Mean-Shift-Based Color Tracking in Illuminance Change

Yuji Hayashi[1] and Hironobu Fujiyoshi[1]

Dept. of Computer Science, Chubu University, Japan
yuji@vision.cs.chubu.ac.jp, hf@cs.chubu.ac.jp
http://www.vision.cs.chubu.ac.jp/

Abstract. The mean-shift algorithm is an efficient technique for tracking 2D blobs through an image. Although it is important to adapt the mean-shift kernel to handle changes in illumination for robot vision at outdoor site, there is presently no clean mechanism for doing this. This paper presents a novel approach for color tracking that is robust to illumination changes for robot vision. We use two interleaved mean-shift procedures to track the spatial location and illumination intensity of a blob in an image. We demonstrate that our method enables efficient real-time tracking of the multiple color blobs against changes in illumination, where the illuminace ranges from 58 to 1,300 lx.

1 Introduction

Tracking is a method of estimating the spatial location of a target in a camera image. It often requires real-time processing, so high-speed processing is essential. For tracking 2D blobs through an image sequence, the mean-shift algorithm is an efficient technique [1–3]. It seeks the nearest mode of a point sample distribution. Collins [4] proposed a method of scale change mean-shift and She [5] proposed a method of considering shape features. The mean-shift algorithm has a low calculation cost and offers high-speed execution.

Tracking is difficult when lighting changes because the RGB values from the image changes with the lighting. Thus, it is not possible to distinguish a moving object or lighting change. In addition, problems of lighting changes are usually treated as those of color transformation between different lighting conditions. Some researchers have proposed linear color transformation [6] and independent transformation [7] of each RGB component, which are derived from a physics-based color model. On the other hand, statistics-based approaches have also been proposed. Miller [8] proposed a method of non-linear color transformation using color eigenflows learned from multiple pairs of images of the same scene under different lighting conditions. It is, however, difficult for a robot vision system to get multiple reference colors in unknown lighting conditions.

This paper presents a novel approach for color tracking that is robust to lighting changes for robot vision. We use two interleaved mean-shift procedures to track the spatial location and illumination intensity of a blob in an image. We show that our method enables real-time tracking of a color blob for varying lighting conditions.

2 Color and illuminance

The illuminance at any surface of known color can be measured by observing the RGB values obtained by a CCD camera. Changes in the light source or meteorological effects can change the illuminance, resulting in changes in the measured RGB values. Figure 1(a) shows various color patches (Blue, Black, Green, Pink, Purple, White, Yellow) under illuminance ranging from 10 to 1400 lx. The setup used in our experiments is illustrated in Figure 1(b). The illuminance on the object's surface was obtained by an illuminance meter placed on the object, and RGB values were captured by a color CCD camera mounted at a height of 280 cm.

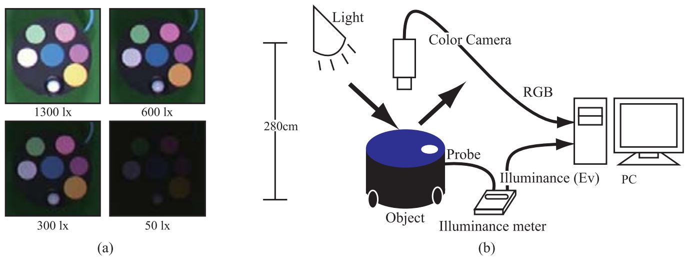


Fig. 1. Experimental setup.

Using the color segmentation technique based on thresholding [9], it is difficult to distinguish color classes in RGB color space, because we cannot create a threshold criterion that specifies how the color space should be divided up into a handful of color classes. Color clustering using the HSI color system is robust to lighting changes, but it is difficult to distinguish a moving object and light change, because it does not represent illuminance on the target. To solve this problem, we augment the RGB color space to make an RGB-illuminance space, and then we use a tracking method that searches for a mode within neighboring pixels.

