Indexing Via Color Histograms

Michael J. Swain[†]and Dana H. Ballard

Department of Computer Science
University of Rochester
Rochester, NY 14627, USA

Abstract

The color spectrum of multicolored objects provides a a robust, efficient cue for indexing into a large database of models. This paper shows color histograms to be stable object representations over change in view, and demonstrates they can differentiate among a large number of objects. It introduces a technique called *Histogram Intersection* for efficiently matching model and image histograms. Color can also be used to search for the location of an object. An algorithm called *Histogram Backprojection* performs this task efficiently in crowded scenes.

1 Introduction

Computer vision is moving into a new era in which the aim is to develop visual skills for robots that allow them to interact with a dynamic, unconstrained environment. To achieve this aim, new kinds of vision algorithms need to be developed which run in real time and subserve the robot's goals. Two fundamental goals are determining the identity of an object with a known location, and determining the location of a known object. Color can be successfully used for both tasks.

Color has been neglected recently as a recognition cue, although it has been used in earlier work [7]. One reason for this may have been the lack of good algorithms for *color constancy*, that is, perceiving a stable perception of color over varying light conditions, as people do in most circumstances. However, recently there has been great progress in correcting for both the chromaticity of the illuminant [5, 2] and for geometric effects such as specularity [3]. Given that reasonable color constancy can be achieved, color has enormous value in recognition because it is a local surface property that is view invariant and largely independent of resolution. Shape cues, by contrast, are highly resolution dependent, and only a highly restricted set are view invariant (e.g. corners, zeros of curvature).

Perhaps another reason that color has not been used is that it is not intrinsically related to the object's identity in the way that other cues, e.g., form, are. This view is well represented by Biederman [1]:

> "Surface characteristics such as color and texture will typically have only secondary roles in primal access ... we may know that a chair has a particular color and texture simultaneously with its volumetric description, but it is only the volumetric description that provides efficient access to the representation of CHAIR."

However, this opinion is easily challenged. There are many examples from nature where color is used by animals and plants to send clear messages of enticement or warning. The manufacturing sector uses color extensively in packaging to market goods. Robotic vision systems can also use representations that are heavily personalized to achieve efficient behaviors. For example, it may not be helpful to model coffee cups as being red and white, but *yours* may be, and that color combination is very useful in locating it.

1.1 What vs. Where

A significant feature of the gross organization of the primate visual brain is the specialization of the temporal and parietal lobes of visual cortex. The parietal cortex seems to be subserving the management of locations in space whereas the temporal cortex seems to be subserving the identification of objects in the case where location is not the issue. In a striking experiment by Mishkin [6], monkeys with parietal lesions fail at a task that requires using a relational cue but have

*This work was supported by NSF research grant DCR-8602958.

†Current address: Department of Computer Science, University of Chicago, 1100 E. 58th St., Chicago IL 60637, swain@gargoyle.uchicago.edu

CH2934-8/90/0000/0390$01.00 © 1990 IEEE

no trouble performing a very similar task that requires using a pattern cue. The reverse is true for temporal lesions. Why should the primate brain be specialized in this way? If we think generally about the problem of relating internal models to objects in the world, then one way to interpret this "What/Where" dichotomy is as a suggestion that image interpretation, the general problem of associating many models to many parts of the image simultaneously, is either too hard or unnecessary, or both (see Table 1.1). In order to build vision systems which function in real-time, perhaps the problem must be simplified.

The approach taken in Section 2 is to answer the question "What" assuming that the approximate location of the object is known. This is done by using a color histogram as the representation, which counts how much of each color occurs in the image.

Section 3 shows how a model histogram can be back-projected onto an image to solve one aspect of the "Where" problem, which is the location of an object of known identity in the image. Again, the view invariance of color precludes the calculation of orientation but simplifies the algorithm enormously.

1.2 Color Histograms

Given a discrete color space defined by some color axes (e.g. red, green, blue), the color histogram is obtained by counting the number of times each color occurs in the image array. Histograms are invariant to translation and rotation about an axis perpendicular to the image plane, and change only slowly under change of angle of view, change in scale and occlusion. Because histograms change slowly with view, a three-dimensional object can be adequately represented by a small number of histograms, corresponding to a set of canonical views [4].

Both the object identification and object location implementations described in the following sections use color histograms to represent objects.

2 Object Identification via Histogram Intersection

Because the model database may be large, we can only afford a highly restricted amount of processing per model, but at the same time we must be able to overcome the problems that hinder recognition, most importantly

- distractions in the background of the object,
- viewing the object from a variety of viewpoints,
- occlusion,
- varying lighting conditions.

