Probabilistic Image-Based Rendering with Gaussian Mixture Model

Wenfeng Li and Baoxin Li
Department of Computer Science & Engineering
Arizona State University, Tempe, AZ 85287, USA
E-mail: {wenfeng.li, baoxin.li}@asu.edu

Abstract

One major challenge in traditional image-based rendering is 3D scene reconstruction by estimating accurate dense depth map, which suffers from the ambiguities in textureless or periodically textured regions. Alternatively, statistical methods may be used to estimate a most likely color for each pixel for photorealistic rendering from multiple views of the same scene. Such statistical methods normally require a relatively large number of input images to achieve reasonable quality for the synthesized image, if the estimation is purely nonparametric. In this paper, based on some reasonable assumptions on the configuration of the multiple views, we propose to use a two-component Gaussian mixture model for the appearance of a given pixel in all the views so that both the problem of occlusion and the problem of noise can be considered simultaneously. Then we use the Expectation-Maximization algorithm to estimate the model parameters. The virtual pixel is given as a maximum likelihood estimate for one of the mixture components. Experiments shows that reasonable performance can be obtained even with only a few input images.

1. Introduction

Image-based rendering (IBR), which generally refers to techniques that generate new images from other given images, has been extensively researched for more than a decade. A typical IBR task is to synthesize a new image for a (virtual) viewpoint, given some captured images from different viewpoints. A natural solution followed by many studies is to reconstruct the 3D geometry of the scene from the input images and then to use per-pixel geometry and color information to synthesize the new image. Typically, stereo matching [4] is used to establish per-pixel correspondence based on a pair of input images. When multiple input images are available, full 3D details can be reconstructed by space carving techniques (e.g., [5,6]). While being conceptually intuitive, dense depth estimation remains to be a challenging task due to reasons such as the ambiguity of correspondence in textureless or periodically-textured regions, occlusion, imaging noises, and lighting variations, etc.

When a large number of views are available, an alternative approach is to use probabilistic inference methods for estimating a most likely pixel from the set of pixels in all the views that are determined by, for example, the camera calibration matrices. For instance, a simple average of all pixels in the set can be used as an estimate for the virtual pixel; or, the color corresponding to the peak in the histogram for the set of pixels can be used instead. These simple techniques, together with other existing work (e.g. [1,7], where slightly more sophisticated methods were used) treat each pixel in the set equally, although in reality the pixels may correspond to different scene points. Consequently, these techniques tend to blend the colors of different scene objects (e.g., foreground objects and the background) and thus degrade the synthesized view. This is even more serious with a small number of views. (With a large number of input images, one might hope that the correct color will beat other distractions.)

In this paper, based on some reasonable assumptions on the configuration of the multiple views, we propose to use a two-component Gaussian mixture model for the appearance of the set of pixels corresponding to a virtual pixel location. Then we use the Expectation-Maximization algorithm to estimate the model parameters. The color of the virtual pixel is given as a maximum likelihood estimate for one of the components. By using such a parametric approach, we aim at achieving reasonable performance even with only a relatively small number of views.

2. Problem Statement and Proposed Method

Given N 2-D input images $I_1,\ldots, I_N$. $\boldsymbol{x}$ is a 2-vector $[u, v]$ which is the coordinate of a pixel in an image. $I_i(\boldsymbol{x})$ represents the color of the pixel located at $\boldsymbol{x}$ in the i^{th} image, which is a 3-vector in the RGB color space. I_0 is the image we want to synthesize, also called the virtual view. If all cameras for capturing the input

Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06)
0-7695-2521-0/06 $20.00 © 2006 IEEE

images are fully calibrated, and if the projection matrix of the virtual view is known, we have the 3×4 projection matrices $\boldsymbol{P}_1, \ldots \boldsymbol{P}_N$, and $\boldsymbol{P}_0$, which project a scene point with homogeneous 3D coordinate $\boldsymbol{X}$ to image points $\boldsymbol{x}_i$, written as

$$\hat{\boldsymbol{x}}_i = \boldsymbol{P}_i \boldsymbol{X}, \quad i = 0,\ldots, N, \hat{\boldsymbol{x}}_i = s[u_i, v_i, 1] \tag{1}$$

