Robust Color Object Detection using Spatial-Color Joint Probability Functions

David Crandall Jiebo Luo
Research & Development Laboratories
Eastman Kodak Company
jiebo.luo@.kodak.com

Abstract

Object detection in unconstrained images is an important image understanding problem with many potential applications. There has been little success in creating a single algorithm that can detect arbitrary objects in unconstrained images; instead, algorithms typically must be customized for each specific object. Consequently, it typically requires a large number of exemplars (for rigid objects) or a large amount of human intuition (for non-rigid objects) to develop a robust algorithm. We present a robust algorithm designed to detect a class of compound color objects given a single model image. A compound color object is defined as having a set of multiple, particular colors arranged spatially in a particular way, including flags, logos, cartoon characters, people in uniforms, etc. Our approach is based on a particular type of spatial-color joint probability function called the color edge co-occurrence histogram (CECH). In addition, our algorithm employs perceptual color naming to handle color variation, and pre-screening to limit the search scope (i.e., size and location) of the object. Experimental results demonstrated that the proposed algorithm is insensitive to object rotation, scaling, partial occlusion, and folding, outperforming a closely related algorithm by a decisive margin.

1. Introduction

Object detection in unconstrained images is an important image understanding task, with potential use in a wide variety of image understanding and content-based indexing applications. Despite years of research attention, there has been little success in creating an algorithm that can reliably detect an arbitrary object in unconstrained images. The best that can be attained in the current state-of-the-art is to build separate algorithms for specific objects or classes of objects.

Building an object detection algorithm for a new object is typically time consuming and labor intensive. There are two basic approaches: training a learning engine on a large amount of exemplars ("machine learning") or using human intuition to craft models for finding the object ("human learning"). Neither approach is easy or universally applicable. Machine learning approaches, while powerful, need a large amount of exemplars under various conditions and may not work well for non-rigid objects. Crafting rules or models for object detection requires extensive human knowledge and intuition, subject to main difficulties including translating human knowledge into rules, generalizing to novel data, and enumerating all possible cases. Further, the resulting detector from either approach is often specialized: a completely new set of rules or new classifier must be generated for each new object.

In this study, we limit our goal to developing a general, repurposable algorithm for detecting a class of objects, namely *compound color objects*, which are objects having a specific set of multiple colors that are arranged in a unique spatial layout. This class of objects includes, for example, flags, cartoon characters, logos, uniforms, signs, etc. The problem is non-trivial because the appearance of compound color objects may vary drastically from scene to scene. Objects like flags and logos often appear on flexible material. For example, a flag is subject to self-occlusion and non-affine distortion, depending on wind conditions. Since orientation of images and objects is not always known, the detector must be invariant to rotation. It must also be robust to color variations due to illuminant changes and inherent color differences from one instance of the object to another.

Object detection is a fundamental problem in computer vision and has received a large amount of attention in the literature. There is a spectrum of object detection approaches, depending on the degree of spatial distortion that must be accommodated. On one end of the spectrum is pixel-by-pixel template matching, which is used for rigid objects *absent* of significant out-of-plane rotation (e.g., frontal face detection). On the other end of the spectrum are flexible models that capture the possible spatial relationships between their component parts. Such an approach is necessary for objects whose spatial arrangements can change significantly (e.g., human bodies and horses in [1]). As one moves toward the latter end of the spectrum, the representations become more abstract but more flexible in the types of distortions that they can handle. However, they also require more high-level knowledge about the target object and become more susceptible to model failures.

Major relevant works includes the following, roughly ordered according to increasing levels of abstraction on the above-described representation spectrum:

- Rowley *et al.* [3] detect faces using a neural network classifier on the intensity patterns in an image. Preprocessing is applied to input images to correct for lighting, contrast and orientation variations.

Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'04)
1063-6919/04 $20.00 © 2004 **IEEE**

– Schneiderman and Kanade [4] detect faces using joint histograms of wavelet features. Their statistical approach allows robustness to variation in facial appearances.
– Oren et. al. [5] use wavelet features to detect pedestrians. Input images are scanned for pedestrians using windows of various sizes and are classified with an SVM.
– Selinger and Nelson [6] represent 3-D objects by several 2-D images taken from different angles. The 2-D images are further abstracted as groups of contour curves. Recognition is performed by exhaustive template matching of the curves.
– Huttenlocher et al. [7] represent objects using edge pixel maps and compare images using the Hausdorff distance between the locations of edge pixels. The Hausdorff distance allows tolerance to some geometric distortion.
– Cootes et al. [8] represent objects using active appearance models (AAMs) that model the shape and grayscale appearance of objects. The models allow detection of moderately flexible objects, like faces.
– Forsyth and Fleck [1] segment an image into candidate horse regions using color and texture features and assembles regions using a graphical model ("body plan") to support the related geometric reasoning.
– Fergus et al. [2] model objects as flexible constellations of parts and use a probabilistic representation for all aspects of the object: shape, appearance, occlusion and relative scale. They learn and recognize object class models from unlabeled and unsegmented cluttered scenes.

In the design of any object detection system, one must choose a suitable representation for comparing the object model to a search image. The representation is typically chosen based on types of distortion that are expected in the object. By definition, the spatial layout of a compound color object is largely fixed, but distortions may occur as a result of camera angle and projection of the object on a non-rigid surface, like flags and logos on fabric. Clearly, intensity or edge patterns are too rigid while graphical models are too fragile for compound color objects.

An attractive representation is color co-occurrence histograms (CCH, also called correlograms in [10]). CCH is a representation that captures the colors within an object as well as some spatial layout information. Chang and Krumm [9] originally proposed an object detection algorithm using CCH. They quantized multiple object models to a small number of colors using a *k*-means clustering algorithm and quantized the test images using the same color clusters. A search image is scanned by comparing the color co-occurrence histograms of large, overlapping regions with those of the model images using histogram intersection. Object locations are refined by a hill-climbing search around regions identified during the rough scan. The example images showed good results for detecting the *same* object amid cluttered scenes. However, the experiments were performed under controlled conditions; all images were taken in a laboratory where object illumination, size, and orientation were kept *constant*. Such assumptions do not hold true for unconstrained images, where illumination and object size can vary widely between images. Also, the computation demands of the algorithm described in [9] are high even though the scale is known due to its exhaustive search procedure.

We present a robust object detection algorithm that is easily re-deployable for most compound color objects using a single model image. Inspired by [9], we use *spatial-color joint probability functions* to perform the detection. Such functions capture the spatial relationships between the colors in an object, but allow for some degrees of distortion. This approach is more flexible than pixel-by-pixel template matching but is simple enough that models can be built without large amounts of training data or input from a human expert. In particular, we propose a novel, more desirable spatial-color joint probability function, referred to as the *color edge co-occurrence histogram (CECH)*.

The major contributions of this work include (1) using perceptual color quantization by color naming to reduce sensitivity to color variations, (2) using a novel color *edge co-occurrence histogram* as an object representation with improved robustness in the presence of geometric distortions, (3) using normalized cross-correlation as the similarity metric for co-occurrence histograms, and (4) using pre-screening to facilitate estimation of rough scale and location and consequently fast search.

The remainder of this paper describes our object detection algorithm in detail. Section 2 describes our methodology, including a perceptual color quantization algorithm, a fast object prescreening algorithm, an object detection algorithm, and techniques for efficient implementation. Section 3 presents experimental results of our algorithm applied to detecting a variety of objects in a variety of different images, with comparisons to the original algorithm in [9]. Finally, we summarize and conclude in Section 4.

2. Methodology

In this study, we assume that the algorithm is provided with a model image *M* of the object being sought, and an input search image *I*, which may contain zero, one, or multiple of the target objects. It is assumed that the target object may appear anywhere in the image, at any reasonable size, with reasonable types and amounts of distortions (*not* limited to affine transforms). Our convention is a coordinate system with the top-left pixel as the origin, *x* values increasing towards the right, and *y* values increasing towards the bottom. For a pixel at location $p = (x,y)$, we use $I(p)$ to represent the color value stored at that location in image *I*.

2.1. Perceptual color quantization by color naming

It is desirable to quantize the color space properly to reduce the dimensionality for computing CCH. In [9], color quantization was simply performed in the RGB color space. However, the appearance of the colors of an object may differ significantly from image to image because of material differences, illumination variations, perceptual color surround effects, etc. The quantization must be designed carefully to ensure that perceptually similar colors are mapped to the same quantized color value, with minimum sensitivity to color variations. RGB color space is known to be sensitive to color variations, prompting researchers to use more desirable color spaces such as Luv and HSV [11].