2.1 RGB-illuminance space

Our approach uses RGB-illuminance space coupled with the estimation of illuminance intensity in each frame to distinguish color classes. An example of color distributions in RB-illuminance space is shown in Figure 2. We can see that it is possible to classify color classes at each illuminance plane, as shown in Figures 2(b) and (c). However, a fixed value for thresholding does not work due to

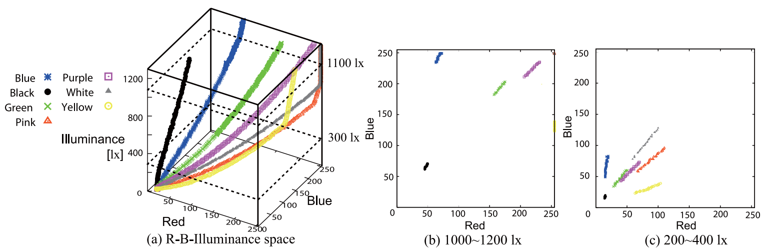


Fig. 2. RGB-illuminance space: (a) color distribution in RB-illuminance space, (b) RB value of each color class at 100 lx, and (c) RB value at 200 lx.

shrinking in the color space with respect to illuminance. In the RGB-illuminance space, reference-based searching such as the k-NN method for color clustering can work, but it takes a lot of time due to the number of reference patterns for each illuminance. Therefore, we use a color-illuminance model for each color class.

2.2 Color-illuminance model

The relationship between RGB values and illuminance is not linear. Thus, we use curve fitting on each RGB distribution over the illuminance intensity. Given the illuminance intensity Ev, we can estimate the RGB color values $\hat{I}_r, \hat{I}_g, \hat{I}_b$ using the following equations:

$$\hat{I}(Ev) = aEv^2 + bEv + c \tag{1}$$

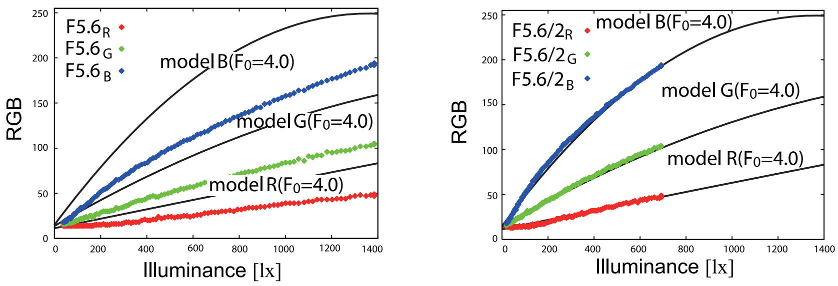
where a, b, and c are unknowns computed by the least-squares method. Note that we assume that the object's surface has diffuse reflection.

2.3 Iris adjustment

The RGB values is influenced by some camera parameters such as iris and white balance. To cope with special lighting situations, the iris (F-number) can be adjusted manually to let in more or less light. The F-number is given by $F = f/D$, where f is the focal length and D is the iris diameter. It affects the amount of light energy admitted to the sensor and plays a significant role in the resulting image. The relationship between intensity I and F-number is expressed by

$$I \propto \left(\frac{D}{f}\right)^2 = \left(\frac{1}{F}\right)^2. \tag{2}$$

The smaller the F-number, the more light admitted to the sensor, and hence the better the image quality achieved in low-light situations. Figure 3(a) shows RGB curves from a color-illuminance model for $F = 4$ and observed RGB values for $F = 5.6$. Using Equation (2), we can convert the RGB values observed at any F-value to the corresponding value at a desired F-value. $F = 5.6$ means $I \propto 31.36$ and $F = 4$ means $I \propto 16$, so the RGB values at 1400 lx with $F = 5.6$ will be same as the RGB values at 700 lx with $F = 4$. Figure 3(b) shows an example of converted RGB values.



(a) RGB(F=5.6) (b) Adjusting RGB

Fig. 3. Adjusting RGB color value by F-number.

If we prepare a color-illuminance model for a given F-value in advance, we can estimate the color-illuminance model at the F-number corresponding to our camera's iris setting. In this paper, we assume that our light source has a constant color temperature, so we do not consider the changes in white balance of the color camera.

