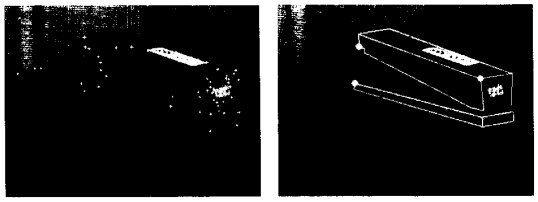
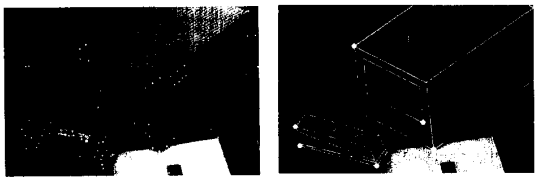
cardinality of the sets of image and model points necessary in an indexing system, while retaining the indexing speedup. The disadvantage to this system is that not all correct matches between image groups and model groups are indexed. Since a far higher fraction of correct matches are indexed than of incorrect matches, probabilistic indexing can be used to help discriminate between correct and incorrect hypotheses. These techniques have been applied to the alignment method and found to speed up the recognition process by a considerable amount.



(a) (b)

Fig. 3. Recognition of a stapler: (a) The corners found in the image; (b) A correctly indexed group and the corresponding model pose.



(a) (b)

Fig. 4. Recognition in a more complicated scene: (a) The corners found in the image; (b) Correctly indexed groups and the corresponding model poses.

AKNOWLEDGMENTS

This work has been supported by a National Science Foundation Graduate Fellowship to the author, NSF PYI Grant IRI-8957274 to Jitendra Malik, and NSF Materials Handling Grant IRI-9114446.

REFERENCES

[1] J. Ben-Arie, "The probabilistic peaking effect of viewed angles and distances with application to 3D object recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 12, pp. 760-744, Aug. 1990.

[2] T.O. Binford, T.S. Levitt, and W.B. Mann, "Bayesian inference in model-based machine vision," *Uncertainty in Artificial Intelligence 3*, L.N. Kanal, T.S. Levitt, and J.F. Lemmer, eds., pp. 73-95. North Holland: Elsevier Science Publishers B.V., 1989.

[3] J.B. Burns, R.S. Weiss, and E.M. Riseman, "View variation of point-set and line segment features," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 15, pp. 51-68, Jan. 1993.

[4] D.T. Clemens and D.W. Jabobs, "Space and time bounds on indexing 3D models from 2D images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 13, pp. 1,007-1,017, Oct. 1991.

[5] M. A. Fischler and R.C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Comm. ACM*, vol. 24, pp. 381-396, June 1981.

[6] W. Förstner, "Image matching," chapt. 16 of *Computer and Robot Vision*, vol. II, by R. Haralick and L. Shapiro, Addison-Wesley, 1993.

[7] W. Förstner and E. Gülch, "A fast operator for detection and precise locations of distinct points, corners, and centers of circular features," *Proc. Intercommision Conf. Fast Processing of Photogrammetic Data*, pp. 281-305, 1987.

[8] D. Forsyth, J.L. Mundy, A. Zisserman, C. Coelho, A. Heller, and C. Rothwell, "Invariant descriptors for 3D object recognition and pose," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 13, pp. 971-991, Oct. 1991.

[9] W.E.L. Grimson and D.P. Huttenlocher, "On the sensitivity of geometric hashing," *Proc. Int'l Conf. Computer Vision*, pp. 334-338, 1990.

[10] W.E.L. Grimson, D.P. Huttenlocher, and D.W. Jacobs, "A study of affine matching with bounded sensor error," *Proc. European Conf. Computer Vision*, pp. 291-306, 1992.

[11] D.P. Huttenlocher and S. Ullman, "Recognizing solid objects by alignment with an image," *Int'l J. Computer Vision*, vol. 5, no. 2, pp. 195-212, 1990.

[12] Y. Lamdan, J.T. Schwartz, and H.J. Wolfson, "Object recogniton by affine invariant matching," *Proc IEEE Conf. Computer Vision and Pattern Recognition*, pp. 335-344, 1988.

[13] D.G. Lowe, "Three-dimensional object recognition from single 2D images," *Artificial Intelligence*, vol. 31, pp. 355-395, 1987.

[14] C.F. Olson, "Fast alignment using probabilistic indexing," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 387-392, 1993.

[15] C.F. Olson, "Probabilistic indexing: Recognizing 3D objects from 2D images using the probabilistic peaking effect," Tech. Report UCB//CSD-93-733, Univ. of California at Berkeley, May 1993.

[16] C.F. Olson, "Probabilistic indexing: A new method of indexing 3D model data from 2D image data," *Proc. Second CAD-Based Vision Workshop*, pp. 2-8, 1994.

[17] I. Weiss, "Noise resistant projective and affine invariants," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 115-121, 1992.

Color Constant Color Indexing

Brian V. Funt and Graham D. Finlayson

***Abstract*—Objects can be recognized on the basis of their color alone by *color indexing*, a technique developed by Swain and Ballard [15] which involves matching color-space histograms. Color indexing fails, however, when the incident illumination varies either spatially or spectrally. Although this limitation might be overcome by preprocessing with a color constancy algorithm, we instead propose histogramming color ratios. Since the ratios of color RGB triples from neighboring locations are relatively insensitive to changes in the incident illumination, this circumvents the need for color constancy preprocessing. Results of tests with the new color-constant-color-indexing algorithm on synthetic and real images show that it works very well even when the illumination varies spatially in its intensity and color.**

***Index Terms*—Color indexing, color constancy, retinex, object recognition.**

I. INTRODUCTION

Swain and Ballard [15] developed a very clever, simple scheme that identifies objects entirely on the basis of color. Their method, which they call *color indexing*, radically departs from traditional object recognition strategies based on geometric properties. Color indexing turns out to be remarkably robust in that variations such as a change in orientation, a shift in viewing position, a change in the scene background, partial occlusion, or even a radical change in shape (e.g., a shirt tossed onto a chair two different ways), degrade recognition only slightly.