Humans tend to use a very small set of basic color names to describe colors in objects, discounting subtle or sometimes even substantial color variations. For example, although the American and British flags contain different Pantone® colors, most humans would describe them both as "red," "white," and "blue." A desirable property of quantization for object detection is to create color clusters that are meaningful and nameable for human observers. This would let a human specify a target object in terms of basic colors, e.g., "find all rectangular objects with a blue square that neighbors interleaving red and white stripes" (i.e., U.S. flag). For automatic object detection, a major advantage of using perceptually related color names is that they are very stable with respect to color variations.

Color naming has been applied to image retrieval in [11]. They partition the CIE LAB color space into 2520 segments and assign names to each partition, based on input from two observers. However, a uniform partitioning is problematic because of the nonlinearities in the LAB space. Also, basing the color naming on the ad-hoc observations of a handful of human observers does not take into account the wide variability in human perception of color.

We propose a more principled perception-based approach to quantizing and naming the color space. We use a two-stage process that employs the standard ISCC-NBS Color Names Dictionary [12]. The ISCC-NBS system defines 267 standard color partitions, each with a standard color name and a representative color (called the centroid color). The ISCC-NBS color names are basic colors with a prefix of one or more adjectives, e.g., "Vivid Red," "Strong Reddish Brown," "Light Grayish Yellowish Brown," etc.

An input image, *I*, is first converted into the CIE LAB color space [13]. Next, each of the pixels is assigned to the standard *name* of the closest ISCC-NBS centroid color, according to the Euclidean distance in LAB space. In the second stage, a look-up table is used to map the assigned color names to a smaller set of colors Q_c: *red, green, yellow, blue, orange, purple, brown, white, black,* and *gray*. The look-up table is constructed by mapping each ISCC-NBS color name to the name produced after all adjectives have been removed, with modifications based on input from human observers or the specific object detection task at hand. Although the color quantization approach is conceptually split into two stages, note that the LAB conversion and both stages of quantization can be performed using a single e composite 3D lookup table.

2.2. Color edge co-occurrence histograms (CECH)

The color co-occurrence histogram (CCH) captures information on the spatial layout of colors within an image. It is a three-dimensional histogram, indexed by color in two dimensions, and spatial distance in the third dimension, that records the frequency that pixels of two given colors occur at a given spatial separation. We define it formally for any region *J* of an image *I* as follows:

$$\text{if A} = \text{CCH}\{J, I\} \text{ then } A(c_1, c_2, d) =$$
$$\text{size}\left(\left\{ (p_1, p_2) \;\middle|\; \begin{matrix} p_1 \in J, p_2 \in I, c_1 = I(p_1), \\ c_2 = I(p_2), d = \text{qdist}(p_1, p_2) \end{matrix} \right\} \right)$$

for all $c_1, c_2 \in Q_c$ and non-negative integers $d \leq T_\text{d}$, where T_d is a constant specifying the neighborhood size (e.g., 16 pixels), *size* counts the number of elements of a set, and *qdist* is a quantized distance function. This CCH definition may seem unusual because only one pixel in the pair must belong to *J*. As explained in Section 2.5, this property turns out to better promote efficient computation of the CCH.

Any function that returns some measure of the distance between two pixels quantized to a non-negative integer may be used for *qdist*. We use a quantized Euclidean distance:

$$\text{qdist}((x_1, y_1), (x_2, y_2)) = \left\lfloor \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \right\rfloor$$

It is possible to use another distance metric or to quantize differently (e.g., non-uniformly). Note that because Euclidean distance is invariant to rotation, the CCH is also invariant to rotation. This is a very desirable feature for object detection. If city-block distance is used, more efficient CCH computation is possible [10], but the rotation invariant property is lost.

The CCH, as used in [9], suffers from a fundamental problem: solid regions of color tend to contribute a disproportionate amount of energy and overwhelm comparison metrics. This causes the CCH to show similarity between images with similar regions of solid color arranged in quite different spatial layouts. We solved this problem with a new construct called the *color edge co-occurrence histogram*. This histogram captures the distribution of separation between pairs of color edges, preventing solid color regions from dominating the histogram. Color transitions are perceptually very important; therefore, CECH better captures the spatial signature of an image region.

