MVA2002 IAPR Workshop on Machine Vision Applications, Dec. 11 - 13, 2002, Nara - ken New Public Hall, Nara, Japan

12—2

Realtime estimation of illumination images using Illumination Eigenspace

Yasuyuki Matsushita†, Ko Nishino‡ *, Katsushi Ikeuchi† and Masao Sakauchi†

† Institute of Industrial Science, The University of Tokyo
yasuyuki@sak, ki@cvl, sakauchi@sak.iis.u-tokyo.ac.jp
‡ Department of Computer Science, Columbia University
kon@cs.columbia.edu

Abstract

An illumination image, which is a part of intrinsic images, represents the effect of a lighting condition of the scene. To properly handle illumination effects such as cast-shadows in the input image, image manipulation using the illumination image is only natural, since it describes variation of lighting effects from a reflectance image which can be considered as an image under the standard illumination. We have shown in previous work [12] that illumination effects are reasonably factored out from the input images by using illumination images. To apply this method as a preprocessing stage to a video surveillance system, realtime estimation of illumination images is required. Unfortunately, the cost of estimation of illumination images in realtime is computationally high. In addition, it is necessary to synthesize background images before deriving illumination images when the scene contains dynamic objects. In this paper, we illustrate our approach to modeling illumination images with principal component analysis (PCA) to directly estimate illumination images from input images which contain moving objects in the scene. We propose this framework presupposing that the camera is fixed and the scene is observed under several lighting conditions.

1 Intrinsic Images

The idea of intrinsic images was first proposed by Barrow and Tenenbaum [2]: the input image I intrinsically is composed of the reflectance image R and the illumination image L, i.e. $I = R \cdot L$. Since the equation is ill-posed, decomposition into the intrinsic images is known to be difficult.

Recently, Weiss [4] proposed an approach to use the series of input images to derive a single reflectance image and the series of illumination images. Since the method relies on the statistics of the natural images, it robustly decouple the reflectance image and the illumination images from the input image sequence. However, since the method does not consider the camera gain parameter, it cannot directly be used with the ordinary inexpensive cameras.

*This work was done while this author was affiliated with Institute of Industrial Science, The Univ. of Tokyo and supported by Japan Science and Technology, Ikeuchi CREST.

We enhance Weiss's method to derive intrinsic images to explicitly take the camera gain into account. We formulate the intrinsic image model as

$$I(x,y,t) = G(t) \cdot R(x,y) \cdot L(x,y,t) \tag{1}$$

where I, G, R and L correspond to input images, camera gains, a reflectance image and illumination images. In log domain, we denote I, G, R and L in a lower case i, g, r and l, respectively. We begin with Equation (2).

$$i(x,y,t) = g(t) + r(x,y) + l(x,y,t) \tag{2}$$

With n spatial derivative filters f_n, we compute a filtered reflectance image r_n by applying Weiss's ML estimation method, which takes a temporal median of filtered input image,

$$\hat{r}_n = median_t\{f_n \star i(x,y,t)\} \tag{3}$$

then compute each filtered illumination image l_n in derivative domain where this estimation was done by $l = i - r$ in Weiss's work.

$$\begin{aligned} \hat{l}_n &= f_n \star i(x,y,t) - median_t\{f_n \star i(x,y,t)\} \\ &= f_n \star i(x,y,t) - \hat{r}_n \end{aligned} \tag{4}$$

Finally camera gains g are computed by taking spatial median of obtained gain images.

$$\hat{g} = median_{x,y}\{i(x,y,t) - \hat{r}(x,y) - \hat{l}(x,y,t)\} \tag{5}$$

This camera gain g actually is a spatial constant which is erased when derivative filters are applied to input images and our method correctly handles the spatially independent factor g.