On the other hand, Swain's algorithm is very sensitive to the lighting. Simple changes in the illumination's intensity—let alone its color—radically alter the algorithm's results. Clearly, one solution to

Manuscript received Feb. 19, 1992; revised Aug. 27, 1993.

The authors are with the School of Computing Science, Simon Fraser University, Vancouver, B.C., Canada V5A 1S6; (604) 291-3126; e-mail funt@cs.sfu.ca.

IEEECS Log Number P95047.

this problem is to preprocess the images with some sort of color constancy algorithm [6], [11], [8], [7] in order to remove the effects of varying illumination conditions and, in effect, normalize the images to a standard illuminant. Swain, in fact, suggests using Novak et al.'s [14] supervised color constancy approach in which a few patches of known spectral reflectance are placed in the scene.

While preprocessing for color constancy could well work, it feels unsatisfying in that color constancy appears to be a more difficult problem than color-based object identification. Preprocessing for color constancy destroys the overall elegance inherent in Swain's method. Simple color adjustment schemes, such as that of placing a known white calibration patch in the scene, will not work well in general because illumination usually varies spatially both in intensity and spectral composition due to shading, multiple sources, and interreflections.

In this paper, we extend Swain's color indexing method to be illumination independent, not by preprocessing for color constancy, but rather by indexing on an illumination-invariant set of color descriptors. Swain indexes directly on a color triple, we instead index on the derivative (Laplacian or first directional derivatives) of the *logarithm* of the colors, which in effect is the ratio of neighboring colors. Since the illumination remains essentially constant locally, ratioing neighboring colors removes the illumination component.

Anyone familiar with Land's work will recognize that the roots of this method lie in retinex theory [11]. Our goal here, though, is not color constancy and we do not propose applying full retinex theory. Extending Swain's algorithm requires only illumination-independent descriptors on which to index, it does not require the actual, correct colors. The descriptors we propose, which are in essence the ratios of the correct colors, can be generated much more easily than the correct colors themselves.

We have tested our algorithm on databases of real and synthetic images and compare the results with those of Swain's algorithm. In general, our new algorithm does slightly worse than Swain's under fixed illumination, but substantially better than Swain's (his fails completely) under illumination that varies both spectrally and spatially.

II. Summary of Swain's Color Indexing

Intuitively, Swain's color-indexing algorithm identifies an object by comparing its colors to the colors of each object in a database. Crucial to this working, however, is that the total area covered by each color also be taken into account. The areas are computed and compared by histogramming the images and intersecting the histograms. Color histograms and histogram intersection form the backbone of Swain's method.

A color histogram is three-dimensional and simply represents the count of the number of pixels in an image having a particular RGB value. Before histogramming, Swain converts from camera RGB triples to an opponent-theory-based [2] color space with axes black-white, red-green and blue-yellow. The space is coarsely partitioned along the black-white axis so that histograms which, at 8 bits per channel, potentially could have 256^3 bins use only 2,048 bins. Computing color histograms requires time proportional to the number of image pixels, but with special, though not elaborate, hardware Swain reports computation times of 0.04 second.

Color histograms of images of every model object, segmented from the image background, are precomputed and stored in a database. Presented with an image of an unknown object, the color indexing algorithm computes its color histogram and intersects it with every one of the stored histograms in order to find the one that matches best. The intersection of histograms H_a and H_b is defined as:

$$\sum_{i,j,k} min\{H_a(i,j,k), H_b(i,j,k)\} \tag{1}$$

Intersection (or match) values are normalized by the number of pixels in the model histogram, thus matches are between 0 and 1. Histogram intersection only requires time proportional to the number of histogram bins and so is very fast. More sophisticated correlation measures could be used, but Swain's results show that they may not be necessary. Swain reports that the algorithm identifies the correct model most of the time.

III. Illumination Independence

We will assume that the ratio of sensor responses across a color boundary remains unaffected by changes in the incident illumination. This assumption is fundamental to retinex theory and directly implied by von Kries adaptation [16]. The assumption entails a model in which the response of the kth sensor class is formed as the product of scalar coefficients representing illumination and surface albedo relative to that class. For any location x on the surface with incident illumination e_k^x. and albedo r_k^x, we have the sensor response ρ_k^x given by

$$\rho_k^x = e_k^x r_k^x \tag{2}$$

This *coefficient model* of sensor response does not hold in general—in fact it is surprising that it holds at all—but it has a long history and for our purposes it is very useful. The main problem with it is that it depends upon the assumption that for surface reflectances $S_1(\lambda)$ and $S_2(\lambda)$, illumination $E(\lambda)$ and sensor response $R(\lambda)$ the following approximation holds:

$$\frac{\int S_1(\lambda)R(\lambda)d\lambda}{\int S_2(\lambda)R(\lambda)d\lambda} \approx \frac{\int S_1(\lambda)E(\lambda)R(\lambda)d\lambda}{\int S_2(\lambda)E(\lambda)R(\lambda)d\lambda} \tag{3}$$

This is clearly not true in general; however, one way to ensure that the coefficient model holds is to use sensors of narrowband sensitivity. Under the coefficient model, the ratio of sensor responses from two locations under the same illumination yields the ratio of surface albedos:

$$\rho_k^1/\rho_k^2 = \left(r_k^1 e_k\right)/\left(r_k^2 e_k\right) = r_k^1/r_k^2 \tag{4}$$

This ratio has little dependence on illumination other than that it be the same at both locations, which it essentially will, for proximate locations under slowly varying illumination.

Taking logarithms of both sides of (4) turns the ratios into differences

$$\ln\left(\rho_k^1\right) - \ln\left(\rho_k^2\right) = \ln\left(r_k^1\right) - \ln\left(r_k^2\right) \tag{5}$$

When these differences are taken between neighboring pixels in a particular direction, they correspond to finite-difference differentiation of the logarithm of the image. Hence, under the coefficient model of sensor response, the derivative of the logarithm of the image remains independent of the color and intensity of the incident illumination.

