Identifying Color in Motion in Video Sensors

Gang Wui[1] Amir Rahimi[1] Kingshy Goh[2] Tomy Tsai[3] Ankur Jain[3]
Yi Wu[4] Edward Y. Chang[1] Yuan-Fang Wang[3]

[1]Electrical and Computer Engineering
UC Santa Barbara, CA 93106
{gwu,amir,echang}@ece.ucsb.edu

[2]Proximex Inc.
Cupertino, CA 95014
kingshy.goh@proximex.com

[3]Computer Science
UC Santa Barbara, CA 93106
{tomyt,ankurj,yfwang}@cs.ucsb.edu

[4]Intel Inc.
Santa Clara, CA 95054
yi.y.wu@intel.com

Abstract

Identifying or matching the surface color of a moving object in surveillance video is critical for achieving reliable object-tracking and searching. Traditional color models provide little help, since the surface of an object is usually not flat, the object's motion can alter the surface's orientation, and the lighting conditions can vary when the object moves. To tackle this research problem, we conduct extensive data mining on video clips collected under various lighting conditions and distances from several video-cameras. We observe how each of the eleven culture colors *can drift in the color space when an object's surface is in motion. In the color space, we then learn the drift pattern of each culture color for classifying unseen surface colors. Finally, we devise a distance function taking color drift into consideration to perform color identification and matching. Empirical studies show our approach to be very promising: achieving over* 95% *color-prediction accuracy.*

1 Introduction

In addition to shape, texture, and some other geometric properties, color is an important cue for describing the surface of an object. Being able to precisely name colors in video sequences can help tasks of object tracking and object search. For instance, one important application is to track a suspicious person, based on the clothing colors, across the fields-of-view (FOV) of multiple cameras. Another application is to search a person with specific soft-biometric characteristics (height, build, skin tone, and clothing colors) in a camera network. The color cue plays an important role in enhancing the reliability of these tracking and search tasks.

Color constancy is the ability of a vision system to accurately describe the color of an object in spite of variations in source illumination and receiver characteristics [1]. Despite decades of research in color-constancy algorithms (a survey is presented in Section 1.1), these algorithms cannot be used to reliably identify color in motion. First, most algorithms assume scene illumination to be uniform in the region of interest [14], or changing gradually [5, 20]. Such an assumption almost always fails in a surveillance scenario. Second, most color-constancy algorithms depend on reliable estimates of parameters such as angles between light sources and the object, reflection angles, and surface materials. These parameters can be unknown or difficult to estimate in real-time when the object being observed is in motion.

Instead of taking the route to model variations in surface orientation, extended light, secondary reflection, and varying color sensitivity of cameras, we employ a statistical learning approach. Our statistical approach samples pixel colors in various conditions, and constructs color classifiers at three levels: pixel, frame, and sequence. Even if a color has not been reliably identified at the pixel level, we show that the accuracy in color identification improves markedly at the frame level, and significantly at the sequence level. Our method essentially trains a *color-drift table* for a camera, and then uses statistical redundancy to mask noise caused by the environmental factors. One important finding of our empirical studies is that even when the color-drift table is trained in an environment different from the actual surveillance environment, the color-prediction accuracy at the video-sequence level can still be very promising (exceeding 95%).

1.1 Related Work

The goal of color constancy is to reduce color variation from fluctuation in source illumination and receiver characteristics. Both source illumination and camera sensitivity can be modeled as continuous functions of wavelength. Let us ignore atmospheric attenuation and scattering, which does not play a significant role in color appearance. The critical elements are then 1) light sources, 2) sensors, and 3) how an object interacts with the incident light. The interaction is often characterized as the ratio of the reflected light and the incident light, which is commonly referred to as the bi-directional reflectance function, or the BDRF. The BDRF depends on many factors, the most important ones being the geometric configuration (i.e., the surface orientation relative to the viewer and the light source) and the wavelength. The BDRF can be characterized by five parameters $(\theta_i, \phi_i, \theta_r, \phi_r, \lambda)$, where λ denotes the wavelength, and θ and ϕ are the azimuth and elevation angles of the incident (i) and reflection (r) directions, respectively, measured in an object-centered reference frame.

Measuring a BDRF is a tedious and daunting task. Although some such data have become available for a variety of surfaces [6], certain reasonable simplifications are often made. The most common assumption is that surfaces are isotropic, or that the BDRF does not change significantly if a surface is rotated about its normal. In this case the BDRF can be simplified into a function that depends on three angles and the wavelength: the incident angle, the reflection angle, and the phase angle (the angle between the incident and the reflection directions).