The CECH is defined as follows:

$$\text{if A} = \text{CECH}\{J, I\} \text{ then } A(c_1, c_2, d) =$$
$$\text{size}\left(\left\{ (p_1, p_2) \;\middle|\; \begin{matrix} p_1 \in \text{edges}(J), p_2 \in \text{edges}(I), c_1 = I(p_1), \\ c_2 = I(p_2), d = \text{qdist}(p_1, p_2) \end{matrix} \right\} \right)$$

where edges(*I*) is the set of pixels in *I* that either have at least one 8-neighbor of a different color or lie on the boundary of *I*. Another advantage of the CECH is that its computation is much more efficient than the CCH on a typical region. This is because image pixels that are not edges can be immediately ignored, saving T_d^2 operations per non-edge pixel. This represents substantial savings because the majority of pixels in a typical image are not edges.

2.3. Prescreening

A fast prescreening step is employed to eliminate image regions that obviously do not contain the target object in order to facilitate efficient search. The prescreening step is accomplished in the following way. Identify the set S_1 of colors occupying a significant percentage (e.g., >10%) of the area of quantized model image M_Q. Next, pass a window over the quantized search image, I_Q. We use a window of size $c \times c$ where c is one-tenth the length of the longer dimension of *I*. For each window centered at *(x,y)*, identify the set $S_{(x,y)}$ of the colors occupying a significant percentage (e.g., >10%) of the window area. A binary mask image, $P_{1,}$ is created that identifies pixels corresponding to possible object locations as 1 and background regions as 0:

$$P_1(x,y)=\begin{cases}1 \text{ if } \dfrac{\text{size}(S_{(x,y)}\cap S_1)}{\text{size}(S_1)}\geq T_A\\ 0 \text{ otherwise}\end{cases}$$

where T_A is some constant equal to, for example, 0.5.

Mask image P_1 removes image regions that do not have the correct colors to contain the target object. However, the local spatial layout of the colors has not been verified. Therefore, we compute a second mask image, P_2, which checks that the local spatial color arrangements are also consistent with the target object. P_2 is computed by randomly choosing pairs of pixels from I_Q no further than T_d pixels apart. For each pair (p_1,p_2) at distance *d*, the corresponding entry $(I_Q(p_1),I_Q(p_2),d)$ in the model CCH is checked, and the probability *p* of occurrence is computed. This probability is added to the two pixels in P_2, i.e., add *p* to $P_2(p_1)$ and $P_2(p_2)$. This process is repeated many times (e.g., 10 *mn* times, where $m \times n$ are the dimensions of *I*). The resulting image P_2 is like a probability map, representing the probability that each pixel is located in the target object. As a result of the random nature of this process, probabilities of individual pixels may be noisy. To reduce such noise, a mean filter is applied to P_2. P_2 is thresholded (e.g., at a fixed threshold of 0.1), and the final mask image *P* is computed by taking the logical AND of P_1 and P_2. A pixel is 1 in *P* only if it satisfies the local color requirements of P_1 and the spatial arrangement requirements of P_2.

Connected-component analysis is performed on *P*, and the minimum enclosing rectangles of all connected components are determined. These rectangles make up the set *R* of search regions that may contain the target object. Only these regions are further considered in the remaining steps of the algorithm. In addition, the prescreening provides a coarse range of the scale.

2.4. Object detection

Each of the search regions in set *R* identified during prescreening is examined separately. Suppose search region R_i is being examined. It is assumed that the model image *M* has dimensions $m_m \times n_m$ and that this represents the smallest possible size of the object in a search image. The largest scaling of the model *M* that fits inside R_i, while preserving its aspect ratio, is determined:

$$\gamma_H=\min\left(\frac{m_i}{m_m},\frac{n_i}{n_m}\right)$$

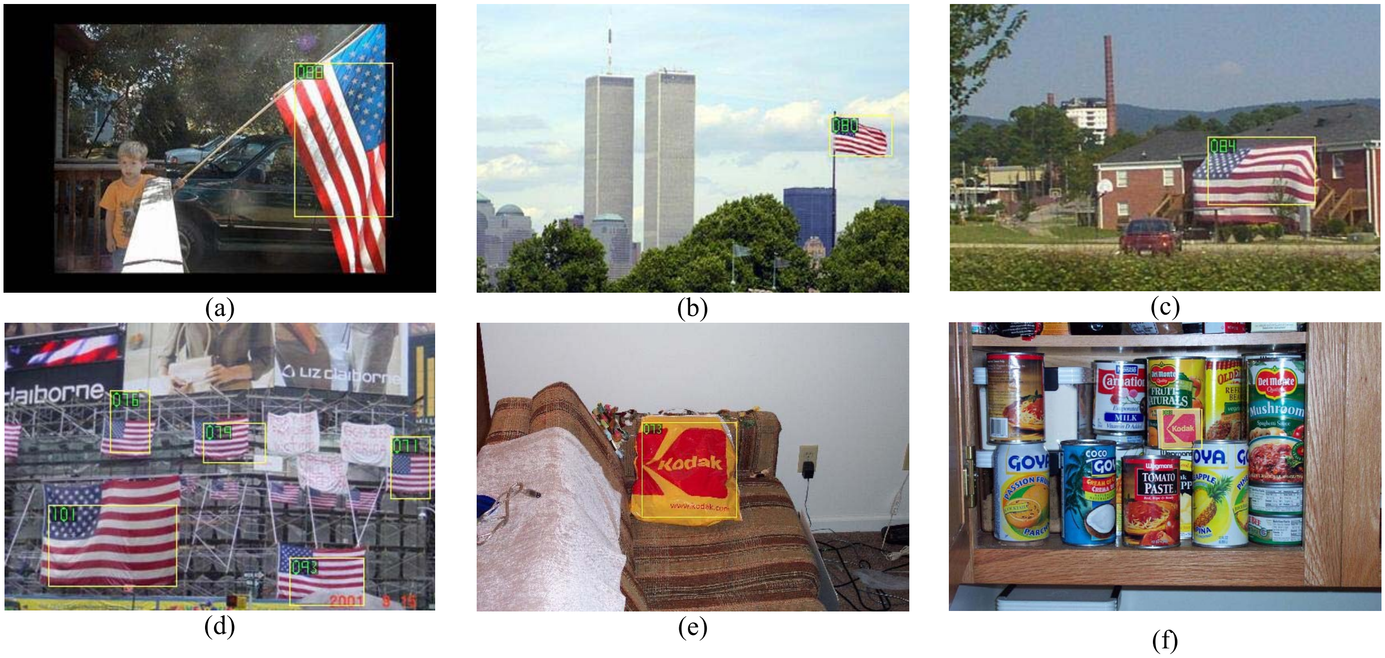
where $m_i \times n_i$ are the dimensions of R_i. We search for objects of arbitrary size by considering multiple scale factors between 1.0 and γ_H. Choosing equally spaced scale factors is one option, but proper size matching is more important at smaller sizes than at larger ones. Therefore, we choose our set of scale factors $\{\gamma_o, \gamma_1, \gamma_2, \dots \gamma_n\}$ such that $\gamma_o = 1$ and $\gamma_{j+1} = \alpha\gamma_j$, where $\alpha > 1.0$ is a constant (e.g., 1.1).

For each scaling factor γ_j, the pixel data in search region R_i is subsampled by γ_j. The resulting subsampled-image region is searched by considering search windows of size $\gamma_L m_m \times \gamma_L n_m$, at increments of Δx and Δy, where Δx and Δy are constants (e.g., equal to 10 pixels). In other words, the set of search window positions within subsampled R_i at scaling factor γ_j is:

$$P_{i,j}=\left\{(a\Delta x,b\Delta y)\,\middle|\, a,b\in Z^*, a\leq\frac{\gamma_j(m_i-m_m)}{\Delta x}, b\leq\frac{\gamma_j(n_i-n_m)}{\Delta y}\right\}$$

We require a meaningful similarity metric between a candidate rectangle and a model. Normalized L1 distance and histogram intersection [14] have been proposed in the past for comparing histograms, but both of these metrics have disadvantages. The L1 distance demands an exact bin-by-bin match between histograms and thus punishes areas that contain background in addition to the target object. Histogram intersection attempts to correct this but causes many false alarms. Histogram intersection, used in the original CCH-based algorithm [9], simply measures the presence of the model colors without verifying that they appear in the correct proportions; we found it to be prone to false positives in our study.

We employed a similarity measure that overcomes the above problems by emphasizing feature comparison along image edges and ensuring that colors are present in the same proportions. Our metric computes the similarity between a model *M* and an image region I_r in the following way. Let C_m be the CECH of *M* and C_r be the CECH of I_r. The mean



(a) (b) (c)

(d) (e) (f)

Figure 1: Sample detection results for U.S. flags in (a) through (d) and the Kodak company logo in (e) and (f).

bin height of C_m is subtracted from each of its bins, and the mean bin height of C_r is subtracted from each of its bins. Next, we define a similarity measure based on a bin-by-bin linear regression between C_m and C_r, i.e., find a scalar b in

$$C_r(c_1,c_2,d)=bC_m(c_1,c_2,d)+\varepsilon(c_1,c_2,d)$$

such that the residuals matrix ε is minimized according to a least-squares criterion. The best-fit value of b and the correlation coefficient cc are computed using the standard linear regression formulae [15].