3 Mean-shift tracking through illuminace space

We propose a method for mean-shift-based color tracking through illuminance space, which represents the spatial location and illumination intensity of a blob in an image.

3.1 Mean-shift in image space

The mean-shift algorithm is a simple nonparametric method for seeking the nearest mode of a sample distribution. It has recently been adopted as an efficient tracking technique. When the mean-shift method is used for object tracking, the gradient density is formed by weight $w(\mathbf{x})$ at each image pixel $\mathbf{x}$. The core of the mean-shift tracking algorithm is the computation of a target's motion vector from a location $\mathbf{x}$ to a new location $\mathbf{x}'$. We get the new location $\mathbf{x}' = \mathbf{x} + \Delta_{\mathbf{x}}$ from the mean-shift vector

$$\Delta_{\mathbf{x}} = \frac{\sum_{i=1}^{N} K(\mathbf{x}_i - \mathbf{x}_0, \sigma) w(\mathbf{x}_i)(\mathbf{x}_i - \mathbf{x}_0)}{\sum_{i=1}^{N} |K(\mathbf{x}_i - \mathbf{x}_0, \sigma) w(\mathbf{x}_i)|}, \tag{3}$$

where the set $\{\mathbf{x}_i\}_{i=1,\ldots,N}$ represents the locations of pixels around the current location $\mathbf{x}$ and K is a kernel function such as the Gaussian kernel. Generally, a weight map is determined using a color-based appearance model. In [3], the weights were obtained by comparing a histogram q_u, where u is a histogram bin index, with a histogram of colors $p_u(\mathbf{x}_0)$ observed within a mean-shift window at the current location $\mathbf{x}_0$. In fact, the weight at pixel location $\mathbf{x}$ is given by

$$w(\mathbf{x}) = \sum_{u=1}^{m} \delta\left[b(\mathbf{x}) - u\right] \sqrt{\frac{q_u}{p_u(\mathbf{x}_0)}}, \tag{4}$$

where m is the total number of features, δ is the Kronecker delta function and $b(\mathbf{x})$ is feature value of the pixel at $\mathbf{x}$.

3.2 Mean-shift in illuminance space for single-color tracking

It is difficult to track a color blob under varying light conditions due to the limitations of color space described in Section 2. We augment the mean-shift tracker to search in illuminance space by introducing two interleaved mean-shift procedures to track the mode in image space and in illuminance space, which represent the spatial location and illumination intensity of the target blob, respectively. These two procedures are described below.

Initial input A color-illuminance model of the target color is deformed by scaling with the current setting of the iris (F-number). The initial input is a deformed color-illuminance model of a specific color and an estimate of the blob's current illuminance intensity Ev and spatial location $\mathbf{x}_0 = (x, y)$ in the image.

Step 1: Mean-shift in image space Given the illuminance intensity Ev in the current frame, the estimated RGB values $(\hat{I}_r, \hat{I}_g, \hat{I}_b)$ are computed using Equation (1) using the color-illuminance model for the specific color. Then, we compute a location weight map $w_{loc}(\mathbf{x})$ between the target color and the RGB values $I(\mathbf{x})$ for each pixel.

$$w_{loc}(\mathbf{x}_i) = \frac{\hat{I}_r(Ev)I_r(\mathbf{x}_i) + \hat{I}_g(Ev)I_g(\mathbf{x}_i) + \hat{I}_b(Ev)I_b(\mathbf{x}_i)}{\sqrt{(\hat{I}_r^2(Ev) + \hat{I}_g^2(Ev) + \hat{I}_b^2(Ev))(I_r^2(\mathbf{x}_i) + I_g^2(\mathbf{x}_i) + I_b^2(\mathbf{x}_i))}} \tag{5}$$