The matching method proposed here, called *Histogram Intersection*, is robust to the first three problems; the last is left to a color constancy module that operates on the input prior to the histogram stage. Histogram Intersection is also extremely efficient and easy to implement.

2.1 Algorithm

Histogram Intersection matches the image color histogram with histograms of each of the models in the database. The higher the match value the better the fit to the model. Given a pair of histograms, I (image) and M (model), each containing n buckets, the intersection of the histograms is defined to be

$$\sum_{j=1}^{n} \min(I_j, M_j).$$

where j ranges over each color in the histograms. The result of the intersection of a model histogram with an image histogram is the number of pixels from the model that have corresponding pixels of the same color in the image. To obtain a fractional match value between 0 and 1 the intersection is normalized by the number of pixels in the model histogram. The match value is then

$$H(I, M) = \frac{\sum_{j=1}^{n} \min(I_j, M_j)}{\sum_{j=1}^{n} M_j}.$$

The Histogram Intersection match value is not reduced by distracting pixels in the background. This is the desired behavior since complete segmentation of the object from the background is difficult to guarantee. The match value is only increased by a pixel in the background if

- the pixel has the same color as one of the colors in the model, and
- the number of pixels of that color in the object is less than the number of pixels of that color in the model.

Histogram Intersection is robust to scale changes but not scale invariant. However, there are a number of ways of determining the approximate depth of an object, from laser or sonar range finders, disparity, focus or touching the object with a sensor. The depth value combined with the known size of the object can be used to scale the model histogram. Alternatively, if it is possible to segment the object from the background and it is not significantly occluded the image histogram can be scaled to be the same size as the model histogram.

The Histogram Intersection approach to histogram matching can be related to classical pattern recognition by considering each bin in the color histogram as a feature. An object is then a point in an n-dimensional

<table>
<tr><td colspan="2" rowspan="2"></td><td colspan="2">Object to Match Against</td></tr>
<tr><td>One</td><td>Many</td></tr>
<tr><td>Image Portions</td><td>One</td><td></td><td>Identification: trying to identify an object whose location can be fixated</td></tr>
<tr><td></td><td>Many</td><td>Location: trying to locate an object whose identity is known</td><td>Image interpretation: Too hard?</td></tr>
</table>

Table 1: The biological organization of cortex into What/Where modules may have a basis in computational complexity.

space, where n is the number of bins in the histogram. If histograms are scaled to be the same size, then Histogram Intersection is equivalent to a scaled sum of absolute differences, that is, if

$$\sum_{i=1}^{n} M_i = \sum_{i=1}^{n} I_i = T$$

then

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

The function $1 - H$ therefore defines a distance metric (a scaled *city-block* metric). If histograms are not the same size, as when the scaling is done by distance, then $1 - H$ is not a metric because of the asymmetry between images and models.

Histogram Intersection is an efficient way of matching histograms. Its complexity is linear in the number of elements in the histograms. Two 16x16x8 histograms can be matched in 2 milliseconds on a SUN Sparcstation 1 (a 12 MIP RISC machine). The color histograms are also efficient to compute using image processing hardware. Generating a histogram from a 512 x 485 image takes about 40 milliseconds using a MaxVideo FeatureMax board, including the time needed to transfer the histogram to the host.

2.2 Experimental Results

An experimental test of histogram intersection suggests that the technique is capable of differentiating among a large database. For a 66 object database of children's shirts, cereal boxes and cleaning detergents, the correct model is the best match 90% of the time and is always one of the top two matches (see [8]). Other, more expensive, matching techniques can be used to verify which of the top scoring models is the correct one, so it is not crucial that the correct model is always the best match. In the experiment the models were segmented from the background prior to generating the model histograms. No segmentation was performed on the images of the unknown objects.

Experiments with three dimensional objects suggest that a fairly small number of different two-dimensional views can be an adequate representation for a three-dimensional object. For example a number of different views of a Snoopy doll were obtained. Even at 45 degrees rotation off the direction the model was obtained, the match value (about 0.6) was higher than 99 percent of the false matches. Experiments show that histogram intersection is insensitive to occlusion, and image and histogram resolution. Without processing the color signals by a color constancy algorithm, histogram intersection is sensitive to changing light conditions [8].

Recognition Times (milliseconds)			
	Database Size		
	19	37	70
Histogram Intersection	38	73	150
Incremental Intersection	15	15	15

Table 2: Recognition times as a function of database size for the standard algorithm Histogram Intersection and the fast indexing scheme Incremental Intersection, using the 10 largest image bins. Recognition accuracy was equivalent for both algorithms. Timings were made on a SUN SPARCstation 1.