A high-quality match between C_m and C_r causes a high correlation coefficient, indicating that the heights of the bins in each CECH are similar in proportion and a value of b close to 1.0, indicating that the size of I_r is close to the size of the target object. We define a similarity measure D_e as a combination of these two criteria:

$$D_e(C_m,C_r) = k_1 cc(C_m,C_r) + \max\left(0,(1-k_1)(1-\left|\log_2 b(C_m,C_r)\right|)\right)$$

where $0 < k_1 < 1.0$ is a constant (e.g., $k_1 = 0.9$). The logarithm ensures that a match is punished equally for being n times too large as it is for being n times too small.

We also must test that the same quantities of color are present in both image regions. To do this, we compute the color histogram of M and I_r, called CH_m and CH_r, respectively. A color-based distance D_c is, therefore defined as follows:

$$D_c(CH_m,CH_r)=k_2 cc(CH_m,CH_r)+ \max\left(0,(1-k_2)(1-\left|\log_2 b(CH_m,CH_r)\right|)\right)$$

where $0 < k_2 < 1.0$ is a constant (e.g., $k_2 = 0.9$). Then the overall similarity between two image regions is given by

$$D(M,I_r) = k_3\,\mathrm{D_e}(C_m,C_r) + (1-k_3)\,\mathrm{D_c}(\mathrm{CH}_m,\mathrm{CH}_r)$$

where $0 < k_3 < 1.0$ is a constant (e.g. $k_3 = 0.5$). Note that the parameters (k_1, k_2, k_3), while empirically chosen, are fixed.

To perform object detection, the similarity measure D is computed between each of the search windows in the sets $P_{i,j}$ and the model. The windows with the best scores at each scale factor are selected. Next, the search window with the best score for each search region R_i is selected among the best windows from each scale factor. Each of the selected search windows is declared to be the target object if its similarity score is above a threshold (e.g., 0.6).

If the model M has an aspect ratio close to 1.0 (i.e., circular or square model), rotation invariant searching is automatically achieved because of the internal rotation invariance of the CECH. In other cases, the searching procedure must be repeated for multiple model orientations. We have found that, in most cases, only two model orientations (horizontal and vertical) are needed, even when the target objects are aligned at other orientations.

2.5. Efficient searching

Computation of the CECH on an $n \times n$ pixel region with distance threshold T_d using a naïve algorithm is $O(n^2 T_d^2)$. A straightforward implementation of the search described in the previous section is, therefore, expensive. Inspired by similar work in the past, we have developed several techniques to drastically reduce computation time.

Note that the CECH is additive, that is, for three regions *A, B,* and *C* in image *I*:

$$\text{if } A \cup B = C \text{ and } A \cap B = 0 \text{ then}$$
$$\text{CECH}\{A,I\} + \text{CECH}\{B,I\} = \text{CECH}\{C,I\}$$

Second, note that, typically, Δx and Δy are set such that $\Delta x << m_m$ and $\Delta y << m_n$, causing a significant amount of overlap between adjacent search windows. This is exploited by storing the CECH of previously visited search windows and using them to compute the CECH of overlapping search windows at the same scale factor. For example, suppose the CECH of region B1 has been computed, and the CECH of an overlapping window B2 is needed. Suppose S1 is the set of pixels in B1 but not B2 and A1 is the set of pixels in B2 but not B1. The CECH of B2 can be computed as follows:

$$\text{CECH}\{B2,I\} = \text{CECH}\{B1,I\} - \text{CECH}\{S1,I\} + \text{CECH}\{A1,I\}$$

This computation is very fast because CECH{B1,I} is already known, and S1 and A1 are small image regions. Once the CECH of one search window has been determined, this technique can be applied recursively to efficiently compute the CECHs of all remaining search windows.