Then the spatial mean-shift vector is obtained as

$$\Delta\mathbf{x} = \frac{\sum_{i=0}^{N} K_{loc}(\mathbf{x}_i - \mathbf{x}_0, \sigma_{xy})w(\mathbf{x}_i)(\mathbf{x}_i - \mathbf{x}_0)}{\sum_{i=0}^{N} \mid K_{loc}(\mathbf{x}_i - \mathbf{x}_0, \sigma_{xy})w(\mathbf{x}_i) \mid} \tag{6}$$

where K_{loc} is a spatial kernel function given by

$$K_{loc}(\mathbf{x}, \sigma_{xy}) = \frac{1}{2\pi\sigma_{xy}^2} \exp(\frac{-(x^2 + y^2)}{2\sigma_{xy}^2}) \tag{7}$$

and the summations are performed over a local window of N pixels around the current location $\mathbf{x}$. Finally, we can get the new location $\mathbf{x}' = \mathbf{x} + \Delta\mathbf{x}$ from the mean-shift vector.

Step 2: Mean-shift in illuminance space Our approach uses a mean-shift procedure to estimate the illuminance intensity by a local window of pixels around the new location $\mathbf{x}' = (x', y')$ obtained in step 1. First, we compute the color similarity at every illuminance $(k = 0, ..., max)$ for each pixel $\mathbf{x}$ by the following equation.

$$S(k, \mathbf{x}) = \frac{\hat{I}_r(k)I_r(\mathbf{x}) + \hat{I}_g(k)I_g(\mathbf{x}) + \hat{I}_b(k)I_b(\mathbf{x})}{\sqrt{(\hat{I}_r^2(k) + \hat{I}_g^2(k) + \hat{I}_b^2(k))(I_r^2(\mathbf{x}) + I_g^2(\mathbf{x}) + I_b^2(\mathbf{x}))}} \quad (k = 0, ..., max) \tag{8}$$

Then, we compute an illuminance weight map $w_{Ev}()$, which is 1D array, by the following equation:

$$w_{Ev}(k) = \sum_{i=0}^{N} K_{loc}((\mathbf{x}_i - \mathbf{x}'), \sigma_{xy})S(k, \mathbf{x}_i) \tag{9}$$

where K_{loc} is a spatial kernel function. This works as a voting mechanism from neighbor pixels using illuminance, as illustrated in Figure 4.

This mean-shift in illuminance space is performed on the 1D array of results to locate the mode. The illuminance mean-shift vector is then obtained by the equation:

$$\Delta Ev = \frac{\sum_{k=0}^{max} K_{Ev}(k - Ev)w_{Ev}(k)(k - Ev)}{\sum_{k=0}^{max} w_{Ev}(k)}, \tag{10}$$

where Ev is the current illuminance, and K_{Ev} is a kernel function for illuminance space given by

$$K_{Ev}(k, \sigma_{Ev}) = \frac{1}{\sqrt{2\pi\sigma_{Ev}^2}} \exp\left(\frac{-k^2}{2\sigma_{Ev}^2}\right). \tag{11}$$

Finally, we can get the new illumination intensity $Ev' = Ev + \Delta Ev$ from the mean-shift vector.

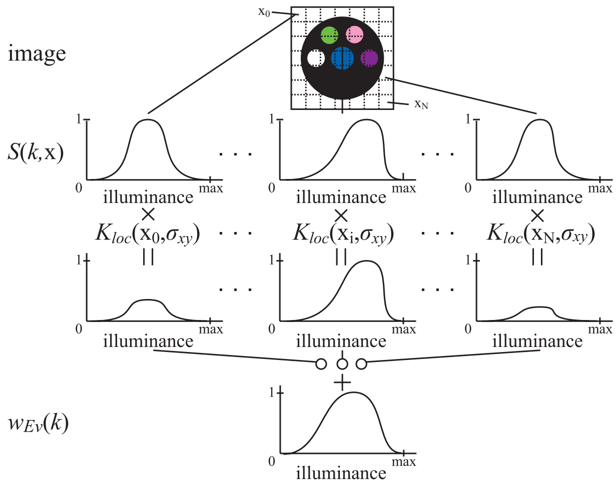


Fig. 4. Calculation of weight map for illuminance space

Since the range of illuminance space is set as $0 < k < max$ (lx), the kernel function for illuminance space K_{Ev} limits the search to around the illuminance estimated in the last frame. However, we cannot get the illumination intensity in a rapid light change if σ_{Ev} is small and the illuminance estimation accuracy is reduced if σ_{Ev} is large. Therefore, we obtain σ_{Ev} from the maximum point k_{max} of the illuminance weight $w_{Ev}(k)$ and the difference in the front frame's illuminance Ev. It is calculated by