Most of the information needed for identification is carried by large buckets in the histograms. An algorithm called Incremental Intersection takes advantage of this fact to index extremely efficiently into very large databases without sacrificing accuracy [8]. Although still linear in the size of the database, the constant of proportionality is extremely low. The efficiency of Histogram Intersection and Incremental Intersection is compared in Table 2.2.

3 Object Location via Histogram Backprojection

The previous sections discussed recognizing an unknown object whose location is known, the "Identification" box in Table 1.1. This section discusses the complementary task, locating a known object, the "Location" box in the same table. This task can also be accomplished using color histograms and an algorithm called *Histogram Backprojection.*

Histogram Backprojection answers the question "Where are the colors in the image that belong to the object being looked for (the *target*)?" The answer is given in such a way so that the colors that appear in other objects besides the target are de-emphasized so that they are less likely to distract the search mechanism. Experiments show that the technique works for objects in cluttered scenes under realistic conditions.

As in Histogram Intersection, in Histogram Backprojection the model (target) is represented by its multidimensional color histogram M. The histogram of the image, I, is also computed and a third histogram R, which is the ratio of M divided by I, is computed. It is this histogram R which is backprojected onto the image, that is, the image values are replaced by the values of R that they index. The backprojected image is then convolved by a mask, which for compact objects of unknown orientation could be a circle with the same area as the expected area subtended by the object. The peak in the convolved image is the expected location of the target, provided the target appears in the image.

More precisely, let $h(c)$ be the histogram function which maps a color c (a three-dimensional value) to a histogram bin. Let D^r be a disk of radius r:

$$D^r_{x,y} = \begin{cases} 1 \text{ if } \sqrt{x^2+y^2} < r \\ 0 \text{ otherwise} \end{cases}$$

Define the *loc* function to return a pixel (x, y) with the value of its argument, and let the $*$ symbol denote convolution. Then Histogram Backprojection can be written:

1. for each histogram bin j do
   $R_j := \frac{M_j}{I_j}$
2. for each x,y do
   $b_{x,y} := \min\ (R_{h(c_{x,y})}, 1)$
3. $b := D^r * b$
4. $(x_t, y_t) := \text{loc}(\max_{x,y} b_{x,y})$

Because the convolution can be carried out on a reduced resolution image, *Histogram Backprojection* is very efficient. It has been implemented on a MaxVideo image processor, on which it runs in real time (0.1 seconds per frame).

4 Previous Work

Early work on classification using color invariably assumed the objects were monochromatic [7]. Recently [9] used color to find multi-colored objects with a mobile camera. Their match technique is histogram-based, like Histogram Intersection, and influenced the development of the match techniques in this paper. Because they desired a scale invariant match value it has important differences. They compare ratios of peaks in the model and image histograms. This technique fails for uni-colored objects, and it can suffer severely under occlusion which cuts across most of the colored regions, or by the presence of another object in the image whose histogram peaks overshadow some of the peaks of object being matched.

5 Summary

We have identified two important sub-problems of the image identification problem, identifying an object in a known location, and locating a known object. Both of these problems can be achieved in real-time using color histograms as a representation, for a variety of everyday objects. Our results suggest that color should be considered as a primary cue for these important tasks in computer vision.

References

[1] I. Biederman. Human image understanding: Recent research and a theory. *Computer Vision, Graphics and Image Processing*, 32(1):29–73, 1985.

[2] D. A. Forsyth. *Colour Constancy and its Applications in Machine Vision.* PhD thesis, Robotics Research Group, Department of Engineering Science, Oxford University, 1988.

[3] Gudrun J. Klinker, Steven A. Shafer, and Takeo Kanade. The measurement of highlights in color images. *International Journal of Computer Vision*, 2:7–32, 1988.

[4] J. J. Koenderink and A. J. van Doorn. The singularities of the visual mapping. *Biological Cybernetics*, 24:51–59, 1976.

[5] Laurence T. Maloney and Brian A. Wandell. Color constancy: A method for recovering surface spectral reflectance. *Journal of the Optical Society of America A (JOSA-A)*, 3(1):29–33, January 1986.

[6] Mortimer Mishkin and Tim Appenzeller. The anatomy of memory. *Scientific American*, 1987.

[7] R. Ohlander, K. Price, and D.R. Reddy. Picture segmentation using a recursive region splitting method. *Computer Graphics and Image Processing*, 8:313–333, December 1978.

[8] Michael J. Swain. *Color Indexing.* PhD thesis, University of Rochester, 1990.

[9] Lambert E. Wixson and Dana H. Ballard. Realtime detection of multi-colored objects. In *SPIE Sensor Fusion II: Human and Machine Strategies*, volume 1198, November 1989.