Another complication is that there exist two major reflection mechanisms [1]: interface reflection and body reflection.[1] Interface reflection occurs at the junction between an object and the surrounding medium. For metallic objects, interface reflection carries the distinct metallic color that is characteristic of a certain metal. For most non-metallic materials, interface reflection is minimally wavelength-dependent but highly direction-preferential. Thus, the reflection has the same spectra as the incident light and concentrates in a direction where the incident and reflection angles are the same and equal to half the phase angle. In contrast, body reflection is usually considered Lambertian and wavelength dependent. Most color constancy algorithms concentrate on analyzing and modeling the body reflection component as it carries the most discriminative information for inferring an object's true color.[2]

Algorithms for color constancy have been available for a couple of decades. The simplest model that accounts for illumination variation is to compute a single statistic, such as a mean, to estimate scene illumination, which is assumed to be uniform in the region of interest. This leads to the so-called greyworld algorithms [14]. Retinex methods separate the illumination effect, which is assumed to change gradually over an image, from the surface reflection effect, which makes drastic and unpredictable jumps across boundaries of objects. In these methods, varying illumination is discounted by discarding small and slow-varying scene changes (e.g., using derivative and thresholding operations).

Linear decomposition [10, 13, 21] methods model illumination change using a linear transformation. This model is justified if illumination and surface reflectance can be expressed as linear combinations of a small number of basis functions.[3] In particular, the diagonal linear model, which maps the image taken under one illumination to another by simply scaling each color channel independently, has been shown to be effective in some application scenarios [2, 3].

Gamut mapping [9, 12] finds all possible variations of the image of a scene under an unknown illumination mapped to the image of the same scene under a known, canonical illumination. It has been shown [12] that a gamut, which is all possible illumination responses due to all known or expected surface reflectances, is a convex set. Two such gamuts, one under a known, canonical illumination and the other under an unknown illumination, are both convex, and they are related by a diagonal transformation (assuming a diagonal linear model). The mapping solution is then obtained by working with the convex hull of the measured gamut from the image of the unknown illumination and finding all feasible transforms that map all vertices of the convex hull into the canonical gamut.

Bayesian correlation [4] applies Bayes' rule for inferring intrinsic object colors from sensed pixel colors. It is assumed that the probability of occurrence of scene illumination and surface reflectance is known. Furthermore, each illumination and surface reflectance combination leads to a sensed color response that is predictable by a suitable mathematic model. Then given that sensed color, we can select the illumination and surface reflectance combination that has the largest prior, and predict most faithfully the sensed color. Many other color constancy models exist (e.g., [18, 19, 28, 29, 15]), and [2, 3] present a comprehensive survey and comparison of some popular ones.

More recently, [26, 16] proposed methods to map sensed colors in one camera to those in another. Such mapping is computed between every pair of cameras. While it was proven in [16] that the space of the mapping functions (or

[1] Many other models also exist, such as the Torrance-Sparrow model widely used in computer graphics [11] and [24] that separates three reflection components: body reflection, specular lobe, and specular spike.

[2] While many algorithms exist that exploit the specular interface reflection for color constancy and other analysis (e.g., [8, 17, 18, 24, 23]), we do not discuss them here because they are less relevant to the proposed techniques.

[3] [7] found that for a daylight data set taken at a single location, three basis functions account for 99% of the variance. The assumption for surface reflectance has also been extensively validated [10, 13, 21, 12, 9].

the brightness transfer functions BTF) is generally of a low dimensionality and can be estimated using standard basis expansion techniques, such a model is valid only for planar surfaces under uniform lighting.

In summary, color perception is an extremely complicated and nonlinear science. To simplify the analysis, many color constancy models assume a single camera; a fixed, frontal surface orientation; and oftentimes, a point light source or spatially-invariant illumination. Or, color-constancy research is often confined to the Mondrian world—a world of flat, frontally presented collages of colored papers. In the real world, we must account for spatially-distributed surveillance cameras operating under different lighting conditions and with varying color sensitivity. To register color accurately, a scheme must take into consideration variations in surface orientation, extended light, secondary reflection, limited spatial resolution, and varying color sensitivity of multiple cameras. The complexity of such modeling for identifying color in motion makes the task daunting, if not impossible. In this work, we use a statistical learning approach, which learns to identify a color through training. In an analogical way, traditional color-constancy algorithms employ a generative approach to derive a function to predict color, whereas our approach is discriminant: we perform color prediction directly from the knowledge learned from data.

1.2 Contribution Summary

Our work makes the following contributions:

1. We conduct extensive data mining to learn how individual colors can drift in different conditions. Based on the drift patterns, we train an ensemble classifier to identify the color of an object in motion.

2. We use the drift patterns (depicted by a *color-drift table*) to better compute the "distance" between two color histograms. For two colors that can drift into each other, their distance is reduced; otherwise, the distance is magnified.

3. Empirical studies show that a color-drift table calibrated in one environment for a camera could be used in a different environment by that camera and still achieve high color-prediction accuracy: over 95% at the video-sequence level. This suggests that camera vendors can provide a default color-drift table to achieve a good baseline performance. One can perform environment-specific tuning on the table to further improve prediction accuracy by a couple of percentage, if desired.

