbf{c}(p) \in \mathcal{C}$.

Suppose we are given two N -pixel RGB color images I_1 and I_2 of the same scene taken under two different sets of photic parameters θ_1 and θ_2 (the images are registered). Each pair of corresponding image pixels p_1^k and p_2^k , $1 \leq k \leq N$, in the two images represents a mapping $\mathbf{c}(p_1^k) \mapsto \mathbf{c}(p_2^k)$. That is, it tells us how a particular pixel’s color changed from image I_1 to image I_2 . This single-color mapping is conveniently represented simply by the vector difference between the two pixel colors:

$$\mathbf{d}(p_1^k, p_2^k) = \mathbf{c}(p_2^k) - \mathbf{c}(p_1^k). \quad (1)$$

By computing N of these vector differences (one for each

*Contact author email: emiller@ai.mit.edu

pair of pixels) and placing each vector difference at the point $c(p_1^k)$ in the color space \mathcal{C} , we have created a vector field that is defined at all points in \mathcal{C} for which there are colors in image I_1 .

That is, we are defining a vector field Φ' over \mathcal{C} via

$$\Phi'(c(p_1^k)) = \mathbf{d}(p_1^k, p_2^k), \quad 1 \leq k \leq N. \quad (2)$$

This can be visualized as a collection of N arrows in color space, each arrow going from a source color to a destination color based on the photic parameter change $\theta_1 \mapsto \theta_2$. We call this vector field Φ' a *partially observed color flow* (see Figure 1a). The “partially observed” indicates that the vector field is only defined at the particular color points that happen to be in image I_1 .

To obtain a *full color flow*, i.e. a vector field Φ defined at all points in \mathcal{C} , we merely interpolate between the values of the partially observed color flow as discussed in our previous paper (Figure 1b). Our interpolation scheme [9] is defined so that a color point with only a single nearby observed neighbor will inherit a flow vector that is nearly *parallel* to that neighbor’s flow vector. The idea is that if a particular color, under a photic parameter change $\theta_1 \mapsto \theta_2$, is observed to get a little bit darker and a little bit bluer, for example, then its neighbors in color space are also defined to exhibit this behavior.

We have thus outlined a procedure for using a pair of corresponding images $\mathcal{I} = (I_1, I_2)$ to generate a full color flow. We will write for brevity $\Phi = \Phi(\mathcal{I})$ to designate the flow generated from the image pair \mathcal{I} .

2.1 Data Collection

Our aim was to capture the structure in color flow space by observing real-world data in an unsupervised fashion. To do this, we gathered data as follows. A large color palette (approximately 1 square meter) was printed on standard non-glossy plotter paper using every color that could be produced by our Hewlett Packard DesignJet 650C pen plotter.

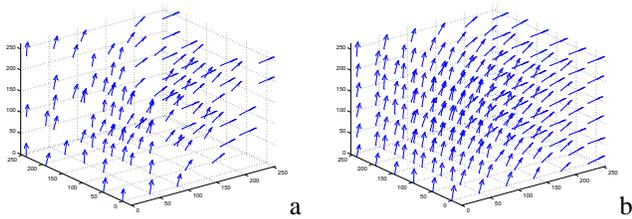


Figure 1: Color flows as vector fields in color space. **a.** A (synthetic) partially observed color flow obtained from a pair of images under different lighting conditions. An arrow appears at each point in color space where a color was observed. **b.** The interpolated completion of this color flow.

The poster was mounted on a wall in our office so that it was in the direct line of overhead lights and computer monitors, but not in the direct light from the single office window. In this series of experiments, we used the Sony DCR-VX2000 NTSC video camera. Settings were chosen to dynamically adjust for brightness so that various non-linearities would be introduced (of course, this makes the camera useful in a wider range of lighting settings). The camera was aimed at the poster so that the poster occupied about 95% of the field of view.