2 Creating Illumination Eigenspace

Our objective is the modeling of illumination images for realtime estimation of illumination images. Our method first create a lot of the illumination images using previously mentioned method, and store them for the realtime estimation. At the first estimation step, to decompose the series of input images into the intrinsic images, it is necessary to remove moving objects from the input images. Therefore we first create background images in each short time range in the input image sequence, assuming that the illumination condition does not vary in that short time period. We employed the simple averaging of the images for the background estimation, while more rich method would give the better estimates. These background images, $B(x, y, t)$, are used as the input image sequence for estimating intrinsic images.

$$B(x, y, t) = G(t) \cdot R(x, y) \cdot L(x, y, t) \quad (6)$$

To store the estimated illumination images, we propose an *illumination eigenspace* to model variation of illumination images of the scene. The illumination eigenspace is an eigenspace into which only illumination effects are transformed. As a preliminary framework, we use PCA to construct an illumination eigenspace of a target scene, in our case, the crossroad shown in Figure 5. PCA is widely used in signal processing, statistics, and neural computing. This process is also called the Karhunen-Loéve transform. The basic idea in PCA is to find the basic components $[s_1, s_2, ..., s_n]$ that explain the maximum amount of variance possible by n linearly transformed components. Figure 2 shows the hyper-plane constructed by mapping illumination images onto the eigenspace using all eigenvectors.

In our case, we mapped $G(t) \cdot L(x, y, t)$ to the illumination eigenspace, instead of mapping $L(x, y, t)$ only. Because when given an input image, the reflectance image $R(x, y)$ is useful to eliminate the scene texture by computing $I(x, y, t)/R(x, y)$, and the resulting image becomes $G(t) \cdot L(x, y, t)$. Let us denote the product of the camera gain and the illumination image, $L'(x, y, t) = G(t) \cdot L(x, y, t)$. We keep the mapping from $L'(x, y, t)$ to both $G(t)$ and $L(x, y, t)$ for deriving each components. Figure 1 shows the process of creating the illumination eigenspace.

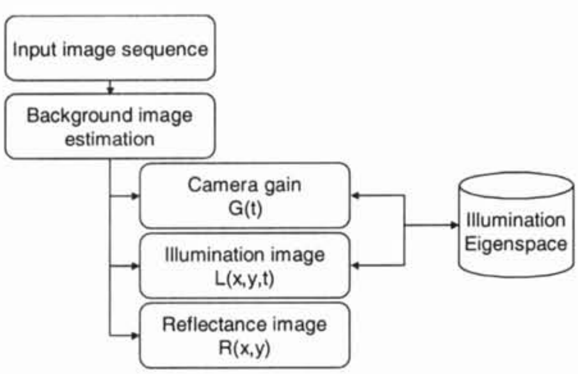

Figure 1: Set up flow of the illumination eigenspace.