It is also possible to reuse information from the CECHs at lower scale factors of an image to speed computation of the CECHs at higher scale factors. Specifically, we note that for an image *I* subsampled at a scale factors γ_1 and $\gamma_2 = s\gamma_1$, where $s \in N$, if A = CECH{$I_{\gamma 1}$} and B = CECH{$I_{\gamma 2}$} then B(c_1,c_2,ds) = A(c_1,c_2,d). In other words, it is possible to recycle some of the bins from the CECH at a lower scale factor when computing the CECH with an even multiple of that scale factor. When s = 2, for example, roughly one-fourth of the cost of computing the CECH can be saved by exploiting this scale redundancy.

3. Experimental results

Our proposed detection algorithm has been tested on a variety of different compound color objects, including flags and logos. Figure 1 presents sample results of the algorithm applied to detecting the flag of the United States and the Kodak company logo. The model image size was 77 × 47 for the flag and 43 × 43 for the logo. In Figure 1, *a* through *d* show reasonable detection results for a variety of images, including flags with rotation and some degree of self-occlusion (image *a*), spatial distortion (image *b*), difficult illumination conditions and confusing flags (image *c*), and multiple flags per image (image *d*). Note that for two of the flags in *d*, the incorrect flag orientation was chosen. This occurs because the CECH is invariant to rotation and the flag is somewhat symmetric with respect to a 90º rotation. Postprocessing could be used to correct this if accurate object orientations are required by an application. Images *e* and *f* also show good results for the logo, despite the distortion from a flexible surface in *e* and the cluttered background in *f.*

The experiment with the American flag was conducted using a single flag model on 98 images uploaded by Internet users to the Tribute to American Spirit PhotoQuilt (http://www.kodak.com/go/photoquilt). This is a challenging dataset because of the wide variety of subject material, illumination, photographic composition skill, and camera type. All images were subsampled to a resolution of 256 × 384 before processing. We counted an accurate detection when the bounding box produced by the algorithm matched to within about 20% of the actual size of the flag (i.e., when the algorithm missed no more than 20% of the actual flag's area and included no more than 20% of the flag's area in the background). The results showed that 81.3% of the flags in the dataset were accurately detected. An additional 5.6% of the flags were detected but had a localization error greater than 20%. Depending on the application, accurate localization may not matter. For example, if the application is to search for images containing flags, inaccurate bounding box localization would not be a problem. The remaining 13.1% of the flags were not detected. Most of these correspond to extremely small flags (e.g., those in Figure 1d) or images with severe color problems. The false alarm rate was 1 per 6 images.

To validate the merits of the proposed algorithm, we conducted a comparison with the only closely related algorithm in [9]. Because that algorithm was designed with the assumption that the precise scale and orientation of the object is known, we actually provided such information to that algorithm in our experiments. Note that in doing so we were giving a major advantage to that algorithm. Without such information, it is conceivable that the algorithm would have to search for ranges of scales and orientations; there is no guarantee of reaching the precise scale and orientation because it is impractical to search all possible values.

The fROC (Free-response Receiver Operation Characteristic) curve of our algorithm is shown in Figure 2. Because the original CCH-based algorithm was designed to find exactly one object in any given image, its performance is characterized by a single operating point (vs. a full curve). Even with the handicap (knowing precise scale and orientation), the original CCH-based algorithm performed poorly because it was not insensitive to color variations and geometric deformations. At the same detection rate of 60%, our algorithm produced only one false positive vs. sixteen by the original CCH-based algorithm, for the entire test set. Note that there is no training set per se for either algorithm because each relies on only one model image. At the same false positive rate of roughly one per six images, the detection rate of our algorithm is 81% vs. 58% by the original CCH-based algorithm. Clearly, our algorithm outperformed by a decisive margin because of the four major components described in Sections 2.1 through 2.4.

Empirically, our algorithm is on the average 2x faster than the original CCH-based algorithm in [9] (for one object of known scale and orientation), even thought we search for multiple scales and orientations for potentially multiple instances of the object.