Since we want to know the color $I_0(\boldsymbol{x})$ for each pixel $\boldsymbol{x}=[u, v]$ in I_0, we trace the ray originating from the camera optical center through the image plane point $\boldsymbol{x}$ to the scene point $\boldsymbol{X}$. If the z value in $\boldsymbol{X}$ is known, which is called depth in this paper, $\boldsymbol{X}$ is determined uniquely by $\boldsymbol{x}$ and $\boldsymbol{P}_0$. Using (1) to map $\boldsymbol{X}$ onto images 1 to N, we get corresponding pixels $I_i(\boldsymbol{x}_i)$ and write the mapping function as $\boldsymbol{x}_i=l_i(\boldsymbol{x})$. Let D be the configuration of depth z for each $\boldsymbol{x}$ in I_0, which is unknown. We want to maximize the *a posteriori* of I_0 and D given the observed data $I_1, \ldots, I_N$, which is

$$p(I_0, D \mid I_1,\ldots I_N) = \frac{p(I_1,\ldots I_N \mid I_0, D) \cdot p(I_0, D)}{p(I_1,\ldots I_N)} \tag{2}$$

In our method, D is used as only a proxy in calculation, and our ultimate goal is to find a good estimation of the colors. Suppose that we do not have any preference over the given views and that there are no special constraints on I_0 and D, then the prior and the denominator can be left out. If we assume that the pixels are i.i.d, we can further write

$$p(I_0, D \mid I_1,\ldots I_N) \propto \prod_x p(I_1(x_1),\ldots I_N(x_N) \mid I_0(x), D(x))$$

$$= \prod_x \prod_{i=1}^{N} p(I_i(x_i) \mid I_0(x), D(x)) \tag{3}$$

2.1 A 2-Component Gaussian Mixture Model

Due to various reasons including lighting condition changes and noise, the projected color of the same scene point $\boldsymbol{X}$ on all input images may not be the same. We model this variation by a Gaussian distribution $N(I_i(\boldsymbol{x}_i); I_0(\boldsymbol{x}), \Sigma)$, i.e., we assume that the desired virtual pixel is the mean of this distribution. (This may require that the virtual view is *among* the given views, and we believe that this is a reasonable requirement or assumption if realistic rendering is the priority.) With this simple distribution, by taking the logarithm of (3), we can find the maximum likelihood estimation (MLE) by minimizing the mean squared error, and the optimal solution for I_0 is the mean of all I_i. This can be viewed as a more general presentation of the strategy of simple average. Experiments show that this simple model works well *if* $\boldsymbol{X}$ is visible in all input images. However, occlusion and noise will greatly degrade the performance of this simple model, and the resultant mean value $I_0(\boldsymbol{x})$ may be distorted severely. We will use the term contamination to denote the problem due to either occlusion or noise. This problem is difficult to address by existing techniques such as using image priors [1]. In the following, we analyze this problem and propose a Gaussian mixture model.

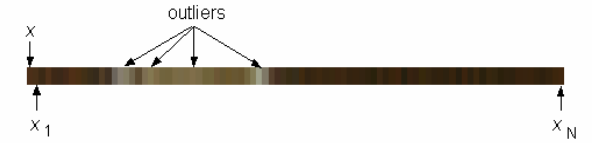

Figure 1. Colors of corresponding points from different views. The outliers do not correspond to the same scene point as the other pixels do.