2 Algorithm

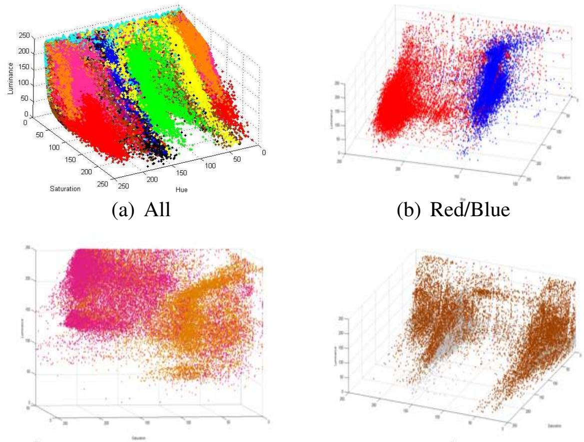
We propose a robust color calibration procedure as follows: We quantize the entire color space into eleven bins (black, white, red, yellow, green, blue, brown, purple, pink, orange, and grey). These colors are usually referred to as culture colors [22], which have been used in literatures of different cultures in the past two-thousand years to refer to colors. Representing the entire color space using a small number of primitives is advantageous for at least two reasons: (1) In most surveillance applications, surveillance subjects occupy small screen areas, and hence, pixels available to construct the color signature of an object are usually quite limited. Coarse quantization of the color space (into 11 bins in the case of culture colors) avoids random fluctuation of color signatures due to insufficient pixel samples, and (2) culture colors facilitate posing queries since these colors are universally perceived and widely used across various cultures. One might argue that having a finer quantization may better discern different objects, e.g., telling a light blue car from a dark blue car. Unfortunately, finer quantization leads to less reliable color prediction, and can be counter-productive in improving prediction accuracy. (A blue car in the shadow can appear to be dark blue and in the sun light blue.) In the situation where more than one object has the same culture color, we can rely on other visual cues (e.g., size, shape, z-y-z coordinate, etc.) or further analyze the color spectrums to tell the objects apart. It should also be noted while we adopt culture colors for their universal appeal; our algorithm is equally applicable to other color quantization schemes.

Instead of taking a "generative" approach discussed in Section 1.1, we employ a discriminant approach to address the complexity of the color-calibration problem. For each sensor, we collect images of calibration markers that are known to be of certain culture colors. (These can be as simple as people wearing certain colored shirts walking around in the field-of-view of the camera.) The sensed pixel values are recorded in a table ${C_{culture}}_{i,k}^{h,s,l}$, where $1 \leq i \leq 11$, and $1 \leq k \leq n_i$, and n_i is the number of color samples collected for the i-th culture color. We train for each culture color a Support Vector Machine (SVM) classifier f_i

$$f_i(C^{h,s,l}) = \sum_{k=1}^{n_i} \alpha_k y_i K(C^{h,s,l}, {C_{culture}}_{i,k}^{h,s,l}). \quad (1)$$

where K is a suitable kernel function (e.g., Gaussian), and y_k is one if the pixel color is C_i, or otherwise zero. A query color sample $C^{h,s,l}$ is then assigned to the culture color with the highest SVM score.

Our color-prediction algorithm consists of three steps: mining, training, and classification. We first observe how a given color can drift under different conditions through extensive data mining. Once we have identified the region of drift in the color space for each color C_i, the training step learns from the data a set of classifiers f_i for C_i, $i = 1 \cdots C$, to predict color membership of a pixel. Finally, the classification step works at three levels—pixel level, frame level, and sequence level—to identify the color of an object.



(a) All (b) Red/Blue

(c) Pink/Orange (d) Brown/Grey

Figure 1. HSL Distribution.

2.1 Mining Testbed

To conduct color calibration and testing, we collected three sets of videos. The first set was taken simultaneously from five different cameras in a room of windows (with draperies to allow or block sun light) under three distances from the cameras, in three lighting conditions (fluorescent lights only, incandescents only, fluorescent lights plus sun light). The second set collected 165 video sequences taken at a hallway by three different cameras simultaneously, in five lighting conditions (day-time or night-time, with or without incandescents and fluorescent lights). And the third set was provided by our partner at a major US airport. We used the first set to produce a *color-drift table* for each camera, and the other two sets for testing (testing results are presented in Section 3). Notice that the three sets of data were collected in different environmental settings.