Images of the poster were captured using the video camera under a wide variety of lighting conditions, including various intervals during sunrise, sunset, at midday, and with various combinations of office lights and outdoor lighting (controlled by adjusting blinds). People used the office during the acquisition process as well, thus affecting the ambient lighting conditions. It is important to note that a variety of non-linear normalization mechanisms built into the camera were operating during this process.

We chose a set of 1000 image pairs $\mathcal{I}^j = (I_1^j, I_2^j)$, $1 \leq j \leq 1000$, by randomly and independently selecting individual images from the set of raw images. Each image pair was then used to estimate a full color flow.

3 The Statistics of Color Flows

One of the primary goals of this work is to statistically model color changes due to lighting variation. Intuitively, we know that the lighting changes that occur in one setting are statistically different than lighting changes that occur in other settings. For example, the most common lighting changes in many windowless offices are due simply to the switching on and off of electrical lights. On the other hand, in a rural outdoor setting, it is the rising and setting of the sun and changes in the weather that most dramatically affect the lighting. It seems reasonable to assume that the statistics of color changes in images are a function of the setting in which those images are acquired.

As discussed previously, the particular properties of the camera also affect the statistics of color flows. Contrast flows and other non-linear effects are the result of normalizations, saturation, and gain control which are all functions of the particular camera being used.

To formalize these ideas, we define an *environment* $\mathcal{E} = (p_L(\cdot), C)$ to be a pair consisting of a probability density $p_L(\cdot)$ over lighting conditions and a particular camera make and model C . The primary purpose of introducing these notions is to emphasize that if color flow statistics are gathered in one environment and applied in another, the results may be unreliable. Since our data was gathered in a (windowed) office environment with a particular camera, the model we produce represents that particular combination. In other

words, we would not expect to be able to model outdoor lighting changes from the office data we gathered.

3.1 Statistics of the Office Environment

Light primarily has an effect when it impinges upon some surface. Ideally, the environment defined above should include the angle of the surface being evaluated relative to the light sources since this variable has a major impact on the irradiance of the surface. However, here we make the simplifying assumption that the distribution over lighting conditions is spatially isotropic. In other words, while at any given time the light may be coming from a particular direction, we model it as equally likely to come from any direction. This isotropic distribution over lighting conditions then induces a distribution over lighting impinging on a surface. If the distribution of lighting is isotropic, then the conditional distribution of incident light on the surface is invariant to the surface orientation. By sampling pairs of these lighting conditions at different times, we obtain various color flows.¹

In this work, we assume that the distribution over color flows in the office environment is a zero-mean Gaussian.² We perform a principal components analysis to obtain the dominant modes of variation of the color flows.³ We call the flows of greatest variation that emerge from this analysis the “color eigenflows”.

Since color flows are models of image *change* rather than images per se, they are perhaps best visualized by being applied to a particular image. By applying flows to an image of a flat object photographed in the environment to which the flow model belongs, we obtain images representing the variability of that object in the environment. We show the result of applying the office-based eigenflows to the picture of a painting in Figure 2. Each column represents one mode of lighting variation for the original image (at the middle of each column) as predicted by the “office” color flow model. Intuitively, the first column represents something similar to brightness. The second column is clearly a shift from a blueish tint to a yellowish tint, and may represent changes

¹Since we are ultimately interested in image differences created by *different* lighting conditions, we are interested in the distribution of color flows induced by comparing images from non-adjacent instants in time. If we were to build a distribution of flows based on adjacent video frames, the majority of the probability mass of color flows would be at the “null” flow.

²Since colors at the boundary of color space (e.g. black, white, fully saturated red, etc.) will always flow away from the boundary of the color cube or not at all, the mean color flow cannot be expected to actually be zero for these colors. However, a non-zero mean flow would imply, for a Gaussian distribution of flows, that the maximum likelihood flow was non-zero. This in turn would imply that two identical images were best explained by some non-zero flow. In order to preserve the “null” flow as the maximum likelihood flow, we make the zero mean assumption.

