Pose from Color

Mark R. Stevens
Worcester Polytechnic Institute
Worcester, MA 01609 USA
stevensm@cs.wpi.edu

Bruce A. Draper, J. Ross Beveridge
Colorado State University
Fort Collins, CO 80526 USA
draper,ross@cs.colostate.edu

Abstract

Color is a powerful cue for determining the pose of $3D$ objects viewed by a single camera. We present a method based on hue histograms for locating multiple objects with respect to a fixed camera. The algorithm is tested on controlled blocks-world scenes and an indoor office scene. On these examples, a pose refinement algorithm performs better when guided by color than when guided by more traditional edge information.

1. Introduction

Model-based pose refinement typically uses geometric cues to locate $3D$ objects in $2D$ images [5]. Comparatively few methods have been developed using color as the dominant cue [9]. A major reason for the lack of enthusiasm about color can be traced to the lack of good methods for dealing with color constancy [10]. We present a unique method of using hue histograms to recover the $3D$ pose of an object given an initial pose estimate. The process compares two hue histograms: the *observed* distribution obtained from the projection of an object onto an image of a scene, and an *expected* distribution computed from off-line training. By adjusting the object pose with respect to the camera, the difference between these two histograms can be minimized and the $3D$ location of the objects is recovered.

To compute the observed histogram for a new image, a hypothesized *scene configuration* is used. This configuration contains the $3D$ pose of each object believed to be present in the scene. Given a camera model, the scene configuration is rendered to produce a hypothesized image. The hypothesized image indicates which pixels in the sensor image correspond to which objects in the given scene configuration. By histograming the hue pixels for each object in this prediction, we can measure deviation from the expected distribution. When each object's pose is perturbed, better configurations with less deviation are found.

The most closely related work on color based pose determination is a technique known as appearance matching [9]. Appearance matching correlates a new image with the entire set of training data in a lower dimensional Eigen-space. Basri and Jacobs have presented a method for pose determination base on matching regions [1]. Swain has used a method for identifying objects based on histogram intersection [11]. Gevers has used the properties of an object's hue for image retrieval [4].

2 Color Properties of Objects

Often, sensor models use a first order bi-direction reflectance density function (BRDF) to characterize the appearance of a Lambertian object in an image. In the BRDF, radiance observed can be broken into two components: the portion due to diffuse reflection and that due to specular reflection. In theory, pixel values for an object of one color will be distributed along an "inverted-T" shape in the RGB color cube [8]. When the colors are projected onto the hue-saturation plane, a line should appear [2]. One end of this line corresponds to pure diffuse reflection. The other endpoint represents a pure specular reflection at the color of the illuminant. We refer to this line as the *dichromatic line.*

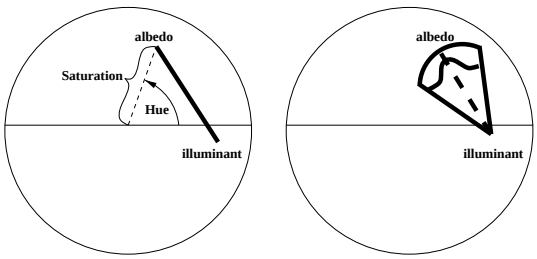

Figure 1. a. The expected colors of a surface lie along a line. b. Surface colors are normally distributed about the dichromatic line.

In images acquired from a CCD sensor, we expect to see pixels normally distributed about the dichromatic

line. For Lambertian surfaces, the expected shape of the cluster is approximated by a pie-slice (see Figure 1b) since pixels will more tightly cluster around the illuminant than the surface color.

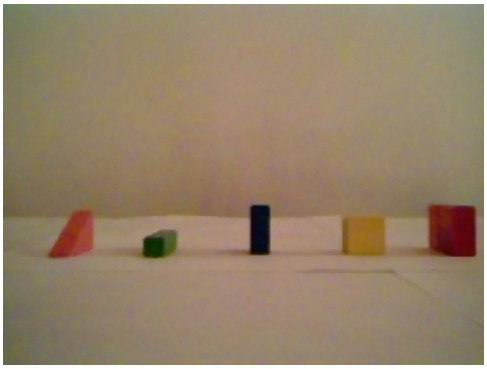

Figure 2. An image of colored blocks on a table. From left to right, the blocks are orange, green, blue, yellow, and red.