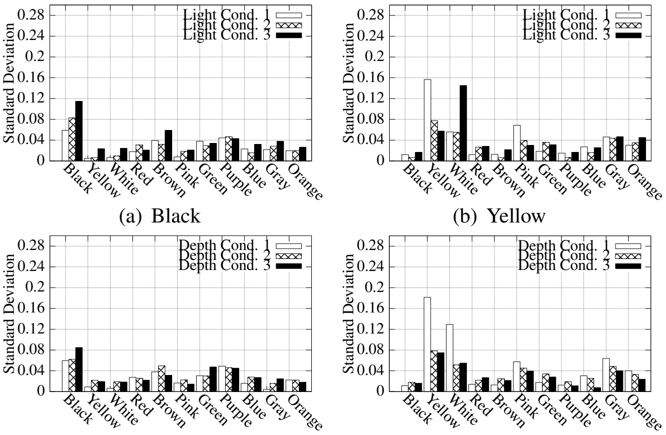
People wearing eleven different single-culture-color shirts walked in the field-of-view of the cameras. Of the same culture color, the shirt color used in the calibration clips is slightly different vs. the shirt color used in the testing clips (e.g., dark blue vs. light blue). This setting allows us to test the robustness of the calibration process. We relied on an object-tracker that we implemented to locate and track the target objects. For locating the area of a shirt, we extracted the pixels beneath the head in the bounding box, assuming that those belonged to a shirt. During the training phase, we removed noisy pixels (pixels that are more than one σ outside of the HSL means) so that we could establish a reasonably clean set of ground truth for each color. In the testing phase, we expected that our statistical prediction model in three levels would tolerate the tracker's noise.

2.2 Training Ensemble Classifier

We plotted for each culture color the pixel distribution in the HSL space. Ideally, each color should appear as one point in the color space. However, with the interferences of various environmental factors, the pixel colors of a culture color spread out like a cloud. Figure 1(a) plots the eleven color clouds, one for each culture color, in the HSL space. Although individual color clouds cannot be perfectly separated in the space from the others, most clouds do not exhibit severe overlaps. Figures 1(b) and (c) show that the red/blue pair and the pink/orange pair are both well separated. Figure 1(d), however, shows that the brown/grey pair is intermixed in the color space. (Each 3D plot was rotated to an angle where the best separation between colors could be viewed.) This figure can also be used to justify the use of one shirt to represent a culture color. A single color in different lighting conditions and distances mushrooms into a cloud. As long as one conducts calibration in various environments, one can obtain a similar cloud for a color. (We do not care about the density, but only about the spread.) In other words, varying environmental factors is equivalent to varying a red shirt. Slight differences can be taken care of by statistical redundancy. To validate, we also conducted color calibration using a color board where each culture color is represented by three samples of different HSL values. The cloud formed by the one-shirt sample is similar to that formed by three samples.

Figure 2 reports how colors drift under different lighting conditions and distances. Because of the space limitation, we only report the drift patterns of *black* and *yellow*. *Black* is representative a group of poorly behaved colors (*black*, *green*, *purple*, and *grey*), which exhibit substantial drifts. *Yellow* and the rest of the colors do not drift too much. The first row of Figure 2 shows the drift patterns of *black* and *yellow*, respectively, under different lighting conditions. The second row shows their drift patterns in different distances (depths). The figure exhibits two useful patterns. First, in different lighting conditions, the drift patterns are similar. Likewise, the drift patterns in different distances are about the same. Second, the drift pattern under lighting changes and under distance changes are very similar. Therefore, we can summarize the drift patterns into a *drift table*. Notice that the magnitudes of drifts of a color under different conditions may be different. To prevent the drift table from overfitting the calibration environment, we only account for the other colors that a particular color could drift into.

Table 1 summarizes the possible shades into which a culture color can drift (after thresholding noise). For instance, black can drift into brown, green, purple, blue, or grey, depending on the location/orientation of a person (wearing the shirt) and the lighting condition. This *drift table* can be useful when we conduct color prediction. For example, given the reading of a pixel being black, using the first column of the table we can predict with high confidence that it does not come from a surface of white, red, yellow, pink, or or-



(a) Black (b) Yellow

(c) Black (d) Yellow

Figure 2. Drift Patterns of Two Representative Colors.

	bk	wt	rd	bu	yl	gn	pp	pk	og	bn	gy
bk	o			x		x	x			x	x
wt		o			x						x
rd			o					x	x	x	
bu	x			o			x				
yl		x			o						x
gn	x			x		o	x			x	x
pp	x		x	x		x	o			x	x
pk			x					o	x		
og			x					x	o		
bn	x		x				x			o	
gy	x	x			x	x					o

Table 1. Color Drifts. bl (black), wt (white), rd (red), bu (blue), yl (yellow), gn (green), pp (purple), pk (pink), og (orange), bn (brown), and gy (grey).

ange. If some pixel readings are blue, we narrow the candidate colors down to the intersection of the black and blue columns: black, blue, green or purple. After we have further considered the prediction confidence of each pixel (a pixel color closer to the middle of a color cloud in Figure 1 enjoys a higher confidence score), we can usually pin down the color. We will say more about the usefulness of this table when we discuss constructing a frame-level classifier.

Figure 1 reveals that the boundaries between color clouds are nonlinear, and that the boundaries between colors can be fuzzy. In addition, despite the large number of training clips that we have collected, the sample culture colors do not cover the entire color space. Therefore, the base-classifier that we choose for classifying colors must be able to model a soft, nonlinear class boundary, and must be able to tolerate noise.