³We are also currently investigating a factor analysis model of color flows.

due to early morning or late afternoon lighting conditions. The third column can be seen to be a non-linear effect that increases contrast in one direction, and decreases it in the other.

Here we emphasize perhaps the most important feature of our model. Since it is a model of image change rather than of images, it can be applied to any image. The resulting image set is an induced model of the variability of the input image under the environment’s changing illumination. A major limitation is that this applies only to images of flat surfaces. However, we will investigate below how color flows can be used in settings with multiple and curved surfaces.

4 Image matching with color flows

The central aim of this workshop is to recognize objects across variations in lighting. One approach to solving this problem is to evaluate the likelihood that two images I_1 and I_2 are of the same object under some model of lighting variability. Here, we assume that two images of the same scene are affected by lighting changes (modeled by the color flow distribution) and by additive independent Gaussian noise (in all three color channels). We then evaluate the likelihood that the difference between two images was caused by a particular lighting change. A threshold on this likelihood can be used to decide whether images “match” or not, i.e. whether they are images of the same scene. To evaluate whether a distribution of color flows can explain the difference between two images, we must first know how to apply a color flow to an image. This is similar to the formulation of handwritten digit matching in [8] where color transformations take the place of spatial transformations.

4.1 Applying a flow to an image

A flow is applied to an image in the following way. Let $\mathbf{c}(p)$ be the color of a pixel p in the source image, and let Φ be a color flow. For each pixel in the new image, its color \mathbf{c}' can be computed as

$$\mathbf{c}'(p) = \mathbf{c}(p) + \alpha\Phi(\hat{\mathbf{c}}(p)), \quad (3)$$

where α is a scalar multiplier that represents the “quantity of flow”. $\hat{\mathbf{c}}(p)$ is interpreted to be the color vector closest to $\mathbf{c}(p)$ (in color space) at which Φ has been computed. If the $\mathbf{c}'(p)$ has components greater than the allowed range of 0–255, then these components must be truncated.

Note that a distribution over flows paired with a new image induces a (usually isomorphic)⁴ distribution over difference images and a corresponding distribution over “new”

⁴As long as the difference image basis vectors are linearly independent, the set of difference images will be in one-to-one correspondence with the color flows.

images obtained from adding the difference images to the original image.

4.2 Estimating image differences under noise with color flows

Central to this paper will be the maximum a posteriori (MAP) estimation of the “color flow” between two images according to our model. That is, given two images I_1 and I_2 , we wish to explain the image difference $D = I_2 - I_1$ as some combination of changes due to photic parameters (modeled by eigenflows) and additive pixel-independent Gaussian image noise.

The difference images $\{D_i\}$ that can be created by applying each eigenflow Ψ_i to an original image I are obtained simply by changing each color in an image by the amount specified in one standard deviation worth of a particular eigenflow. We notate this as

$$D_i = I(\Psi_i).$$

These images will form a basis for the linear space of difference images that can be created from the eigenflows. We then seek the MAP solution to the equation

$$D = \sum_{i=1}^E \gamma_i D_i + n,$$

where E is the number of eigenflows used, and n is pixelwise and channelwise independent Gaussian noise in the image with diagonal covariance Λ_n . Also, the image difference D should be viewed as an “unwrapped” vector here of size $3N$, where N is the number of pixels in the image. In matrix form, we can write this as

$$D = \mathbf{D}\gamma + n,$$

where \mathbf{D} is a matrix whose columns are the basis difference images and γ is the vector of coefficients multiplying the basis vectors.

To solve this problem, we need Λ_n , an estimate of the noise variance, and Λ_γ , the covariance of the image difference coefficients. Since, by definition, the difference images already incorporate a multiplier proportional to the standard deviation of the color flow in that particular direction, Λ_γ is just the identity matrix.

To obtain an estimate of image noise, we looked at the mean of the magnitude of the difference between a pixel’s red, green, and blue components between successive images. This turned out to be about 4.4 on a scale of 0-255 per color channel. This distribution also turned out to be substantially dependent upon the color in the initial image, but we assumed isotropic noise in this work, i.e. we set Λ_N to $4.4 \times \mathbf{I}$.