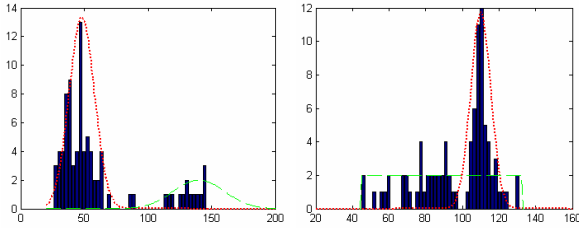

Figure 2. Histograms of color distribution for two pixel locations respectively. The left one may be modeled by two Gaussian components, while the right one is better approximated by a mixture of Gaussian and uniform distributions.

Fig. 1 shows 89 sampling pixels for the same scene point from different views with outliers pointed out. Fig. 2 illustrates the histogram of colors from all corresponding points determined by the projection matrices. Without loss of generality, only the R component is drawn for. Ideally, most of these colors should be the same. However, it is obvious from Fig. 2 that other clusters exist, corresponding to different colors. These pixels of different colors are from outliers. If the outliers are (mostly) from the same scene object (e.g., a relatively homogeneous background), they may form a Gaussian peak as illustrated in Fig. 2(left). On the other hand, if the outliers are from different scenes points, they may result in the (approximately) uniformly-distributed cluster as in Fig. 2(right). Thus the practical problems are to determine whether the set of pixel values corresponding to a virtual pixel are contaminated, and if yes, what the likely model for the contamination may be. To address these problems, we introduce a binary hidden variable V: if $\boldsymbol{X}$ is not contaminated in the i^{th} image then V_i=1, otherwise V_i=0. In implementation, a weighting factor α is used to allow components to be probabilistically associated with each of the classes. Furthermore, to consider the two situations illustrated in Fig. 2 in the same framework, we use another parameter β to control the switch between a uniform distribution and a Gaussian distribution, for modeling the contamination. Thus if we believe a point $\boldsymbol{X}$ is not

contaminated with probability α, the per-pixel mixture likelihood becomes

$$\begin{aligned} &p(I_i(x_i) \mid I_0(x), D(x)) = \\ &\alpha \cdot N(I_i(x_i); I_0(x), \Sigma) + \\ &(1-\alpha) \cdot \{\beta \cdot N(I_i(x_i); I'(x), \Sigma') + (1-\beta) \cdot U\} \end{aligned} \quad (4)$$

where β is a binary switching variable determined by the clustering results (i.e., choose 1 or 0 depending on which value maximizes the likelihood). The task is then to find the MLE maximum likelihood estimate under this 2-component model.

2.2 An EM Solution

For finding the MLE based on Eqn. (4), we use the EM algorithm [9]. In [2] and [3], the EM algorithm is used to estimate the depth from multiple views. There are some possible disadvantages in doing this. For instance, the likelihood function with respect to D is full of local minima but the EM algorithm converges to only a local minimum. Also, there is no analytical dependence between the colors and the depth for a general scene. In our approach, we pursue the estimation for the best color on each depth D^*. Let $\Theta=(I_0, D^*)$, the EM algorithm is as follows.

E-step: Based on the current estimate for Θ^*, the expectation of complete data log-likelihood is

$$\begin{aligned} &E_{p(V|\Theta^*)}[\log L] = \\ &\sum_x \sum_{i=1}^{N} \Pr ob(V_i = 1 \mid \Theta^*) \log(\alpha \cdot N(I_i(x_i) \mid I_0(x), \Sigma) \\ &+ \sum_x \sum_{i=1}^{N} \Pr ob(V_i = 0 \mid \Theta^*) \log((1-\alpha) \cdot \\ &\{\beta \cdot N(I_i(x_i); I'(x), \Sigma') + (1-\beta) \cdot U\}) \end{aligned} \quad (5)$$

where

$$\begin{aligned} &\Pr ob(V_i = 1 \mid \Theta^*) = \\ &\frac{\alpha \cdot N(I_i(x_i) \mid I_0(x), \Sigma)}{\alpha \cdot N(I_i(x_i) \mid I_0(x), \Sigma) + (1-\alpha) \cdot \{\beta \cdot N(I_i(x_i); I'(x), \Sigma') + (1-\beta) \cdot U\})} \\ &\Pr ob(V_i = 0 \mid \Theta^*) = 1 - \Pr ob(V_i = 1 \mid \Theta^*) \end{aligned}$$