To address these three issues, we propose a three-level ensemble classifier, which uses soft-margin Support Vector Machines (SVMs) [30] as the base learning algorithm. At the pixel level, we classify each pixel of the object of interest. At the frame level, we aggregate the pixel-level predictions to form a frame-level prediction for the object of interest. At the sequence level, we match temporal patterns of drift to further strengthen color-prediction accuracy.

Pixel level

The number of pixels collected for each culture color is in the order of millions. Training an SVM classifier with such a large number of training instances can be time-consuming[4]. To conserve time, we can employ a *sampling* approach to select a small fraction of the pixels as the training data.

To label a query pixel with one out of eleven possible colors, we construct eleven one-per-class (OPC) SVM classifiers. Let $\mathbf{x}$ denote a query instance (a pixel of unknown color). Let $P(C_i|\mathbf{x})$ denote the probability estimate that $\mathbf{x}$ belongs to class C_i. The class of $\mathbf{x}$ is predicted as

$$\omega = argmax_{1 \leq C_i \leq M}\, P(C_i|\mathbf{x}). \qquad (2)$$

The SVM output of $C_i = f_i(\mathbf{x})$ is an uncalibrated value, and it might not translate directly to a probability value useful for estimating confidence. To estimate $P(C_i|\mathbf{x})$, Platt [25] suggests using a parametric model to fit the posterior $P(C_i|\mathbf{x})$ directly without having to estimate the conditional density $p(f|C_i)$ for each C_i value:

$$P(C_i = 1|\mathbf{x}) = \frac{1}{1 + exp(A \times f(\mathbf{x}) + B)}. \qquad (3)$$

This model assumes that the SVM outputs are proportional to the *log* odds of a positive example. The parameters A and B of Equation 3 are fitted using maximum likelihood estimation from a training set. More precisely, A and B are obtained by minimizing the negative log likelihood of the sigmoid training data using a model-trust minimization algorithm.

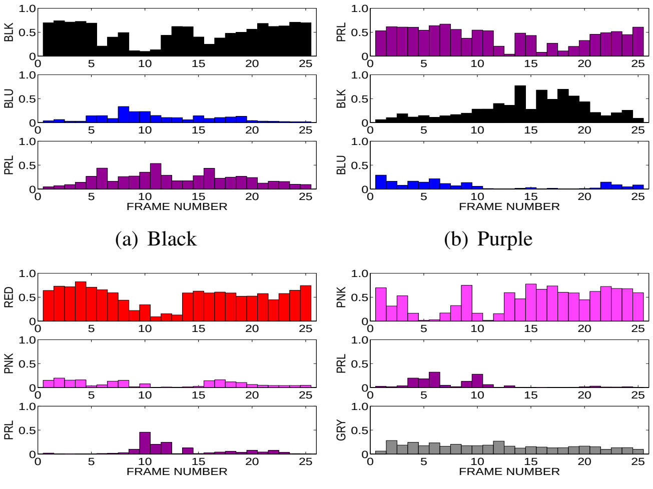
Frame level

At the frame level, we can employ two approaches. The first approach is to aggregate the pixel-level decisions for a frame. A naive way to perform such aggregation is to conduct majority voting. One major shortcoming of this approach is that the true color might not appear as the dominant color. (A purple shirt backing the light may be dominated by black.) To remedy this problem, we use the *drift table* (Table 2.2) to weight pixel predictions. Suppose at the frame level, we have a pixel-color set of {black, blue, red}. The *drift table* can tell us that the intersection of the three color columns yields only purple. If we still have more than one candidate colors, we can either vote for the color with

[4] The computational intensity of the fastest SVM solver is $O(n^{2.3})$, where n denotes the number of pixels.

the highest aggregated probability[5], or defer the decision to the sequence-level classifier, which aggregates frame-level predictions.

The second approach bypasses the pixel-level classifier. For each frame, we generate an eleven-bin color histogram. The color histogram is a vector of eleven features each representing the percentage a culture color appearing on the target object in that frame. This approach takes advantage of the *drift table* in a different way to compute the distance between two color histograms. More specifically, when a distance function like *earth-mover* [27] is employed, we can reduce the distance between two colors that can drift into each other, and penalize the distance between two that cannot drift into each other.



(a) Black

(b) Purple

(c) Red

(d) Pink

Figure 3. The Drift Patterns of Color Sequences.

Sequence level

At the top of the ensemble is the sequence-level classifier, which aggregates the predictions at the frame level. The motivation of the sequence-level classifier can be explained via Figure 3. The figure shows four representative color patterns each in a sampled frame sequence. For instance, Figure 3(a) shows a 25 sampled frame-sequence of a black shirt. Some black pixels drift into either blue or purple. Occasionally, purple becomes the dominant color in the sequence (frames 9, 10, 11, and 16). We can observe very similar drifts in the purple sequence. The purple sequence in Figure 3(c) can occasionally be dominated by black. Fortunately, we can identify for each color when we have a sequence of frames. In addition, a color drifts into black typically because of poor lighting conditions. These observations help us arrive at two heuristics for performing highly accurate sequence-level color-prediction.