Figure 2: Effects of the first 3 eigenflows. Each image in the center row is a copy of the original image. Each column represents ± 4 standard deviations of a particular eigenflow.

With Λ_γ and Λ_N , we can write down a ‘‘ridge regression’’ style MAP estimate of γ which is

$$\begin{aligned}\hat{\gamma}_{MAP} &= (\mathbf{D}^T \Lambda_N^{-1} \mathbf{D} + \Lambda_\gamma^{-1})^{-1} \mathbf{D} \Lambda_N^{-1} D \quad (4) \\ &= (\mathbf{D}^T \Lambda_N^{-1} \mathbf{D} + \mathbf{I})^{-1} \mathbf{D} \Lambda_N^{-1} D. \quad (5)\end{aligned}$$

Using this estimate of γ , then the color flow estimate of the difference image becomes

$$\hat{D} = \sum_{i=1}^N \hat{\gamma}_i D_i$$

and the residual ‘‘unexplained’’ part of the image difference is

$$\begin{aligned}\sum_{x,y} \|D(x,y) - \hat{D}(x,y)\| & \quad (6) \\ = \sum_{x,y} \|\mathcal{I}_2(x,y) - \mathcal{I}_1(x,y) - \hat{D}(x,y)\|. & \quad (7)\end{aligned}$$

Setting a threshold on this value can then be used to accept or reject the hypothesis that the two images match. In our previous paper, we give preliminary comparisons of residual error obtained by using standard models of color change and the color flows method. Next we discuss the advantages and disadvantages of the color flow model relative to other lighting invariance methods in addressing various problems related to lighting invariance.

5 Problems related to lighting invariance

5.1 Generative models for images

One of the most generic tasks in the domain of lighting invariance is to generate arbitrary new views of an object given some model of the object, while producing as few ‘‘unrealistic’’ views as possible. There are many models of lighting invariance which can be used to produce novel images of an object under new lighting conditions of which ([2, 1, 12, 3, 7]) are just a few. They differ in several respects.

5.1.1 What is being modeled

First, approaches such as Belhumeur et al.’s [2] and Basri et al.’s [1] differ from our approach in the effects that they are modelling. In both of these, the goal is to model distant light sources on Lambertian objects with constant lighting throughout the scene (with the exception of self-shadows). There is no attempt to model the distribution of lighting color, or non-linear camera effects. There is also no attempt to deal with the effects of nearby light sources, which will

have varying angles of incidence even upon flat surfaces. While our model also makes the Lambertian assumption, it differs in that it puts a probability density on lighting changes for flat surfaces. It is also easily adaptable to explain the effects of nearby light sources, as we will discuss in the section on shadows below. However, it does not directly address self-cast shadows.

5.1.2 Creating the object model

Before lighting invariance models can be used to predict new images or for other purposes, the model itself must be created. Current methods vary widely in the complexity of the task of creating the model. Predicting images of an object according to the procedure used in [1] requires that one have a full 3-D model of the object at hand, and hence either a special apparatus for each object, or a number of sample images coupled with an algorithm for constructing such a 3D model.

Many object specific models have been examined, such as the color eigenface model described in [10]. However, these models must be trained separately for each class, with enough images to develop a good estimate of a possibly high dimensional space. They have no generalization ability across classes.

Under certain assumptions, as few as 3 images per object can be used to generate all of the grayscale variations of an object as described in [2]. We mention again that this model assumes light sources at infinity and perfect Lambertian reflectances.

One of the appealing features of our model is that, while it is trained on many hundreds or thousands of images, it can then be applied to new objects using only a single example of that object. While our model works best for flat objects, it gives reasonable results for non-flat objects as well, and effectively models ambient light changes, i.e. lighting changes that have an approximately equivalent effect on all parts of the object.

By smoothly varying the eigenflow coefficients across an image, we can approximate nearby lighting effects as well. It is easy to train such a model on a particular type of change too, such as shadow changes.