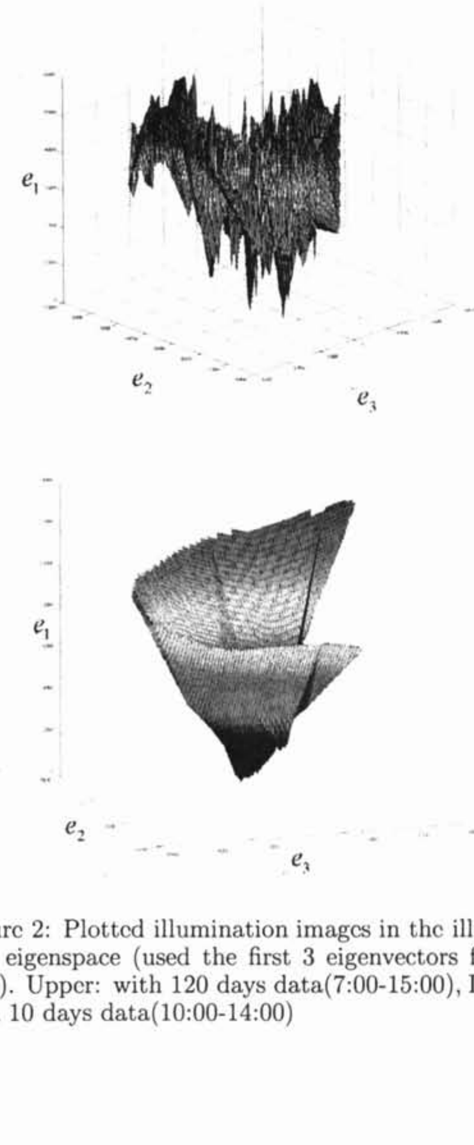

Figure 2: Plotted illumination images in the illumination eigenspace (used the first 3 eigenvectors for display). Upper: with 120 days data(7:00-15:00), Lower : with 10 days data(10:00-14:00)

The upper side image shows the result of mapping all the product of the illumination image and the camera gain, $L'(x, y, t)$, from 120 days(7:00-15:00) while the lower side image illustrates the result of using only 10 days(10:00-14:00) of $L'(x, y, t)$. In the lower side figure, while the three axes represent the first three eigenvectors, the graph is transformed so that the variation along different days is aligned to the vertical axis, which is the first eigenvector (the eigenvector with the largest eigenvalue). Also, the variation along the time-line is shown as the parabolic curve when the graph is sliced orthogonal to the vertical axis. For example, the upper part represents illumination variation along the time-line of a sunny day, and lower part represents that variation on rainy and cloudy days. As can be seen clearly, the most significant variation caused by illumination and time in the $L'(x, y, t)$ can be captured with the first few eigenvectors. So, by constructing an eigenspace of the $L'(x, y, t)$ sequence with the first k significant eigenvectors, and mapping all $L'(x, y, t)$s onto the eigenspace, we can obtain a very efficient representation of the variation of illumination in the input image sequence.

3 Direct estimation of Illumination images

Using the illumination eigenspace, direct estimation of illumination image is done given an input image which contains moving objects. We consider that the global similarity of the illumination image is measured by the distance weighed by contribution ratio of eigenvalues in the illumination eigenspace. Thus, we divide the input image by a reflectance image to get a pseudo illumination image L^* which includes dynamic objects. Using it as a query, the best approximation of the corresponding illumination image $\hat{L}$ is estimated from the illumination eigenspace.

$$\hat{L}' = \arg\min_{L'_i} \sum_j w_j \sqrt{\left(\mathcal{F}(\mathbf{L}^*, j) - \mathcal{F}(\mathbf{L}'_i, j)\right)^2} \qquad (7)$$

where $\mathcal{F}$ is a function which maps an illumination image onto the illumination eigenspace, and $w_j = \lambda_j / \sum_\Omega \lambda_i$ in which we denote λ an eigenvalue. Finally, the true illumination image $L(x, y, t)$ and the camera gain $G(t)$ are derived using the mapping table from $\hat{L}'$. For a high-dimensional nearest neighbor search, we employed the SR-tree method [11] which is known for its fast search algorithm especially for high-dimensional and non-uniform data structures such as natural images. Figure 3 shows the data flow of this direct estimation process.

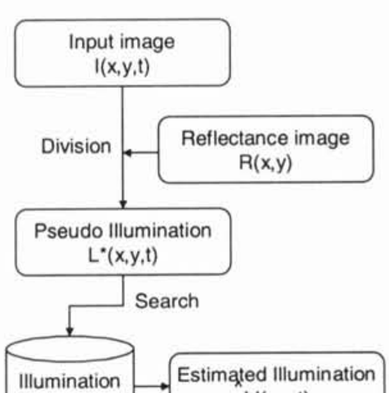

Figure 3: Flow chart of directly estimating the illumination images.

The number of stored images for this experiment was 2048 and the contribution ratio was 84.5% at 13 dimensions, 90.0% at 23 dimensions, and 99.0% at 120 dimensions. The graph of the cumulative contribution ratio is as shown in Figure 4. We choose to use 99.0% of eigenratio for this experiments. Thus the compression ratio is about 17:1, and the space needed to store the subspace is about 32 MBytes.