Temporal rule. A color prediction is made only after k frames have been observed. The final color decision is a simple majority vote of the frames in the temporal window.

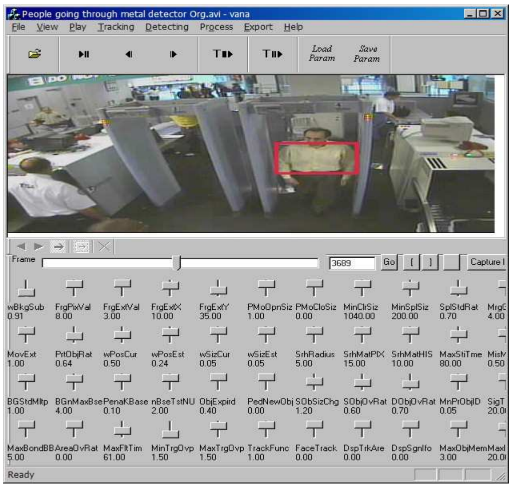
Spatial rule. Poor lighting conditions often cause a color to seem dark, and sunlight can cause a color to seem substantially brighter. Therefore, at certain spatial locations, we can discount the frame-level prediction. The near or far information about an object can be inferred by the size of a bounding box provided by the tracker. The spatial rule filters out the frames in which the bounding box indicates that a reliable color reading cannot be obtained. The spatial rule can be conveyed to the classifier via the tracker.

3 Experimental Evaluation

Our experiments were designed to answer two questions:

1. Can color be reliably predicted at the pixel level, the frame level, and the sequence level?
2. Can a color-drift table trained in one environment be utilized in a different environment to reliably perform color prediction?

We have discussed in Section 2 the construction of our training data and the training process. For testing, we obtained two datasets prepared by our partner corporation, which has been working on surveillance projects with major US airports. The data of the first testbed was collected in an office environment with three surveillance cameras monitoring the hallway. The second testbed consists of video clips taken at a major US airport. Figure 4 shows a passenger passing through a metal security gate, while the person's shirt color is being read and registered.



Frame 3689

wBkgSub 0.91	FrgPixVal 8.00	FrgExtVal 3.00	FrgExtX 10.00	FrgExtY 35.00	PMoOpnSiz 1.00	PMoCloSiz 0.00	MinClrSiz 1040.00	MinSplSiz 200.00	SplStdRat 0.70	MrgC 4.00
MovExt 1.00	PrtObjRat 0.64	wPosCur 0.50	wPosEst 0.24	wSizCur 0.05	wSizEst 0.05	SrhRadius 5.00	SrhMatPIX 15.00	SrhMatHIS 10.00	MaxStiTme 80.00	MisM 0.50
BGStdMltp 1.00	BGnMaxBse 4.00	PenaKBase 0.10	nBseTstNU 2.00	ObjExpird 0.40	PedNewObj 0.00	SObSizChg 1.20	SObjOvRat 0.60	DObjOvRat 0.70	MnPrObjID 0.05	SigT 20.0
MaxBondBB 5.00	AreaOvRat 0.00	MaxFltTim 61.00	MinTrgOvp 1.50	MaxTrgOvp 1.50	TrackFunc 1.00	FaceTrack 0.00	DspTrkAre 0.00	DspSgnIfo 0.00	MaxObjMem 3.00	MaxI 20.0

Figure 4. Airport Metal Gate.

[5] Notice that considering probability before and after using the drift table can yield different predictions. In this example, purple cannot be recovered without using the drift table.

3.1 Pixel Level Evaluation

The accuracy at the pixel level suffers from a dismal 37.7% average error. This is hardly a surprise considering the information presented in our various figures and tables.

Because of the dismal prediction accuracy at the pixel level, and because of the long SVM running time on the one-million-pixel training data (a classifier may need to be retrained when additional training data become available), we decided that the pixel level approach was unattractive.

3.2 Frame Level Evaluation

For each frame, we generated an eleven-bin color histogram denoted as $P = \{(p_1, w_{p_1}), \cdots, (p_{11}, w_{p_{11}})\}$, where p_i represents one culture color and w_{p_i} represents the percentage this culture color appearing on the target object in that frame.