M-step: To maximize (5), take derivative with respect to variables to be estimated, we can get

$$I_0(x) = \frac{\sum_{i=1}^{N} \Pr ob(V_i = 1 \mid \Theta^*) I_i(x_i)}{\sum_{i=1}^{N} \Pr ob(V_i = 1 \mid \Theta^*)}$$

$$\Sigma = \frac{\sum_{i=1}^{N} \Pr ob(V_i = 1 \mid \Theta^*)(I_i(x_i) - I_0(x))(I_i(x_i) - I_0(x))^T}{\sum_{i=1}^{N} \Pr ob(V_i = 1 \mid \Theta^*)}$$

$$\alpha = \sum_{i=1}^{N} \Pr ob(V_i = 1 \mid \Theta^*) / N$$

Thus, the mean is estimates by a weighted mean of all observed data, and the weight of each component is proportional to the probability with which it belongs to the uncontaminated class. The above is for the situation when the second component is Gaussian. When the second component is uniform instead, in M-step we computer U as

$$U = \frac{1}{\max\{\|I_i(x) - I_j(x)\| + 1\}}$$

The procedure will stop (converge) when all variables do not change (or change little).

2.3 Weakly Calibrated Views

If we have a large number of views, full calibration of all views will be expensive and also less practical. However it is relatively much easier to compute only the fundamental matrix between any pair of views. We now describe how the proposed method can work with only the epipolar constraint (weak calibration).

For any point $\boldsymbol{x}=[u\ v]$ in I_0, its corresponding point $\boldsymbol{x}_1=[u_1\ v_1]$ in image I_1 is constrained by an epipolar line which can be computed with the fundamental matrix F_{01}. If we know the horizontal disparity between $\boldsymbol{x}$ and $\boldsymbol{x}_1$ which is $d=u_1-u$, $\boldsymbol{x}_1$ is determined. With the correspondence between $\boldsymbol{x}$ and $\boldsymbol{x}_1$, a third point $\boldsymbol{x}_2$ can be located in image I_2 by the intersection of two epipolar lines constrained by $\boldsymbol{x}$, $\boldsymbol{x}_1$ and the fundamental matrix F_{02}, F_{12}. Now I_1 and I_2 work as two anchors to locate other corresponding points in all input images. Thus the 3D depth z of the scene is mapped into a disparity d in an anchor view, and we can use the method described in Sect. 2.2.

One difficulty is to find the fundamental matrix between the virtual view and the two anchor views without camera calibration data. We have developed an approach to this problem [11]. Another problem with epipolar constraints is the degeneracy when two epipolar lines cannot determine a reliable intersection, such as the case when two epipolar lines are nearly parallel. Our experiments proved that pixels from such views will be classified as outliers and will not affect the final result. Our experiments reported in the next section are based on only weak calibration.

3. Experiments

Sample results based on experiments with the "monkey scene" data set from [1] are presented in this section to illustrate the performance of the proposed method. In the dataset there are 89 images from different viewpoints each with camera calibration information. We randomly pick the input views, compute their fundamental matrices. Different numbers of input images are tested in our algorithm.

With only 8 to 10 inputs, we can achieve reasonable results. Fig. 3(b) is one sample result synthesized with 8 images. For comparison we also implement an algorithm based on a simple Gaussian model (Fig. 3(a)) with the same number of inputs, where artifacts are obvious in regions such as those marked by red circles. Fig. 4 supplies another example from different virtual viewpoint.