5.1.3 Probabilistic image generation

Another issue regarding image generation is being able to associate a likelihood or probability with the images generated. By applying a 3x3 linear matrix operator to each color in an image, we can generate a wide variety of plausible images from a single exemplar. However, with such a generative model there is no notion of the likelihood of such images. In particular, choosing the matrix randomly according to a uniform distribution over the components of

the matrix, the model will produce mostly images with implausible color distributions.

The same is true for other generative image processes. Of course, these other methods could be extended to include probability densities over the generated images.

The color flow model explicitly assigns a probability density to each color change that is particular to the environment from which the flows were collected. This should make it more effective in applications where the likelihood can be used to help discriminate among explanations.

5.2 Shadows

Shadows pose a number of interesting problems in computer vision. Shadows confuse tracking algorithms [11], backgrounding schemes and object recognition algorithms. For example, shadows can have a dramatic effect on the magnitude of difference images, despite the fact that no “new objects” have entered a scene. Shadows can also move across an image and appear as moving objects. Many of these problems could be eliminated if we could recognize that a particular region of an image is equivalent to a previously seen version of the scene, but under a different lighting. For example, suppose that the lighting impinging upon a flat surface has changed due to a nearby lamp being turned on. The changing angle of incidence will make it difficult to model the image transformation as a single mapping of color space from one image to the other.

In Figure 3a, we show a simple background image. In Figure 3b, a person and his shadow have appeared in the image. We consider the problem of distinguishing between the person (a new object) and the shadow (a lighting change). We did the following experiment. With simple image differencing, we segmented the image into two approximately connected regions that did not match the previous background (Figure 3c). For each component, we then flowed (chose eigenflow coefficients) the region from image a to image b according to Eq. 4. Figure 3d shows the full image based on the shadow flow.

To distinguish between shadows and non-shadows, we want the average residual error for non-shadows to be high while the average residual error for shadows to be low. Since these are real images, however, a constant color flow across an entire region may not model the image change well.

However, we can easily extend our basic model to allow linearly or quadratically (or other low order polynomially) varying fields of eigenflow coefficients. That is, we can find the best least squares fit of the difference image allowing our γ estimates to vary linearly or quadratically over the image. We implemented this technique by computing flows $\gamma_{x,y}$ between corresponding image patches (indexed by x

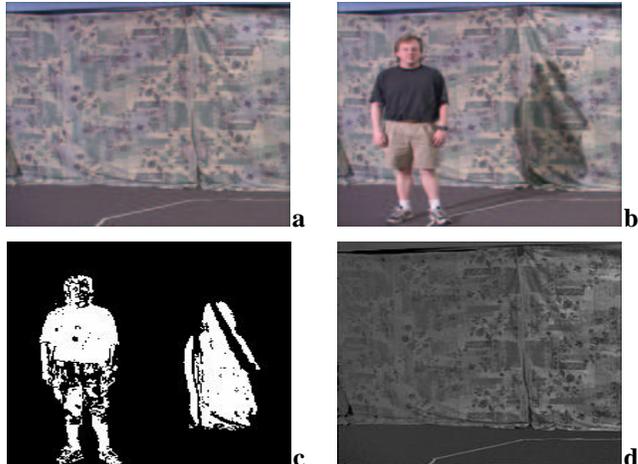


Figure 3: **a** A background image. **b** A new object and shadow have appeared. **c** For each of the two regions, a “flow” was done between the original image and the new image based on the pixels in each region. **d** The color flow of the original image using the eigenflow coefficients recovered from the shadow region. The color flow using the coefficients from the non-shadow region are unable to give a reasonable reconstruction of the new image.

	constant	linear	quadratic
shadow	36.5	12.5	12.0
non-shadow	110.6	64.8	59.8

Table 1: Error residuals for the shadow and non-shadow regions after color flows. Constant, linearly varying, and quadratically varying flows were used.