Let P and Q be two histograms: $P = \{(p_1, w_{p_1}),\cdots,(p_{11}, w_{p_{11}})\}$ and $Q = \{(q_1, w_{q_1}),\cdots,(q_{11},w_{q_{11}})\}$. We compared two methods for accounting frame difference. The first method used the L1 distance. The second method used the earth mover distance (EMD). EMD measures the amount of work necessary to transform one histogram to the other. Let f_{ij} denote the flow between feature p_i of P and q_j of Q. Let $D = [d_{ij}]$ be the ground distance matrix where d_{ij} is the ground distance between p_i and q_j. The objective function for computing the minimal work is expressed as

$$WORK(P, Q, F) = \sum_{i=1}^{11} \sum_{j=1}^{11} d_{ij} f_{ij}, \qquad (4)$$

subject to a set of constrains. Once the minimal transportation problem is solved, earth-mover distance between P and Q is calculated as

$$EMD(P, Q) = \frac{\sum_{i=1}^{11} \sum_{j=1}^{11} d_{ij} f_{ij}}{\sum_{i=1}^{11} \sum_{j=1}^{11} f_{ij}} \qquad (5)$$

We employed the *color-drift table* as the *ground distance matrix* in EMD. If color i and color j could drift to each other, we assigned $d_{ij} = 1 - (t_{ij} + t_{ji})/2$, where t_{ij} is the percentage of color i that drifts to color j. If two could not drift to each other, $d_{ij} = 1.0$. Finally, we assigned $d_{ii} = 0$ because any color could (of course) drift to itself. Once we obtained the pair-wise distances of all the frames, we converted it to a kernel matrix and trained frame-level classifiers.

Table 2 reports the classification performance of using L1 distance and EMD. Using EMD to reduce distance between colors that can drift into each other and penalize the distance between two that cannot drift into each other, we can improve classification accuracy by about two percentile points.

	EMD	L1
Culture Color	Mean(Variance)	Mean(Variance)
RED	0.0928(0.0242)	0.1078(0.0238)
BLUE	0.0486(0.0134)	0.0621(0.0200)
BLACK	0.2500(0.0291)	0.2544(0.0312)
GREEN	0.0094(0.0083)	0.0056(0.0068)
PINK	0.1033(0.0223)	0.1028(0.0212)
YELLOW	0.1105(0.0199)	0.1447(0.0225)
ORANGE	0.1747(0.0249)	0.1260(0.0201)
BROWN	0.2614(0.0356)	0.2786(0.0330)
PURPLE	0.2856(0.0316)	0.2488(0.0304)
WHITE	0.0670(0.0209)	0.1260(0.0330)
GREY	0.5033(0.0329)	0.5161(0.0344)
Total	0.1792(0.0070)	0.1841(0.0065)

Table 2. Frame Level Misprediction.

Culture Color	Different Environment	Same Environment
RED	0.0000(0.0242)	0.0000(0.0000)
BLUE	0.0000(0.0134)	0.0000(0.0000)
BLACK	0.0200(0.0291)	0.0000(0.0000)
GREEN	0.0000(0.0083)	0.0000(0.0000)
PINK	0.0000(0.0223)	0.0000(0.0000)
YELLOW	0.0000(0.0199)	0.0000(0.0000)
ORANGE	0.0000(0.0249)	0.0000(0.0000)
BROWN	0.0200(0.0356)	0.0500(0.0632)
PURPLE	0.0300(0.0316)	0.0400(0.0569)
WHITE	0.0000(0.0209)	0.0000(0.0000)
GREY	0.4100(0.0329)	0.2800(0.1303)
Total	0.0436(0.0368)	0.0336(0.0228)

Table 3. Sequence Level Misprediction.

3.3 Sequence Level Evaluation

At the sequence level, the prediction accuracy looks very promising. The first column of Table 3 shows that after we have employed our temporal and spatial heuristics to intelligently sample the more reliable frames in video sequences, we can achieve an average accuracy of 95.6%. The only color that still cannot be reliably predicted is grey. Grey can turn into white under bright light and into other colors near colored objects. It is encouraging to see that the table calibrated in the training setting can be successfully generalized to unseen settings.

Notice that this far we have presented the results of calibrating the drift table using the training data and using the table on the testing data for color prediction. Our final experiment was set up to use testing data to calibrate the drift table, and then apply the table to a set-aside subset of the testing data. In other words, the testing environment is the same as the training environment. The second column of Table 3 shows that the prediction accuracy at the sequence level reached 97%. This result indicates that environment-specific calibration can be helpful to improve a small fraction of color-prediction accuracy. Camera vendors can provide a default table to achieve a good baseline performance, and one can further perform environment-specific calibration to improve (though slightly) color-prediction accuracy.

4 Conclusion

In this paper, we have presented a statistical approach to tackle a very difficult problem in video surveillance—identifying a color in motion. As illustrated in our examples and experimental results throughout the paper, a color can drift into other colors because of many factors. Any attempt to predict a color at the pixel level is virtually futile . Traditional color models cannot be effective for identifying color in motion simply because the many changing parameters make most parameter assumptions invalid. Instead, through a statistical approach, we showed that color-prediction can be done quite reliably at the frame level, and very reliably at the sequence level.

References

[1] K. Barnard. *Modeling Scene Illumination Color for Computer Vision and Image Reproduction: A Survey of Computational Approaches*. PhD thesis, Ph.D. thesis, Simon Fraser University, 2002.