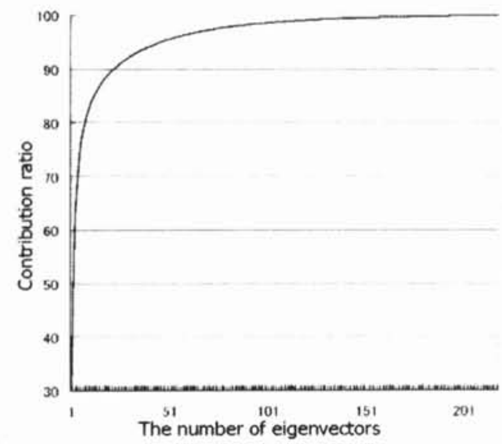

Figure 4: Contribution ratio of the illumination eigenvectors.

4 Result

The result of the illumination image search is shown in Figure 5. In this figure, starting with the left hand side column, the first column shows input images I, the second column shows pseudo illumination images L^*, the third column corresponds to estimated illumination images $\hat{L}'$. The right end column shows the background images which correspond to the estimated illumination images. The nearest neighbor search in PCA is reasonably robust to estimate the most similar illumination image L' from the pseudo illumination image L^*. However, since the sampling of the illumination images is sparse, there are slight differences in shadow shapes. It is possible to acquire the exactly correct illumination image L when the database is dense enough, but it is not easy to prepare such a database. To solve this problem, we are considering to work on shadow interpolation for generating appropriate illumination images. We believe the illumination images derived from our framework has great advantage in even simple interpolation schemes since they are totally free from the scene texture.

As for the computational cost, the average time of the nearest neighbor search is shown in Table 1 with MIPS R12000 300MHz, when the number of stored illumination images is 2048 and the image size is 360×243. Since the input image is obtained at the interval of $33ms$ (at 30 frames/sec), the estimation time is fast enough for the realtime processing.

Dimension	13	23	48	120
Contribution ratio(%)	84.5	90.0	95.0	99.0
NN Search time(μs)	6.7	6.8	7.9	12.0

Table 1: Dimension of the illumination eigenspace, Contribution ratio and NN search cost.