and y), and then minimizing the following form:

$$\arg \min_M \sum_{x,y} (\gamma_{x,y} - M c_{x,y})^T \Sigma_{x,y}^{-1} (\gamma_{x,y} - M c_{x,y}). \quad (8)$$

Here, each $c_{x,y}$ is a vector polynomial of the form $[x \ y \ 1]^T$ for the linear case and $[x^2 \ xy \ y^2 \ x \ y \ 1]^T$ for the quadratic case. M is an $E \times 3$ matrix in the linear case and an $E \times 6$ matrix in the quadratic case. It defines each of the E planes or quadrics respectively. The $\Sigma_{x,y}^{-1}$'s are the error covariances in the estimate of the $\gamma_{x,y}$'s for each patch.

Allowing the γ 's to vary over the image obviously greatly increases the capacity of a matcher, but by limiting this variation to linear or quadratic variation, the capacity is still not able to qualitatively match “non-matching” images. Note that this smooth variation in eigenflow coefficients can model either a nearby light source *or* a smoothly curving surface, since either of these conditions will result in a smoothly varying lighting change.

We consider three versions of the experiment. In the first,

we assign a single vector of flow coefficients to all of the pixels in the region. In the second experiment, we allowed the γ values to vary linearly across the image. This clearly should lead to a reduction in the error of both regions' residuals. In the final experiment, we fitted quadratically varying γ values to estimate the image difference. The results of these experiments appear in Table 1.

In each case, the residual error for the shadow region is much lower than for the non-shadow region. Of course, we have not specified where to select the threshold so that this procedure works in general. Furthermore, there are other methods available, such as normalized correlation and methods such as [4, 5], which could also distinguish between these two regions. However, this demonstrates another potential application of our model. We believe because it can handle non-linear camera effects and can be adjusted across the image that it can successfully model a great deal of the variability in true shadows, whereas it still does not have so much capacity as to match images which are not in fact of the same scene. Clearly, however, it still has limitations, such as when a shadow is so dark that it cannot be distinguished from a black object.

5.3 Inferring scene geometry through lighting change

Another interesting application of lighting invariance models is the detection of discontinuities in surfaces and surface normals. Such discontinuities can be used to perform image segmentation and other image processing tasks. If images were truly linear in the impinging lighting, then given two images of a scene under different lights, we could detect discontinuities in surfaces by simply using the following technique. First, calculate the "ratio image"

$$r(x, y) = b_1(x, y)/b_2(x, y),$$

where b_1 and b_2 are just the pixel brightnesses in image 1 and image 2. If the difference between the surface normals of two adjacent patches is small, then the change in the angle of incidence upon those patches (for a particular global lighting change) will also be approximately equal. This is a simple consequence of the assumed linearity of imaging, as described in [6].

However, this approach depends critically on a linear lighting model. It is easy to see however, that there are non-linear effects in image acquisition which cause problems for such methods. An example of this non-linearity is demonstrated in Figure 4. The top two images in Figure 4 are two photographs of a box covered with multicolored paper. The photos show the top and one side of the box. The lower left image is the ratio of these two images. Since this is the ratio of images of two distinct smooth surfaces, it should have only two regions of smoothly varying pixels. However, it is

clear that the ratios seen are variable, even within the individual regions. Examining the original images, it is clear that the ratio image is a function not only of surface normal, but also of albedo. The darker regions in the original images show different values in the ratio image than the lighter regions. The fact that the ratio image is still a function of the albedo is direct evidence of a non-linearity in the imaging process.

If we want to detect discontinuities in surface normals, we need to be able to non-linearly map colors between corresponding points in two images. To measure the success of such a mapping, we need a measure of the discontinuity of a ratio image. We chose the integral of the edge magnitudes of the ratio image as such a measure, similar to what was done in [6].