$$\sigma_{Ev} = (\sigma_{max} - \sigma_{min}) \times \frac{\mid Ev - k_{max} \mid}{Ev_{max} - Ev_{min}}, \tag{12}$$

where σ_{max} is the maximum value of σ_{Ev}, σ_{min} is the minimum value of σ_{Ev}, Ev_{max} is the maximum value of Ev, and Ev_{min} is the minimum value of Ev.

Step 3: Iteration Iterate by interleaving steps 1 and 2 until both $|\Delta \mathbf{x}| < \varepsilon_{xy}$ and $|\Delta Ev| < \varepsilon_{Ev}$.

3.3 Mean-shift for multiple-color tracking

We augment the single-color tracking method described in Section 3.2 to multiple colors. Multiple color-illuminance models and weight maps for each target color are prepared in advance. For the mean-shift in image space, we compute a spatial location weight map w^c_{loc} for each color class c by Equation (5) using each color-illuminance model. Then, the weights for spatial location are integrated into one weight by selecting the maximum value at the same pixel. The integrated weight map for spatial location w'_{loc} is obtained from each color weight map $w^c_{loc}(c = \mathit{color\ variety})$ by

$$w'_{loc}(\mathbf{x}_i) = w_{loc}(\mathbf{x}_i)^{c1} \frac{w_{loc}(\mathbf{x}_i)^{c1}}{w_{loc}(\mathbf{x}_i)^{c2}}, \tag{13}$$

where $w_{loc}(\mathbf{x}_i)^{c1}$ is the 1st maximum value in multiple colors c at $\mathbf{x}_i$ and $w_{loc}(\mathbf{x}_i)^{c2}$ is 2nd one. Here, the color of $c1$ class, which has the maximum value, is stored for the next step of computing the mean-shift vector in illuminance space.

For the mean-shift in illuminance space, we compute the color similarity for each illuminance for each pixel $\mathbf{x}$ using the color-illuminance model of the target's color. Then, the weight for illuminance space (1D array) is computed according to Equation (9). This mean-shift procedure is iterated until convergence, as described in 3.2.

4 Experimental results

The performance of the proposed method was evaluated by experiments in terms of robustness and accuracy in varying light conditions.

4.1 Experiments

A color camera was mounted at a height of 2800 [cm], as shown in Figure 1. In these experiments, the color temperature of the light source (light color) was fixed, and the white balance and iris value were not changed during the tracking task. Initial illuminance Ev and spatial location (x, y) of the colored object to be tracked were given as initial values for mean-shift tracking. To determine the accuracy of the location estimation, we compared the values estimated by the proposed method to ground truth, which was measured manually by a human. We also measured the illuminance intensity on the surface of the tracked object.

4.2 Experimental results for single-color tracking

Figure 5(a) shows tracking examples of the proposed method and the mean-shift weight map. Figure 6 shows the location errors for the proposed method and the general mean-shift method and the estimated illuminance on the object, which ranged from 50 to 1200 lx as a result of changes in the intensity of the light source. We can see that our method achieved more accurate location estimation than the general mean-shift method. Since it simultaneously computes location and illuminance, i.e., the location of the colored object while estimating the surface illuminance, our method can track the object under varying lighting conditions. When the light changes rapidly, e.g., due to flickering, our method can track a colored object by calculating σ_{Ev} of the illuminance kernel function K_{Ev} at each frame (see Figure 6(a)).