As derivative operators, we have experimented with both the Laplacian and a set of four directional first derivatives corresponding to the directions to a pixel's four immediate neighbors. Another natural choice would be the magnitude of the gradient; however, the Laplacian (or Laplacian of the Gaussian so as to include some smoothing) is simpler to compute, has a theoretical relationship to the center-surround cells of the human visual system [12], and is in keeping with Horn's [9] lightness algorithm.

Applying the Laplacian to the logarithm of the three channels yields a new 3-tuple for every pixel and it is these that are then histo-

grammed. In the case of the set of four directional derivatives, all four derivatives generate 3-tuples—four 3-tuples per pixel—that are counted and included as part of the histogram.

Since we found that the choice between the Laplacian and the set of four directional derivatives did not matter much, we will subsequently use "derivative" unqualified to mean either method. Intuitively, in histogramming these derivative 3-tuples, we are replacing Swain's measure of color area with a measure of color edge length. In all three channels, the derivative will be nearly zero except near the color edges where two regions of different color meet. All pixels along that color edge will generate roughly the same 3-tuple and will accumulate to the same histogram bin. The total accumulation in a particular bin, therefore, represents a measure of the length of the boundary between two specific colors. Note that there is no need to identify the color boundaries explicitly, since it is the derivative taken at all images points which is histogrammed.

Even though the RGB triple representing the color of the light reflected from the two regions changes as the incident illumination changes, the ratio of the RGB triples from the two regions does not change significantly. Since the derivatives of the logarithms of the three RGB channels reflect the ratios of the RGBs from the neighboring regions, the derivatives continue to map to the same histogram bins even as the illumination is changed. Obviously, the illumination does not affect the length of the boundary between regions. The histogram of the derivative tuples, therefore, will be invariant to the illumination.

Those derivative tuples which are near zero indicate a little bit about the shape of the object. The image of a uniform color, planar surface under spatially-uniform illumination, for example, would have, in theory, a precisely zero derivative. Wrinkling the same surface, on the other hand, would lead to a wide variety of derivative values near zero, but few at zero. Because there is no color change, only a shading change, the derivatives in all three channels will be identical (because in the logarithm of intensity, the surface albedo introduces a constant which differentiation eliminates).

IV. The Algorithm

First a database of histograms must be created. The *ratio histogram* of each image is computed as follows:

1) Logarithm step

$$i_k(x, y) \Leftarrow ln(\rho_k(x, y)) \quad k = 1 \ldots 3$$

1) Differentiation convolution step (either 2a, 2b, or 2c)

2a) Laplacian

$$d_k(x, y) \Leftarrow \nabla^2 i_k(x, y) \quad k = 1 \ldots 3$$

2b) Laplacian of Gaussian

$$d_k(x, y) \Leftarrow \nabla^2 G * i_k(x, y) \quad k = 1 \ldots 3$$

2c) Four directional derivatives

$$d_{m,k}(x, y) \Leftarrow \nabla_m i_k(x, y) \quad k = 1 \ldots 3, m = 1 \ldots 4$$

where m indexes direction.

3) Histogram step (either 3a) or 3b)

3a) Laplacian or Laplacian of Gaussian

$$H(i, j, k) \Leftarrow \sum_{x,y} z = \begin{cases} & d_1(x, y) & i \\ 1 \ \textit{if} & d_2(x, y) = & j \\ & d_3(x, y) & k \\ 0 \ \textit{otherwise} & & \end{cases}$$

3b) 4 directional derivatives

$$H(i, j, k) \Leftarrow \sum_{m=1}^{4} \sum_{x,y} z = \begin{cases} & d_{m,1}(x, y) & i \\ 1 \ \textit{if} & d_{m,2}(x, y) = & j \\ & d_{m,3}(x, y) & k \\ 0 \ \textit{otherwise} & & \end{cases}$$

Once a database of ratio (or derivative) histograms has been compiled for a set of known objects, it can be used to recognize an instance of one of these objects in a new image. The ratio histogram of the new image is calculated and then intersected with each of the model histograms in the database. Histogram intersection proceeds according to (1), which is exactly as it was for Swain.

Steps 1) and 2) represent the only additional computation required to obtain illumination independence. The logarithm in Step 1) can be done by table lookup in hardware and the derivative operators of step 2) can be implemented as convolution. For this application, the derivative operators require only a relatively small support since the intention is only to compare colors within small regions so as not to violate the assumption of constant illumination.

A. Implementation Details

An implicit premise in Swain's color histogram method is that colors should appear with equal likelihood. A uniform distribution of colors implies, however, a nonuniform distribution of ratios so for ratio histograms the bin sizes need to be adjusted accordingly. For example, suppose colors, in a single sensor class, are integers in the interval [1, 3]. If all colors are equally likely the following ratios will occur with equal probability:

$$\frac{1}{1}, \frac{1}{2}, \frac{1}{3}, \frac{2}{1}, \frac{2}{2}, \frac{2}{3}, \frac{3}{1}, \frac{3}{2}, \frac{3}{3}.$$

Clearly, ratios close to 1 are more likely than ratios close to 3. This simple illustration implies that the ratio histogram should sample ratio space nonuniformly.

Assuming that colors are uniformly distributed, we can solve for the probability distribution of ratios for a given sensor channel. Let us denote the maximum ratio that can occur as R^{max}, so we can assume that sensor catches fall in the closed interval $[1, R^{max}]$ (this could be forced by appropriate scaling). For two intensity values A and B chosen uniformly and randomly from this interval, what is the probability, $\Pr\left(\max\left(\frac{A}{B}, \frac{B}{A}\right) < R\right)$, that the ratio of $\frac{A}{B}$ (and its reciprocal) is less than R? Rewriting as $\Pr(A < BR) - \Pr\left(A < \frac{B}{R}\right)$, by standard techniques we can solve for the cumulative probability distribution of the ratios:

$$\Pr(\max(\frac{A}{B}, \frac{B}{A}) < R) = \frac{[(R^{\max})^2 + 1 - R - \frac{(R^{\max})^2}{R}]}{(R^{\max} - 1)^2} \tag{6}$$

Experimentally R^{max} was found to be approximately 4.5. Substituting 4.5 for R^{max} in (6) gives the distribution graphed in Fig. 1. It is readily apparent that ratios close to one appear with much greater frequency than ratios close to 4.5.