Figure 2 shows an example image of five colored blocks. Figure 3 shows the HS projection of the pixels for the green lambertian block. For emphasis, we have added an outlined pie-shape covering the observed values. Also shown in the figure is the CIE color curve, the standard for our illuminant, given in normalized color ($g = 0.866r + 0.831r^2 + 0.134$). We know that the true color of the illuminant will be a point lying on this curve. From observation, we have placed the illuminant at $(0.128, 0.098)$. If only one illuminant is present in the scene, all objects of interest will share a common endpoint for their respective dichromatic line. It is therefore possible to introduce a new coordinate system centered about the illuminant. This adjusted coordinate system is also shown in the figure.

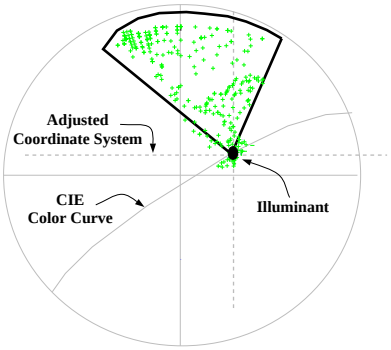

Figure 3. The HS values of the green block.

When objects are not isolated, inter-reflections between objects will exist. These reflections will have two effects on the pixel distribution in HS space. The first is due to noise from the reflecting object and is manifested as a widening of the pie slice (increase in standard deviation of the hue distribution). The second is a rotation of the entire pie-slice, about the illuminant, in the direction of the other object's color (change in the mean of the observed distribution).

2.1 Hue Histograms

Figure 4 shows the normalized hue distribution for each face of the green block in Figure 2. From the figure, we can see that two of the three distributions (for the top, Face 1, and side, Face 2) are shifted to the right. These translations are due to reflections from the blue block. Since the origin has been shifted to the illuminant color, the inter-reflections manifest themselves as translations in the distribution mean.

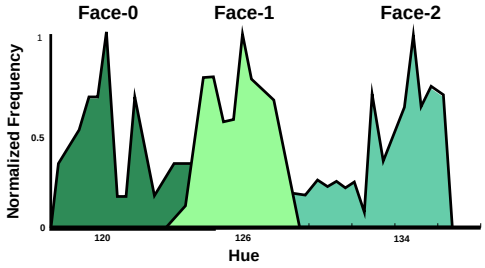

Figure 4. Hue Histograms for the green block.

Two methods may be used for computing the expected hue distribution of a surface:

1. Generate hue distribution from domain knowledge about the surface material and illuminant color.
2. Bootstrap distributions from training images. Pixels from an object viewed in isolation define the expected distribution.

We have chosen to implement the second method. This method has the added advantage of not requiring that the objects be uni-colored. Figure 5 shows the bootstrapped hue distributions for the objects in the dataset.

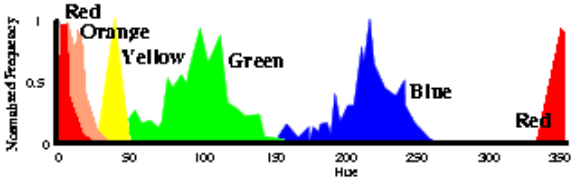

Figure 5. Trained hue histograms.

3 Pose Refinement

Our past work has shown the benefit of using pose-space search for matching [6]. Given an initial scene

configuration, perturbations to the pose of each model are made. For each perturbation, the scene is rendered. This rendering provides a correspondence between the actual features of the $3D$ model and the image pixels. An error function measures the quality of this correspondence. In this paper, match quality is measured in two ways: a) comparing hue histograms, and b) comparing gradient support under face boundaries.

Our search strategy is based on the Simplex algorithm developed by Nelder and Mead [7]. Their algorithm adjusts the worst error point in the simplex in order to lower it's error. In this domain, a different simplex is used for each object. Each point in the simplex represents a pose, and each iteration of search adjusts one object's simplex at a time. The process continues until no further adjustments can be made. Once convergence is reached, the object pose is set to the simplex point with the lowest error.