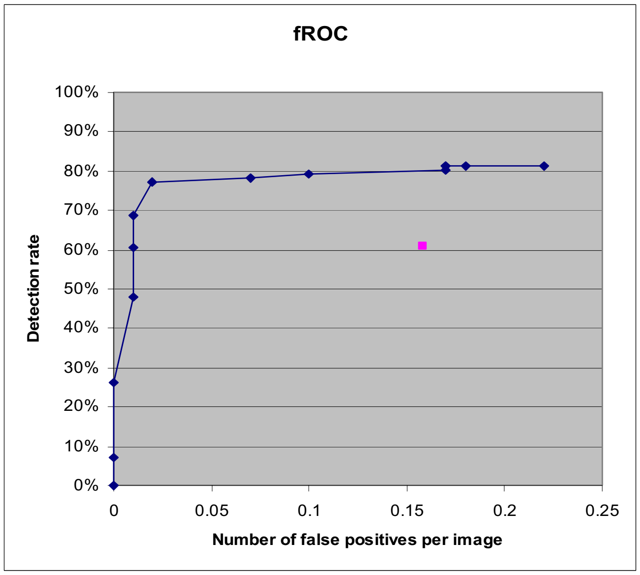


Figure 2: The fROC curve of our algorithm and comparison with the algorithm (one operating point) in [9].

4. Conclusions

This paper addresses the problem of detecting *different instances* of the *same type* of *compound color object* in unconstrained images. We proposed an algorithm based on color edge co-occurrence histograms and additional mechanisms for coping with color and geometric variations. Experiments demonstrated its efficacy on different compound color objects in an extremely challenging dataset. It represents a significant robustness improvement over the appealing but fragile original algorithm in [9], and the same negligible overhead for either machine learning (it often takes only a single model image) or human learning (it takes no object-specific rule or model crafting).

There are several opportunities for future work in this area. While our perceptually motivated color quantization produces good results on most images, some colors are difficult to quantize due to ambiguity. For example, a light pink color could be assigned to white or red, and the best choice may vary from image to image. A possible solution is to allow the quantization algorithm to assign *multiple* quantized colors to a single pixel, and build fuzzy CECHs that incorporate all of the possible quantized colors into the feature space.

While the CECH records gross spatial layout information, it is not capable of discerning subtle shape differences like images containing different lines of text. This is beneficial for handling object distortion but can cause false alarms. Future work could study alternative forms of spatial-color joint probability functions that may provide a richer set of spatial information while still offering robustness to object distortion.

Acknowledgements

The authors wish to thank Dr. Geoff Woolfe for his help with the color-naming algorithm, and www.kodak.com for permission to use their images.

References

[1] D. Forsyth and M. Fleck. "Body Plans", *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, 1997.

[2] R. Fergus, P. Perona, and A. Zisserman. "Object Class Recognition by Unsupervised Scale-Invariant Learning", *Proc. of IEEE Conf on Computer Vision and Pattern Recognition,* 2003.

[3] H. Rowley, S. Baluja, T. Kanade, "Rotation Invariant Neural Network-Based Face Detection", *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition,* 1998.

[4] H. Schneiderman and T. Kanade, "A Statistical Method for 3D Object Detection Applied to Faces and Cars", *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, 2000.

[5] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, T. Poggio, "Pedestrian Detection Using Wavelet Templates", *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition,* 1997.

[6] A. Selinger, R. C. Nelson, "Appearance-based Object Recognition Using Multiple Views", *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition,* 2001.

[7] D. P. Huttenlocher, G. A. Klanderman, and W. J. Ricklidge, "Comparing Images Using the Hausdorff Distance", *IEEE Transactions on Pattern Analysis and Machine Intelligence,* (15), 1993, pp. 850-863.

[8] T.F. Cootes, G.J. Edwards, and C.J. Taylor, "Active appearance models", *Proc. of European Conference on Computer Vision,* 1998.

[9] P. Chang and J. Krumm, "Object Recognition with Color Cooccurrence Histograms", *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition,* 1999.

[10] J. Huang, S. Kumar, M. Mitra, W.-J. Zhu, R. Zabih, "Image Indexing Using Color Correlograms", *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, 1997.

[11] Q. Iqbal and J. Aggarwal, "Combining Structure, Color, and Texture for Image Retrieval: A Performance Evaluation", *Proc. of Intl. Conf. on Pattern Recognition,* 2002.

[12] K. Kelly and D. Judd, *Color Universal Language and Dictionary of Names.* National Bureau of Standards Publication 440, U.S. Government Printing Office, Washington, DC, 1976.

[13] E. Giorgianni and T. Madden, *Digital Color Management: Encoding Solutions,* Addison-Wesley, Reading, MA, 1997.

[14] M. Swain and D. Ballard, "Color Indexing", *International Journal of Computer Vision,* (7) 1, 1991, pp. 11-32.

[15] G. Box, W. Hunter and J. Hunter, *Statistics for Experimenters.* John Wiley & Sons, New York, 1978.