Given the cumulative probability distribution for ratios, we can calculate the optimal distribution of histogram bins based on the opt-

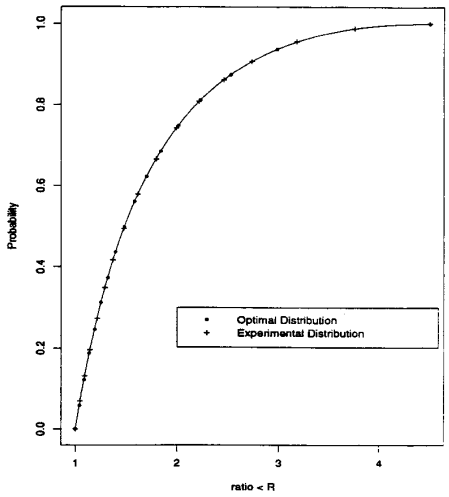


Fig. 1. The cumulative probability distribution for ratios when all colors are equally likely and the maximum ratio is 4.5. Bin boundries for the optimal distribution are marked with "•" characters. Bin boundaries for the experimental distribution are marked with "+" characters.

imality criterion that ratios should be distributed across the bins with equal probability. By appealing to the information-theoretic notion of entropy, it is straightforward to show that an equi-probability partitioning maximizes the information conveyed by a histogram [4]. By examining Fig. 1 it is clear that the optimal distribution will partition ratio space finely close to 1 and sparsely close to R^{max}.

Let us assume that there are n bins per sensor channel, where the ith bin is sensitive to ratios in the interval $[x_{i-1}, x_i]$ ($x_0 = 1$ and $x_n = R^{max}$). We must find the x_i which satisfy:

$$\Pr\left(\max\left(\frac{A}{B}, \frac{B}{A}\right)\right) \in [1, x_i] = \frac{i}{n} \tag{7}$$

Using (6) the problem of determining x_i reduces to finding the roots of the polynomial:

$$\frac{1}{\left(R^{\max}-1\right)^2}\left(\left(R^{\max}\right)^2 + 1 - x_i - \frac{\left(R^{\max}\right)^2}{x_i}\right) = \frac{i}{n} \tag{8}$$

However the partition defined in (8) is appropriate if, and only if, the ratio at a color edge is truly invariant to illumination change. Any illumination-induced variance in ratios may cause migration across bin boundaries. *Ratio migration* is most likely to occur when ratio space is finely partitioned. Thus, where the optimal distribution would be a fine partitioning, the competing demands of *ratio migration* require a coarser partitioning. Conversely, where the optimal distribution samples coarsely, ratio migration is less likely to occur; consequently ratio space can be sampled more finely. The actual ratio-space partition used in the color constant color indexing experiments is shown in Fig. 1. A more detailed discussion of the problem of ratio migration is given in [4].

Histogram matches must also be normalized in a slightly different way for ratio histograms than for Swain's color histograms. Swain normalizes by the number of pixels in the model histogram; whereas, we normalize relative to the maximum number of the counts found in the image and model histograms. We do not segment objects from the background prior to calculating model histograms, so Swain's normalization can result in an image with a lot of color edges being falsely matched to a model with fewer color edges, since by chance some of the large number of histogram counts from the image may match some of the limited number of counts from the model. Normalizing with respect to the model's smaller histogram in effect says that very few matches are expected and so overemphasizes the chance matching. A similar problem would arise if intersections were normalized by the count in image histograms. Normalizing on the maximum of the image and model histograms ensures that a good match occurs only when the intersection is large and the two histograms are of similar size.

V. Test Results

The color constant color indexing algorithm performs well on a variety of real and synthetic images. Objects are correctly identified despite substantial changes in the spectral power distribution of the illuminant. Unsurprisingly, Swain's algorithm performs poorly when the illumination changes. It should be noted that in the tests of Swain's algorithm we use RGB histograms, not opponent-color histograms (he tests both) and prior background segmentation is not performed on the model images.

To evaluate color constant color indexing we first consider whether or not ratios suffice for Swain's original problem under controlled illumination. Second, on synthetic images for which the surface reflectances, illuminants, and camera parameters can be completely controlled, we test how the two methods compare. Finally, we test both methods on real images where the illumination is allowed to vary.

A. Tests of the Ratio Representation

Even if color ratios are independent of illumination, this says little about ratios as a representation for color indexing. Are ratio histograms sufficient to discriminate between a large number of objects?

To answer this question, we ran the color constant color indexing algorithm on the database of images Swain used in his experiments. The full database of 66 images is shown in color in [15] (Fig. 4, p. 30). Swain's algorithm imposes few restrictions on the objects except that they be multicolored. A shirt thrown on the floor is a typical example. Its position and orientation can change, its shape can change, and even some occlusion can be tolerated, such as when the shirt sleeve falls across the shirt. Two-dimensional position and orientation have no effect on the histogram so they make no difference to the matching. A change in shape that preserves the color areas will also leave the histogram unchanged; however, occlusion will affect the histogram and may reduce the match confidence. Swain's algorithm handles substantial changes in three-dimensional orientation by storing several views of the object in the model database. Of course for Swain's algorithm the illumination conditions must be the same for the object and the model. As well the size must remain roughly the same, which means that the camera's distance to the object must be similar to what it was to the model.

We eliminated 11 of Swain's 66 model images having saturated responses, because ratios relative to saturated pixels cannot be expected to be constant. For our test, then, the model database contains histograms of 55 images and a second set of 24 different images of the same objects taken in different positions and orientations is matched against this database. The 24 test objects are shown (originals of course are color) in Fig. 2 and their corresponding correct matching models in Fig. 3.