3.1 Color Cues

An error function to evaluate a given scene configuration using the hue-histogram method is now presented. We have chosen a measure which computes the area between two cumulative distributions:

$$\mathcal{D}^k_{a\to b} \;=\; \frac{\sum_{i=0}^{n} \left|a_i - b_{(i+k)\oplus n}\right|}{\sum_{i=0}^{n} \left(a_i + b_{(i+k)\oplus n}\right)} \qquad (1)$$

where a is the cumulative distribution of the off-line data normalized by sample size, b is the observed normalized cumulative distribution, and n is the number of bins in the histograms (set to 120). We use k to denote a possible shift in the observed histogram relative to the training distribution. This translation compensates for reflections from other objects. Since these distributions are based on angles, the distributions are circular (i.e. a_n is next to a_0). The $\oplus$ operator is modular division used to account for wrap around.

Equation 1 computes the set difference between the two distributions, normalized by their set union. Therefore, $\mathcal{D}^k_{a\to b}$ is guaranteed to be in the range $[0,1]$ and will decrease as the two distributions become equivalent. To evaluate the match for a given face, $\mathcal{F}$, we must find the best shift which aligns the two distributions:

$$\mathcal{E}_{\mathcal{F}_j} \;=\; \min\left(\mathcal{D}^{-5}_{c\to d}\,,\; \mathcal{D}^{-4}_{c\to d}\,,\; \ldots\,,\; \mathcal{D}^{+4}_{c\to d}\,,\; \mathcal{D}^{+5}_{c\to d}\right) \qquad (2)$$

where c is the training distribution for $\mathcal{F}_j$, and d is the observed distribution. The shifted comparison is performed for every visible projected face in the model. The comparisons are then combined to create a measure of how well each model, $\mathcal{M}_i$, matches the image given a hypothesized scene configuration:

$$\mathcal{E}_{\mathcal{M}_i} \;=\; \sum_{j\in\mathcal{F}} (w_j)\left(\mathcal{E}_{\mathcal{F}_j}\right) / \sum_{j\in\mathcal{F}} (w_j) \qquad (3)$$

where w_j represents the number of predicted pixels for a given face. The error for the entire scene is the average error for each model.

3.2 Edge Cues

As a point of comparison with the color measure, an error function for matching $3D$ model edges to a $2D$ image is now presented. This error function has ties to model driven edge detection [3]. The edge cues are based on the gradient lying under each edge as it is projected into the $2D$ image using the current scene configuration. To obtain the gradient at each pixel, the following 1×9 mask is used:

$\vec{A} \;=$

-1	-2	-3	-4	$0\bullet$	$+4$	$+3$	$+2$	$+1$

Such a mask provides gradient information for a large area about the true edge. Therefore, even when the pose is in error by several pixels, the search algorithm will have the directional information necessary to improve the pose. The gradient magnitude at each pixel is defined as:

$$\mathcal{G}_{x,y} \;=\; \sqrt{\left(\vec{X}\cdot\vec{A}\right)^2 + \left(\vec{Y}^{\top}\cdot\vec{A}\right)^2} \qquad (4)$$

where $\vec{X}$ is the column vector of pixels in the horizontal direction centered around the current pixel, and $\vec{Y}$ is the analogous row vector of vertical pixels. Since we are dealing with color imagery, each element of this vector actually contains the V color component from the HSV of the pixel. The gradient magnitude can be turned into an error function by examining each pixel lying under the set of all projected edges, $\mathcal{E}$, for each model, $\mathcal{M}_i$:

$$\mathcal{E}_{\mathcal{M}_i} \;=\; \frac{1}{|\mathcal{E}|}\sum_{x,y\in\mathcal{E}} (1.0 - \mathcal{G}_{x,y}) \qquad (5)$$

where (x,y) represents the pixels underlying all edges $\mathcal{E}$. The error for the entire scene is just the average gradient response under each edge for each model.