[2] K. Barnard, B. Funt, and V. Cardei. A Comparison of Computational Color Constancy Algorithms; Part One: Methodology and Experiments with Synthesized Data. *IEEE Transactions in Image Processing*, 11:972–984, 2002.

[3] K. Barnard, L. Martin, A. Coath, and B. Funt. A Comparison of Computational Color Constancy Algorithms; Part Two: Experiments with Image Data. *IEEE Transactions in Image Processing*, 11:985–996, 2002.

[4] D. Brainard and W. T. Freeman. Bayesian Color Constancy. *J. Opt. Soc. Am. A*, 14:1393–1411, 1997.

[5] D. Brainard and B. A. Wandell. Analyhsis of Retinex Theory of Color Vision. *J. Opt. Soc. Am. A*, 3:1651–1661, 1986.

[6] K. J. Dana, B. van Ginneken, S. K. Nayar, and J. J. Koenderink. Reflectance and texture of real-world surfaces. Technical Report CUCS-048-96, Columbia University, 1996.

[7] E. R. Dixon. Spectral Distribution of Australian Daylight. *J. Opt. Soc. Am. A*, 68:437–450, 1978.

[8] M. D'Zmura and P. Lennie. Mechanisms of Color Constancy. *J. Opt. Soc. Am.*, 68:1662–1672, 1986.

[9] G. Finlayson and S. Hordley. A Theory of Selection for Gamut Mapping Color Constancy. In *Proc. IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recognit.*, 1998.

[10] G. D. Finlayson, M. S. Drew, and B. V. Funt. Spectral Sharpening: Sensor Transformations for Improved Color Constancy. *J. Opt. Soc. Am. A*, 11:1553–1563, 1994.

[11] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes. *Computer Graphics: Principles and Practice, 2nd ed.* Addison-Wesley, Reading, MA, 1990.

[12] D. A. Forsyth. A Novel Algorithm for Color Constancy. *Int. J. Comput. Vision*, 5:5–36, 1990.

[13] B. V. Funt and G. D. Finlayson. Color Constant Color Indexing. *IEEE Trans. PAMI.*, 17, 1995.

[14] R. Gershon, A. D. Jepson, and J. K. Tsotsos. From [R, G, B] to Surface Reflectance. *Perception*, pages 755–758, 1988.

[15] G. Healey and A. Jain. Using Physics-Based Invariant Representations for the Recognition of Regions in Multispectral Images. In *Proc. IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recognit.*, pages 750–755, San Francisco, CA, Jun. 1996.

[16] O. Javed, K. Shafique, and M. Shah. Appearing Modeling in Tracking in Multiple Non-overlapped Cameras. *CVPR*, 2005.

[17] G. J. Klinker, S. A. Shafer, and T. Kanade. Using a Color Reflection Model to Separate Highlights from Object Color. In *Proc. Int. Conf. Comput. Vision*, pages 145–150, 1987.

[18] G. J. Klinker, S. A. Shafer, and T. Kanade. A Physical Approach to Color Image Understanding. *Int. J. Comput. Vision*, 4:7–38, 1990.

[19] R. Kondepudy and G. Healey. Use of Invariants for Recognition of Three-Dimensional Color Textures. *J. Opt. Soc. Am. A*, 11(11):3037–3049, Nov. 1994.

[20] E. H. Land. Recent Advances in Retinex Theory. *Vision Research*, 26:7–21, 1986.

[21] D. H. Marimont and B. A. Wandell. Linear Models of Surface and Illuminant Spectra. *J. Opt. Soc. Am. A*, 9:1905–1913, 1992.

[22] R. Mausfeld and D. Heyer. *Color Perception: Mind and the Physical World*. Oxford University Press, Oxford, England, 2003.

[23] S. K. Nayar, X.-S. Fang, and T. Boult. Separation of reflection components using color and polarization. *Int. J. Comput. Vision*, 21(3):163–186, 1997.

[24] S. K. Nayar, K. Ikeuchi, and T. Kanade. Surface Reflection: Physical and Geometric Perspectives. *IEEE Trans. Pattern Analy. Machine Intell.*, 13, 1991.

[25] J. Platt. Probabilistic outputs for svms and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*. MIT Press, 1999.

[26] F. Porikli. Intercamera Color Calibration using Cross-Correlation Model Function. *IEEE International Conference on Image Processing*, 2003.

[27] Y. Rubner, C. Tomasi, and L. J. Guibas. The Earth Mover's Distance as a Metric for Image Retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.

[28] D. Slater and G. Healey. Using a Spectral Reflectance Model for the Illumination-Invariant Recognition of Local Image Structure. In *Proc. IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recognit.*, pages 770–775, San Francisco, CA, Jun. 1996.

[29] M. Tsukada and Y. Ohta. An Approach to Color Constancy Using Multiple Images. In *Proc. Int. Conf. Comput. Vision*, 1990.

[30] V. N. Vapnik. *Statistical Learning Theory*. Wiley, New York, NY, 1998.