Each algorithm's match performance is assessed with reference to three indicators: match rankings, percentile match, and match tolerance. The position of the correct match in the sorted list of match values is called its rank, so an image is correctly identified if it has

Fig. 2. Images of 24 test objects shown here in black and white.

Fig. 3. The 24 model objects corresponding to the test objects shown in Fig. 1 that were correctly identified by the algorithm. These same images are shown in color in [14] (Fig. 4, p. 30).

rank 1. The match percentile for each image is defined as $\frac{N-r}{N-1}$, where r is a rank and N is the number of models. Each image is also matched with a certain tolerance relative to the next best matching model. If the correct match has rank i then the match tolerance is $m_i - m_{i-1}$, where m denotes match value. An algorithm that correctly identifies images most of the time, but with high average tolerance, may be preferable to one that correctly identifies images more often, but with lower average tolerance.

Table I illustrates the match performance for four algorithms. Swain's, ours with the Laplacian of Gaussian operator ($\sigma = 1.0$), ours with the simple Laplacian operator, and ours with the first derivative operators. Firstly, as Swain reports, color-indexing works well. The second algorithm, color constant indexing with smoothing, shows reasonable performance—19 of the 24 images have first place rankings. However, match tolerance is much reduced and, more importantly, two of the images are very poorly matched—ranks of 18 and 27.

TABLE I
COMPARATIVE PERFORMANCE: SWAIN'S IMAGES

Indexing Algorithm	*Rank: 1*	*2*	*3*	*> 3*	*Av. Percentile*	*Av. Tolerance*
Swain's	23	2	0	0	0.999	0.1212
Laplacian of Gaussian ($\sigma = 1$)	19	0	2	3	0.961	0.0613
Simple Laplacian Indexing	21	2	3	0	0.997	0.0986
1st derivative indexing	22	2	0	0	0.998	0.1023
Grey-level Indexing	13	1	2	8	0.958	0.0208

The third row of Table I gives the match statistics for a simple 3×3 Laplacian mask without Gaussian smoothing. Match performance is increased. Finally, the last row of Table I gives the match statistics for color constant color indexing using the set of four directional derivatives. Because the directional derivatives operate independently, information from each of the four directions contributes to the ratio histogram. Since the ratio histograms perform nearly as well as Swain's color area histograms, we conclude that ratios are a sufficiently rich representation for color constant indexing.

The poorer performance of the Laplacian of Gaussian indexing can in large part be attributed to the effects of too much smoothing. Swain used reduced images of resolution 128×90, so the addition of even a small amount of Gaussian smoothing means that the Laplacian operator may straddle more than one edge at a time. For our higher resolution images (detailed in the forthcoming sections), however, Laplacian of Gaussian indexing performs comparably with indexing using the other derivative operators.

The last row in Table I shows the performance of the 1st derivative indexing algorithm using only single-band (in this case the green channel), grey-level images. While the recognition rates are certainly not as high as when color is used, they are surprisingly good.

B. Tests on Synthetic Images

To the extent that changes in the spectral power distribution of the illumination are modelled by a single scalar multiplication in each sensor channel, the ratio histograms should be relatively illumination independent. To evaluate the coefficient rule approximation (4) in color ratio indexing, we constructed synthetic images using various measured spectra. These images are free from noise, specularities, and other confounding processes that could confuse object identification. As such, they represent a minimal world for object identification.

Surface reflectances were chosen from a set of 40 reflectance spectra that included the Macbeth color checker spectra [13], ceramic spectra [1], and four of the natural reflectances (#98, #106, #203, #324) measured by Krinov [10]. Seven illuminants were used, from reddest to bluest; these are CIE standard A, a 3,600K blackbody radiator, D48, D55, D65, D75, D100. Fig. 4 shows the sensor response functions derived from the Kodak Wratten filters #25 (red), #58 (green), and #47B (blue) by multiplying by the spectral sensitivity function of our CCD camera. The response of sensor class k, $R_k(\lambda)$, to an illumination $E(\lambda)$ striking a surface reflectance $S(\lambda)$ was calculated as:

$$\rho_k = \int E(\lambda)\, S(\lambda)\, R_k(\lambda) d\lambda \tag{7}$$

Thirty synthetic mondrian objects were generated; a mondrian contains numerous, two dimensional rectangular patches of uniform reflectance placed side by side. Each mondrian has the same overall size but contains between four and ten (randomly selected) surface reflectances. If a mondrian has m patches, then these are distributed according to the formula: patches in x direction $= \lceil \sqrt{m} \rceil$ and patches in y direction $= \left\lceil \frac{m}{\sqrt{m}} \right\rceil$. Patches are, as far as possible, of uniform size. For example, the mondrian for the case of $m = 7$ has three patches in the first row, three in the second, and one in the third.

For each illuminant, images of the 30 mondrians were generated. To separate the issue of brightness change from that of hue change in the illumination, the illuminant spectra were normalized such that their squared area is one. Without loss of generality, the mondrians imaged under D55 are used as the model set. Match results for Swain's algorithm and for color constant color indexing (using the Laplacian of Gaussian index) are given in Table II. Note the second column displays the number of match failures. An algorithm fails to identify an image if the intersection with the correct model is zero. If this is the case the match rank is undefined.

As expected, Swain's algorithm performs badly—155 of the 180 mondrians have a zero intersection with the correct model. Indeed, color indexing performs so badly that it is not meaningful to discuss average percentile match or average tolerance. The need for some form of color constancy is readily apparent.

Color constant color indexing performs extremely well. All 180 mondrians are correctly identified and with high tolerances.

The poor performance of Swain's color indexing is graphically ill

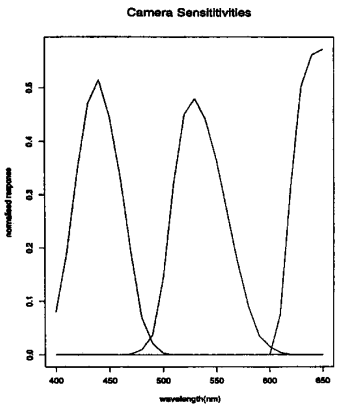


Fig. 4. The camera response function of our CCD camera multiplied by the Kodak Wratten filters numbers #25, #47B, and #58.