4. Conclusion and Future Work

While having some attractive advantages, probabilistic IBR typically requires a large number of views. To alleviate this problem, we proposed an approach based on a 2-component Gaussian mixture model for the pixels linked to a virtual pixel. Experiments show the effectiveness of the method in face of only a small number of views. It was noticed in our experiments that some details are not kept well when these details are surrounded by textureless background. This can be explained by the consistency of the background pixels, which causes them to be classified as the true object pixels. One potential remedy is to use the texture synthesis technique such as the image prior used in [1] and [7].

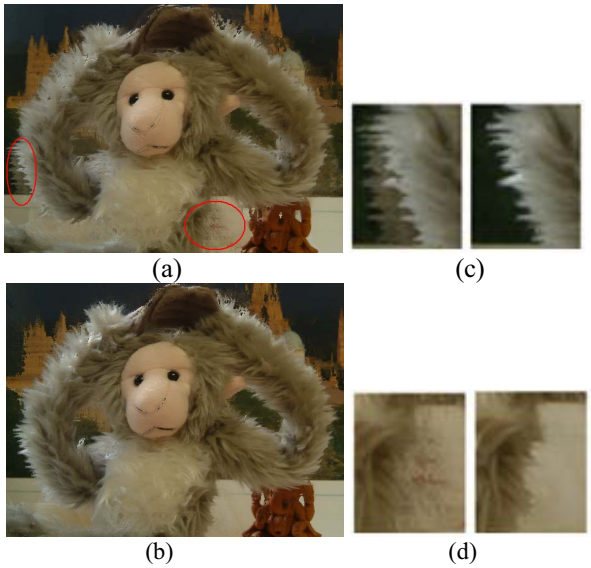

Figure 3. (a)Synthesized view with a simple Gaussian model; (b) Synthesized view with the proposed method; (c)-(d) A close look at the two encircled regions and comparisons (the right hand side images are from (b)).

References

[1] A. Fitzgibbon, Y. Wexler, and A. Zisserman, "Image-Based Rendering Using Image-Based Priors", ICCV, pp. 1176-1183, 2003.

[2] P. Gargallo, P.F. Sturm, "Bayesian 3D Modeling from Images Using Multiple Depth Maps", CVPR, pp. 885-891, 2005.

[3] C. Strecha, R. Fransens, and L. V. Gool, "Wide-Baseline Stereo from Multiple Views: A Probabilistic Account", CVPR, pp. 552-559, 2004.

[4] D. Scharstein and R. Szeliski, "A Taxonomy and Evaluation of Dense Two-frame Stereo Correspondence Algorithms", IJCV, 47(1): pp. 7–42, 2002.

[5] A. Broadhurst and R. Cipolla, "A Statistical Consistency Check for the Space Carving Algorithm", ICCV, pp. 388-393, 2001.

[6] S. M. Seitz and C. R. Dyer, "Photorealistic Scene Reconstruction by Voxel Coloring",CVPR, 1997.

[7] O. Woodford and A. Fitzgibbon, "Fast Image-based Rendering using Hierarchical Image-based Priors", BMVC, 2005.

[8] D.J. Miller and J. Browning, "A Mixture Model and EM-Based Algorithm for Class Discovery, Robust Classification, and Outlier Rejection in Mixed Labeled/Unlabeled Data Sets", *IEEE Trans. Pattern Analysis and Machine Intelligence,* 25(11), pp. 1468-1483, 2003.

[9] A. Dempster, N. Laird, and D. Rubin, "Maximum-Likelihood from Incomplete Data via the EM Algorithm", *J. Royal Statistical Soc. B*, 39, pp. 1-38, 1977.

[10] P.H.S. Torr, A. Zisserman, and S.J. Maybank, "Robust Detection of Degenerate Configurations for the Fundamental Matrix", ICCV, p. 1037, 1995.

[11] W. Li, J. Zhou, B. Li and M. I. Sezan, "Virtual View Specification and Synthesis in Free Viewpoint Television Application", *Third International Symposium on 3D Data Processing, Visualization and Transmission,*2006.

Figure 4. Synthesized view from another viewpoint. Top: Based on a simple Gaussian model; Bottom: From the proposed method.