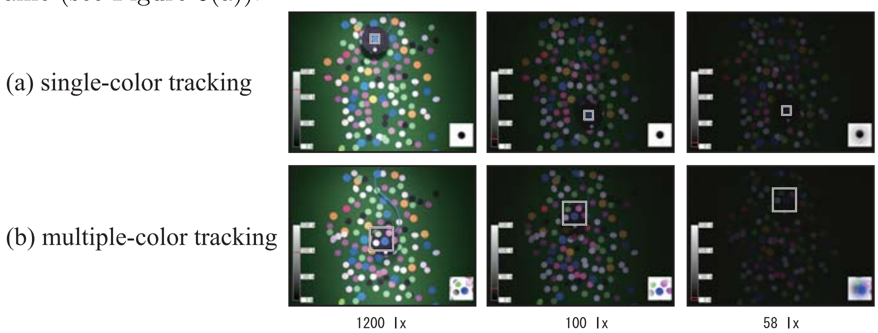
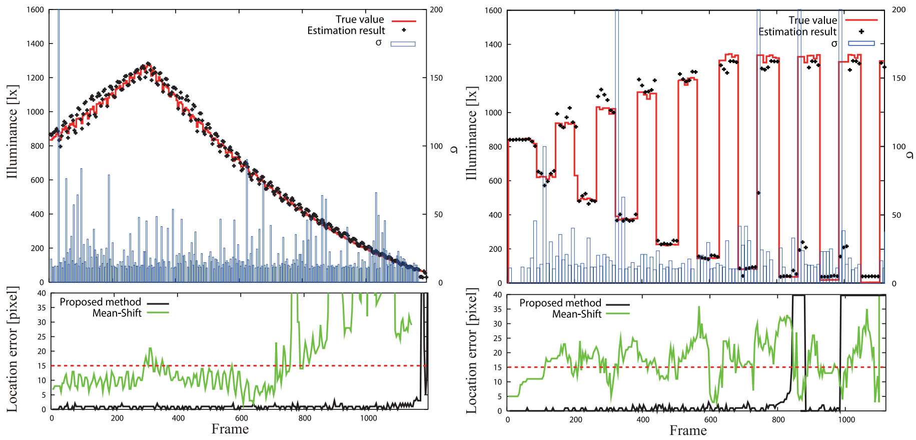


Fig. 5. Tracking example of the proposed method.

Figure 5(b) shows examples of multiple-color tracking by our method using the integrated weight map. It is clear that our method can be easily applied to track multiple colors.



(a)Smooth light change (b)Rapid light change

Fig. 6. Experimental results.

5 Conclusion

In this paper, we proposed a tracking method using two interleaved mean-shift procedures to track the mode in illuminance space, which represents the spatial location and illumination intensity of a blob in an image. We demonstrated that our method enables real-time color tracking that is robust to changes in illumination, where the illuminance ranges from 50 to 1200 lx. Since this method estimates the illuminance from the pixels of the tracked object and not by using the entire image, reliable color tracking is achieved even when the lighting changes. Color tracking when the color temperature of the light source (light color) varies is left as future work.

References

1. K. Fukunaga and L. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory*, 21:32–40, 1975.
2. D. Comaniciu and P. Meer. Mean shift analysis and applications. *IEEE International Conference on Computer Vision*, 2:1197–1203, 1999.
3. D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:564–577, 2003.
4. R. Collins. Mean-shift blob tracking through scale space. *IEEE Conference on Computer Vision and Pattern Recognition*, 2:234–240, 2003.
5. K. She, G. Bebis, H. Gu, and R. Miller. Vehicle tracking using on-line fusion of color and shape features. *IEEE Conference on Intelligent Transportation Systems*, pages 731–736, 2004.
6. D. H. Marimont and B. A. Wandell. Linear models of surface and illuminant spectra. *Journal of the Optical Society of America*, 9(11):1905–1913, 1992.
7. M. S. Drew, J. Wei, and Z.-N. Li. Illumination-invariant image retrieval and video segmentation. *Pattern Recognition*, 32:1369–1388, 1999.
8. E. G. Miller and K. Tieu. Color eigenflows: Statistical modeling of joint color changes. *IEEE International Conference on Computer Vision*, 1:607–614, 2001.
9. J. Bruce, T. Balch, and M. Veloso. Fast and inexpensive color image segmentation for interactive robots. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 3:2061–2066, 2000.