TABLE II
COMPARATIVE PERFORMANCE: IDENTIFYING 30 SYNTHETIC MONDRIANS UNDER SIX DIFFERENT, SPATIALLY CONSTANT ILLUMINATIONS

Indexing Algorithm	*Rank: 1*	*Failures*	*Av. Percentile*	*Av. Tolerance*
Swain's	20 out of 180	155	N/A	N/A
Laplacian of Gaussian ($\sigma = 1$)	180 out of 180	0	1.000	.568997

ustrated in Fig. 5. On the left-hand side of the figure the intensity, in each of the three sensor channels, of a test mondrian under CIE A (a reddish illuminant) is shown. This is contrasted, on the right of the figure, with the appearance of the same mondrian under D100 (a bluish illuminant). A large color shift has occurred; red responses become weaker and blue responses stronger. Since the input colors change so dramatically it is unsurprising that color indexing performs poorly.

We calculated the Laplacian of Gaussian ($\sigma = 1.0$) of the log color image for the same mondrian under the same illuminants. We contrast the derivative images in Fig. 6. Under both illuminants the same derivative image is observed—there is no "color" shift—so color constant color indexing will work well.

B.1 A Test with Varying Illumination

The assumption of a single point source illuminant located at infinity is unrealistic. In practice most real world scenes have a complex illumination field—the intensity and spectral distribution of the incident illuminant vary across the scene.

To test color constant color indexing under illumination that varies spatially in both intensity and spectral composition, we generated more test mondrians. From left to right across the mondrian the illumination is calculated as $\alpha * \text{CIE\_A} + (1 - \alpha) * \text{D100}$ (where $\alpha = (x - \text{left})/(\text{right} - \text{left})$ and x is the x coordinate of the mondrian). Given these test mondrians (and the models calculated previously)

Color constant color indexing has perfect match success. Moreo we ran color indexing and color constant color indexing. The respective match statistics are listed in Table III. Over the average tolerance of the matches is high. In contrast color indexing performs poorly

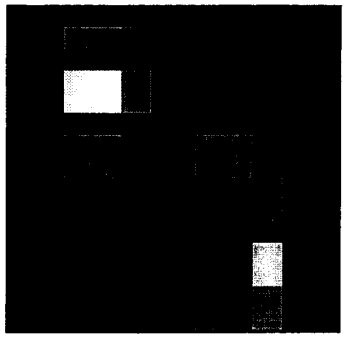


Fig. 5. The red (top), green (middle) and blue (bottom) intensity images of a synthetic mondrian viewed under CIE A (reddish) illuminant are on the left. On the right are images of the same mondrian viewed under the D100 (bluish) iluminant. Note how much the intensities change between the left and right columns. Color histograms based on the two columns will differ substantially.

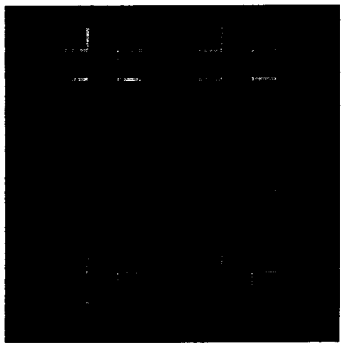


Fig. 6. The result of applying the Laplacian of Gaussian operator to the mondrian images in Fig. 5. Note that the derivative images change very little between the left and right columns even though the illumination has changed from CIE A to D100. Ratio histograms based on the two columns will, therefore, also be very similar.

TABLE III
COMPARATIVE PERFORMANCE: SYNTHETIC IMAGES WITH ILLUMINATION VARYING SPATIALLY IN INTENSITY AND SPECTRAL COMPOSITION

Indexing Algorithm	*Rank: 1*	*Failures*	*Av. Percentile*	*Av. Tolerance*
Swain's	7 out of 30	12	N/A	N/A
Laplacian of Gaussian ($\sigma = 1$)	30 out of 30	0	1.000	.567207

with 12 out of 30 mondrians having a zero intersection with the correct model histogram.

B.2 Tests Using Human Cone Sensitivities

Using the Vos and Walraven [17] estimate of human cone sensitivities as sensors, see Fig. 7, we can generate synthetic images and examine to what extent color constant color indexing would be affected if the the cones were the sensors. In Table IV we present the theoretical performance results using the cones. The first row contains the match statistics for all six test illuminants, i.e., 180 mondrians (set1). The second row contains statistics for the test illuminants excluding CIE A and D100 (set2)—120 mondrians. CIE A and D100 represent the extremes in the spectral variation of the illuminants.

A comparison of Tables II and IV reveals that the broadband nature of the cones does impair the algorithm's performance. Match performance is increased when CIE A and D100—the extremes of the spectral variation in the illumination—are factored out. Lower rankings result and both the average match tolerance and average percentile match increase.

TABLE IV
LAPLACIAN OF GAUSSIAN INDEXING ($\sigma = 1$) USING HUMAN CONE SENSITIVITIES

Images	*Rank: 1*	*2*	*3*	*> 3*	*Failures*	*Av. Percentile*	*Av. Tolerance*
Set1	135 out of 180	10	7	18	10	0.97	0.194
Set2	108 out of 120	1	3	5	3	0.99	0.256

Finlayson et al., [5] recently demonstrated that *sharper* (or more narrowband) sensor curves can be created by taking linear combinations of the human cone sensitivities. Fig. 7 contrasts the original cone sensitivities with sharpened counterparts. One would expect that response ratios relative to these sharpened curves to be more invariant to illuminant change and consequently provide a more useful index for object recognition. We repeated the experiment, detailed above, using the sharpened sensors; the results are shown in Table V.