We performed the following experiment. We took two images, I_1 and I_2 of a smooth surface with highly variable albedo. Using a linear model as described in [9], we found the nearest (least squares error) transformation \hat{I}_{lin} of I_1 to I_2 . We then did the same with the eigenflows to create \hat{I}_{eig} . The integral of the edge images were then computed for \hat{I}_{lin} and \hat{I}_{eig} . We found that this integral was about 7% smaller for the eigenflowed ratio image. While this difference is not great, we believe it indicates that this non-linear technique could improve the usefulness of ratio image methods.

6 Conclusions and future work

While this work is preliminary in nature, we believe it suggests that color flow techniques may help us address problems created by nonlinearities in image acquisition. Clearly we need to perform more experiments in a variety of settings. We would like to avoid making certain assumptions (such as the isotropy of the distribution over lighting changes) by collecting more data for our color flow model.

Acknowledgements

This work was supported in part by ONR grant #'s N00014-00-1-0089 and N00014-00-1-0907. We thank Chris Stauffer and John Fisher for helpful discussions.

References

- [1] R. Basri and D. Jacobs. Lambertian reflectance and linear subspaces. In *International Conference on Computer Vision*, pages 383–390, 2001.
- [2] P. N. Bellhumeur and D. Kriegman. What is the set of images of an object under all possible illumination conditions? *International Journal of Computer Vision*, 28(3):1–16, 1998.
- [3] G. Buchsbaum. A spatial processor model for object color perception. *Journal of the Franklin Institute*, 310, 1980.

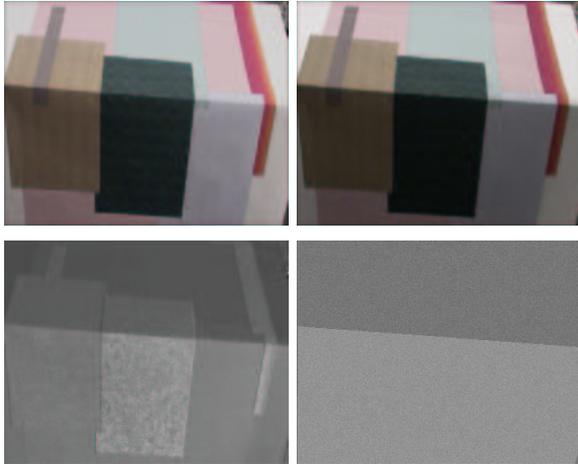


Figure 4: Evidence of non-linear color changes. The ratio of the top two images is shown in the lower left corner. The ideal ratio image, corresponding to a linear lighting model, is shown in the lower right.

- [4] G. Gordon, T. Darrell, M. Harville, and J. Woodfill. Background estimation and removal based on range and color. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1999.
- [5] T. Horprasert, D. Harwood, and L. S. Davis. A robust background subtraction and shadow detection. In *Proceedings of the Asian Conference on Computer Vision, 2000*, 2000.
- [6] D. Jacobs, P. Belhumeur, and R. Basri. Comparing images under variable illumination. In *IEEE Computer Vision and Pattern Recognition*, pages 610–617, 1998.
- [7] J. J. McCann, J. A. Hall, and E. H. Land. Color mondrian experiments: The study of average spectral distributions. *Journal of the Optical Society of America*, A(67), 1977.
- [8] E. Miller, N. Matsakis, and P. Viola. Learning from one example through shared densities on transforms. In *IEEE Computer Vision and Pattern Recognition*, 2000.
- [9] E. Miller and K. Tieu. Color eigenflows: Statistical modeling of joint color changes. In *International Conference on Computer Vision*, volume 1, pages 607–614, 2001.
- [10] M. Soriano, E. Marszalec, and M. Pietikainen. Color correction of face images under different illuminants by rgb eigenfaces. In *Proceedings of the Second International Conference on Audio- and Video-Based Biometric Person Authentication*, pages 148–153, 1999.
- [11] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers. Wallflower: Principles and practice of background maintenance. In *IEEE Computer Vision and Pattern Recognition*, pages 255–261, 1999.
- [12] Yair Weiss. Deriving intrinsic images from image sequences. In *International Conference on Computer Vision*, 2001.