4 Evaluation and Results

Experiments were run to compare pose refinement using edge and color cues. Five blocks of varied shape and color were placed in a row on a table. Seven images were taken with the camera panning from left to right. In the non-frontal views, some of the blocks partially occluded

other blocks. The true $3D$ pose of each model was randomly perturbed ten times. Blocks were translated up to $\pm 4.5cm$, which is 1.125 times the largest single-axis bounding box dimension for any block. Blocks were also rotated by up to $\pm 25.0°$. The color pose refinement algorithm was run for each perturbed pose estimate and best result recorded. Similarly, the edge based pose algorithm was run on the same starting configurations and best result recorded.

To compare the use of color versus the use of edge features, two metrics are used. The first is the percentage of *true* object pixels mis-classified P_{TO} (the incorrect object label was given to the object pixel), and the second is the percentage of *labeled* object pixels mis-classified P_{LO} (the incorrect label was given to a labeled pixel). Table 1 shows the values for P_{TO} and P_{LO}. From the table, we see that the hue-based pose refinement algorithm mis-classifies fewer pixels than does the edge based algorithm.

	P_{TO}	P_{LO}
Perturb	0.7112	0.6796
Hue	0.3042	0.2576
Edge	0.5887	0.4071

Table 1. The P_{TO} and P_{LO} values.

Figures 6a and 6b show the histogramed values for P_{TO} and P_{LO} across all 70 perturbations. Pairwise T-tests were run to compare all combinations of these distributions. For P_{TO}, the results showed a statistically significant difference between distributions for the two algorithms.

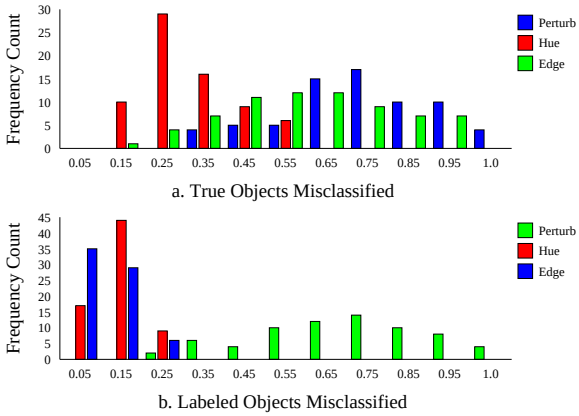

Figure 6. Mis-classification histograms

Figure 7 shows the result of applying the pose refinement algorithm to a more complex set of data. The algorithm was given the perturbed configuration shown in the top image of the figure. The Simplex algorithm then converged to what is shown in the bottom image based solely on color information.

Figure 7. Result for office world.

References

[1] R. Basri and D. Jacobs. Recognition using region correspondences. In *ICCV*, 1995.

[2] S. Buluswar. *Studies in Outdoor Color Machine Vision*. PhD thesis, University of Massachusetts at Amherst, 1998.

[3] P. Fua and Y. G. Leclerc. Model driven edge detection. In *IUW*, pages 1016 – 1021. Morgan Kaufmann, 1988.

[4] T. Gevers and A. Smeulders. Color-metric pattern-card matching for viewpoint invariant image retrieval. In *ICPR*, volume 13, 1996.

[5] W. E. L. Grimson. *Object Recognition by Computer*. MIT Press, 1990.

[6] Mark R. Stevens and J. Ross Beveridge. Precise Matching of 3-D Target Models to Multisensor Data. *Image Processing*, 6(1):126–142, 1997.

[7] J. Nelder and R. Mead. A simplex method for function minimization. *Computer Journal*, 1965.

[8] S. Shafer. Using color to separate reflection components. *Color Research Application*, 10:210–218, 1985.

[9] Shree K. Nayar and Sameer A. Nene and Hiroshi Murase. Real-Time 100 Object Recognition System. In *IUW*. Morgan Kaufmann, 1996.

[10] D. Slater and G. Healey. The illumination-invariant recognition of 3d objects using local color invariants. *PAMI*, 18:206–210, 1996.

[11] M. J. Swain. *Color Indexing*. PhD thesis, University of Rochester, 1990.