Performance has increased dramatically. Almosts all of the 180 test mondrians are matched with rank 1 and there is only one failure. Removing the extremes in test illuminants (CIE A and D100) results in almost perfect match success. The import of this is that color constant color indexing need not require a visual system to have narrowband sensors; it is sufficient that there exist linear combinations which are narrowband.

TABLE V
LAPLACIAN OF GAUSSIAN INDEXING ($\sigma = 1$) USING *SHARPENED* HUMAN CONE SENSITIVITIES

Images	*Rank: 1*	*2*	*3*	*> 3*	*Failures*	*Av. Percentile*	*Av. Tolerance*
Set1	172 out of 180	4	3	0	1	0.998	0.384
Set2	119 out of 120	1	0	0	0	1.000	0.457

C. Tests on Real Images

Under illuminants of three different color temperatures (3,600K, 4,200K, and 5,400K) pictures were taken of 11 objects comprised of three T-shirts, three cereal/detergent boxes, three sweaters, a Sun User's manual, and a child's toy, for a total of 33 images. When the illumination was changed, so were other viewing conditions; shirts and sweaters were deformed, objects were rotated and occluded. The camera responds linearly with intensity and its spectral response functions are as plotted in Fig. 4.

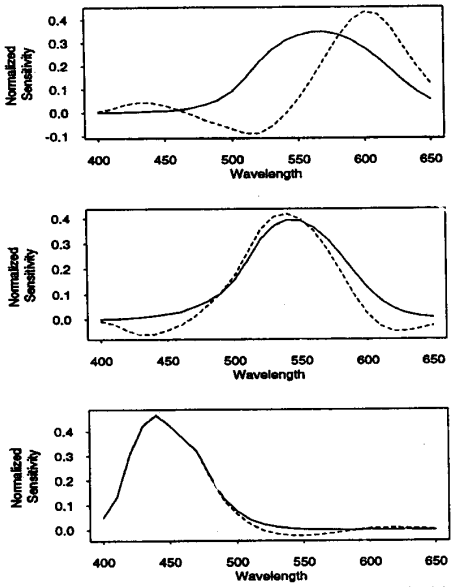


Fig. 7. Vos Walraven fundamentals (solid line) are contrasted with the sharpened sensitivities (dotted line).

Table VI summarizes the match statistics for color constant color indexing. A model database was constructed using the 11 images taken under one illuminant and then the other 22 images were matched against it. This was repeated for each illuminant. In the table, each row corresponds to a different choice of model database. Performance is good and is independent of the illuminant.

TABLE VI
REAL IMAGES WITH SPATIALLY VARYING ILLUMINATION: LAPLACIAN OF GAUSSIAN INDEXING ($\sigma = 1$)

Database	*Rank: 1*	*2*	*Av. Percentile*	*Av. Tolerance*
3600K	21	1	0.995	0.165
4200K	22	0	1.000	0.145
5400K	22	0	1.000	0.137

TABLE VII
REAL IMAGES WITH SPATIALLY VARYING ILLUMINATION: SWAIN'S ALGORITHM

Database	*Rank: 1*	*2*	*3*	*> 3*	*Av. Percentile*	*Av. Tolerance*
3600K	14	5	0	3	0.90	0.08
4200K	10	2	3	7	0.768	0.066
5400K	10	4	3	5	0.80	0.071

Table VII tabulates the results for Swain's algorithm. While its performance is poor under varying illumination, it is better than it might have been. This is partly due to the experimental conditions under which the pictures were taken. The color temperature of the illuminant was changed by placing filters in front of the light source. Unfortunately, these filters also diminished the intensity of the light. To compensate for this, camera gain and aperture were adjusted. All pictures were made to have pixels which lie close to the maximum camera response, i.e., 255. Both aperture and gain adjustments are linear so should not affect ratio constancy.

Normalizing images in this way encourages Swain's algorithm to work, since these camera adjustments create an approximate form of color constancy. Nonetheless, even under these favorable experimental conditions Swain's algorithm performs badly. The optimal choice of model set appears to correspond to the 3,600K illumination. However, even here 36% of images are wrongly identified; this is extremely poor performance given the small database size. Furthermore, rankings of five and seven were recorded. Given that there are only 11 models, such match performance is clearly unacceptable.

D. Future Work: K-Nearest Neighbor Classification

We expect that color constant color indexing would perform even better if histogram matching were to be based on K-nearest neighbor classification. For K-nearest neighbor classification (see Duda and Hart [3] for a description), several images of each model object taken under different illumination conditions would be stored in the model database. The image of an unknown object would then be identified on the basis of the K best matches to the model database.

The K-nearest neighbor method requires that histogram intersection be a metric. Fortunately, Swain has already demonstrated that if two histograms are of the same size then their intersection is a distance metric. In particular, histogram intersection is equivalent to the scaled sum of absolute differences, commonly referred to as the *city-block* metric. Consider the intersection of two histograms M and I each with n bins.

$$if \sum_{i=1}^{n} M_i = \sum_{i=1}^{n} I_i = T \Rightarrow 1 - H(I, M) = \frac{1}{2T} \sum_{i=1}^{n} |I_i - M_i|$$

For the case of ratio histograms, ratios close to one are eliminated so for their intersection to be metric the ratio histograms must be

normalized to have equal total bin counts. Results for color constant color indexing when this metric condition is enforced are given in Table VIII.

TABLE VIII
MATCHING WHEN HISTOGRAM INTERSECTION IS A METRIC

Database	*Rank: 1*	*2*	*3*	*> 3*	*Av. Percentile*	*Av. Tolerance*
Swain's Images	22	0	0	2	0.992	0.098
Synthetic Images	180	0	0	0	1.00	0.558
Real Images	19	3	0	0	0.986	0.11

As the table shows, normalization has only a slightly deleterious effect on the matching, so even when the metric condition is satisfied ratio histograms provide a good representation that remains stable under changes in illumination.

When the histogram of the image of an unknown object is matched to the model database, the K best matches are used. Of course, if all K matches are to the same object then this is a strong match. In general, however, it should be sufficient to select the most numerously matched object. Since Swain's method is vulnerable to illumination changes it cannot be generalized to the K-nearest neighbor method in that the illumination distorts the histogram so significantly that we cannot expect the K nearest neighbors ($K > 1$) to all represent the same object.