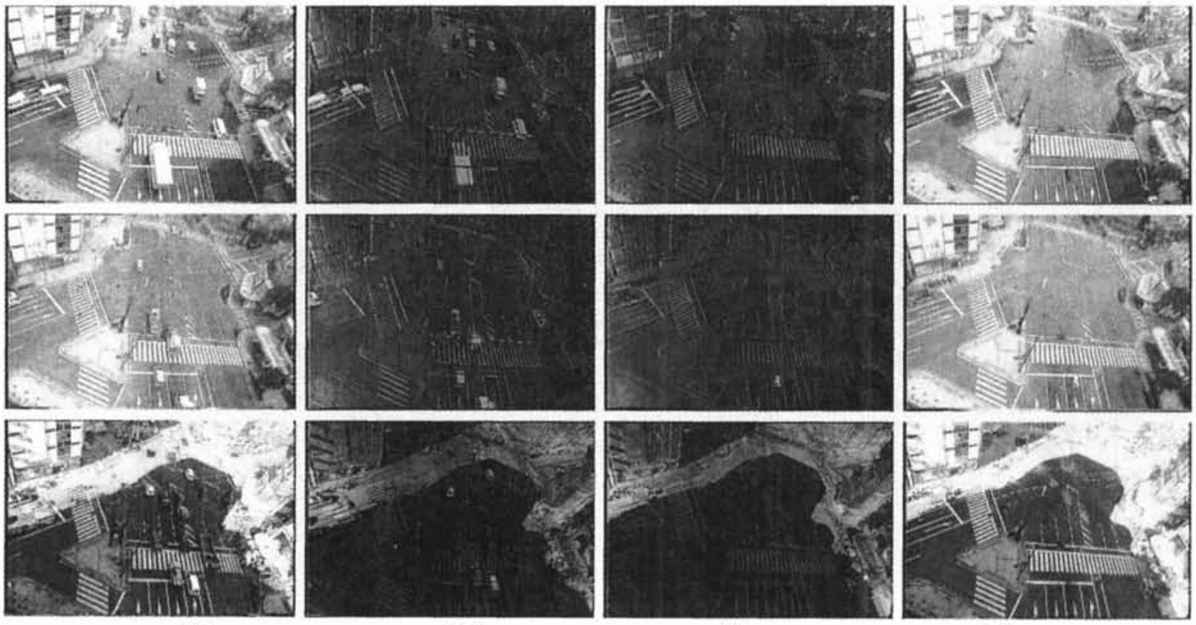

Figure 5: The result of estimating illumination images. (a)Input images I, (b)Pseudo illumination images L^* which are computed by directly dividing input images by a reflectance image, (c)Estimated illumination images $\hat{L}'$ derived by nearest neighbor search in illumination eigenspace, (d)background images corresponding to estimated illumination images

5 Conclusion

In this paper, we present a method to estimate illumination images directly from input images in realtime. Estimated illumination images are used to normalize and manipulate target image sequence with regard to illumination variation, for example, to eliminate shadows, as a preprocessing stage of video surveillance systems using illumination eigenspace. The direct estimation method is demonstrated over an urban scene image dataset which has drastic variations in lighting.

References

[1] Chris Stauffer and W.E.L Grimson. "Adaptive background mixture models for real-time tracking", CVPR99, Fort Colins, CO, (June 1999).

[2] H.G.Barrow and J.M.Tenenbaum, "Recovering intrinsic scene characteristics from images.", In A. Hanson and E. Riseman, editors, Computer Vision Systems. Academic Press, 1978

[3] H. Farid and E.H. Adelson, "Separating reflections from images by use of independent components analysis", Journal of the Optical Society of America, 16(9):2136-2145, 1999.

[4] Yair Weiss, "Deriving intrinsic images from image sequences", Proc. of 9th IEEE International Conference on Computer Vision, Jul 2001.

[5] J. Huang and D. Mumford, "Statistics of Natural Images and Models", IEEE Conference on Computer Vision and Pattern Recognition, pp. 541-547, 1999.

[6] P.L. Rosin and T. Ellis, "Image difference threshold strategies and shadow detection", 6th British Machine Vision Conf., Birmingham, pp. 347-356 1995.

[7] Matthew A. Turk, David G. Morgenthaler, Keith D. Gremban, and Martin Marra, "VITS - A vision system for autonomous land vehicle navigation" IEEE Transaction on Pattern Analysis and Machine Intelligence, 10(3):342-361, May 1988

[8] I.T. Jolliffe, "Principal Component Analysis", Springer-Verlag, 1986.

[9] Shree K. Nayar, Hiroshi Murase, and Sameer A. Nene, "Parametric Appearance Representation", In S. K. Nayar and T. Poggio, editors, Early Visual Learning, pages 131-60, 1996.

[10] Turk, M. and Pentland, A., "Eigenfaces for recognition", The Journal of Cognitive Neuroscience, 3(1): 71-86, 1991.

[11] Norio Katayama and Shin'ichi Satoh, "The SR-tree: An Index Structure for High-Dimensional Nearest Neighbor Queries," Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data (May 1997) pp. 369-380.

[12] Yasuyuki Matsushita, Ko Nishino, Katsushi Ikeuchi and Masao Sakauchi, "Handling Shadow and Illumination for Video Surveillance", In *Proc. of the First European Conf. on Color in Graphics, Image and Vision (CGIV '02)*, pp153–158, Apr. 2002.