VI. CONCLUSION

Color constant color indexing based on comparing the color of neighboring locations works very well. Our experiments show that it extends Swain and Ballard's color-based object recognition scheme to scenes of uncontrolled illumination without affecting overall performance. The strategy of histogramming color ratios (in essence edges) instead of areas circumvents the need for an illumination-independent description of the actual colors in the scene.

VII. ACKNOWLEDGMENTS

This work has been supported by the Center for Systems Science at Simon Fraser University and the Natural Sciences and Engineering Research Council of Canada. We also wish to thank Michael Swain for providing us with his image database.

REFERENCES

[1] R.S. Berns and K.H. Petersen, "Empirical modelling of systematic spectrophotometric errors," *Color Res. Appl.*, vol. 4, p. 243, 1988.
[2] G. Buchsbaum and A. Gottschalk, "Trichromacy, opponent colours coding and optimum colour information transmission in the retina," *Proc. R. Society London B.*, vol. 220, pp. 89–113, 1983.
[3] R.O. Duda and P.E. Hart, *Pattern Classification and Scene Analysis*, John Wiley & Sons, 1973.
[4] G.D. Finlayson, *Colour Object Recognition*. Simon Fraser Univ. School of Computing Science, 1992.
[5] G.D. Finlayson, M.S. Drew, and B.V. Funt, "Spectral sharpening: Sensor transformations for improved color constancy," *J. Opt. Soc. Am. A*, vol. 11, pp. 1553-1563, 1994.
[6] D. Forsyth, "A novel algorithm for color constancy," *Int'l. J. Computer Vision*, vol. 5, pp. 5–36, 1990.
[7] B.V. Funt and M.S. Drew, "Color constancy computation in near–Mondrian scenes using a finite dimensional linear model," *Computer Vision and Pattern Recognition Proc.*, IEEE Computer Society, pp. 544–549, June 1988.
[8] B.V. Funt and J. Ho, "Color from black and white," *Proc Second Int'l. Conf. Computer Vision*, Tarpon Springs, pp. 2–8. IEEE Computer Society, Dec. 1988, and *Int'l. J. Computer Vision*, vol. 3, pp. 109-117, 1989.
[9] B.K.P. Horn, "Determining lightness from an image," *Computer Vision, Graphics, and Image Processing*, vol. 3, pp. 277–299, 1974.
[10] E.L. Krinov, "Spectral reflectance properties of natural formations," *Technical Translation TT-439*, Nat'l. Research Council of Canada, 1947.
[11] E.H. Land and J.J. McCann, "Lightness and retinex theory," *J. Opt. Soc. Am.*, vol. 61, pp. 1–11, 1971.
[12] D. Marr, *Vision*. Freeman, 1982.
[13] C.S. McCamy, H. Marcus, and J.G. Davidson, "A color-rendition chart," *J. App. Photog. Eng.*, pp. 95–99, 1976.
[14] C.L. Novak and S.A. Shafer, "Supervised color constancy using a color chart," Technical Report CMU-CS-90-140. Carnegie Mellon Univ. School of Computer Science, 1990.
[15] M.J. Swain and D.H. Ballard. "Color indexing," *Int'l. J. Comput. Vision*, vol. 7(1), pp. 11–32, 1991.
[16] G. West and M.H. Brill, "Necessary and sufficient conditions for von Kries chromatic adaption to give colour constancy," *J. Math. Biol.*, vol. 15, pp. 249–258, 1982.
[17] G. Wyszecki and W.S. Stiles, *Color Science: Concepts and Methods, Quantitative Data and Formulas*. New York: John Wiley & Sons, 2nd ed., 1982.

Linear Time Euclidean Distance Transform Algorithms

Heinz Breu, Joseph Gil, David Kirkpatrick, and Michael Werman

***Abstract*—Two linear time (and hence asymptotically optimal) algorithms for computing the Euclidean distance transform of a two-dimensional binary image are presented. The algorithms are based on the construction and regular sampling of the Voronoi diagram whose sites consist of the unit (feature) pixels in the image. The first algorithm, which is of primarily theoretical interest, constructs the complete Voronoi diagram. The second, more practical, algorithm constructs the Voronoi diagram where it intersects the horizontal lines passing through the image pixel centers. Extensions to higher dimensional images and to other distance functions are also discussed.**

***Index Terms*—Distance transform, Voronoi diagram, algorithm, Euclidean distance.**

I. INTRODUCTION

A two-dimensional binary *image* is a function, I, from the elements of an n by m array, referred to as *pixels*, to {0, 1}. Pixels of unit (respectively, zero) value are referred to as *feature* (respectively, *background*) pixels of the image. We associate the pixel in row r and column c with the Cartesian point (c, r). Thus, any distance function defined on the Cartesian plane induces a distance function on the space of image pixels. For a given distance function, the *distance transform* of an image I is an assignment, to each pixel p, of the distance between p and the closest feature pixel in I. The *nearest-neighbor transform* of an image is an assignment, to each pixel p, of the identity (or distance information sufficient to compute the identity in $O(1)$ time) of a feature pixel closest to p. Assuming that computing the distance between two pixels is an $O(1)$ time operation, it should be clear that the distance transform can be constructed from the near-

Manuscript received July 29, 1993; revised Nov. 20, 1994.
H. Breu and D. Kirkpatrick are with the Department of Computer Science, University of British Columbia, Vancouver, Canada V6T 1Z5.
J. Gil is with The Faculty of Computer Science, Technion—Israel Institute of Technology, Haifa 32000, Israel; e-mail yogi@CS.Technion.AC.IL; yogi@NeXt.CS.Technion.AC.IL.
M. Werman is with the Institute of Computer Science, The Hebrew University, of Jerusalem 91904, Jerusalem, Israel; e-mail werman@cs.huji.ac.il.
IEEECS Log Number P95048.