An Efficient Image-Based Rendering Method[†]

Jian Yao Wai-Kuen Cham
Department of Electronic Engineering
The Chinese University of Hong Kong, Shatin, N.T., Hong Kong
E-mail: {jianyao,wkcham}@ee.cuhk.edu.hk

Abstract

Given a set of images of the same scene, we propose an efficient method for realistically synthesizing a new image seen from a new viewpoint. Compared with some existing techniques which explicitly reconstruct the 3D geometry of the scene, we directly reconstruct the colour of each pixel in the new image by utilizing a weighted photoconsistency constraint. Global photoconsistency constraint of one pixel is firstly utilized to generate a list of plausible colours for each rendered pixel in the new image. Considering the existing of partial occlusion or the deficiencies in the image-formation model, we iteratively update the colour for each rendered pixel based on local texture statistics similar to the input images. Experimental results on the generation of a new image from a new viewing position from a set of input images show that our proposed method is promising and satisfactory.

1 Introduction

Recently, there has been much interest in computer vision and graphics in image-based rendering (IBR) methods [1–3], which generate new views of scenes from novel viewpoints, using a collection of images with known viewpoints. The creation of novel views using pre-stored images or image-based rendering has many potential applications, such as visual simulation, virtual reality [4], and telepresence, for which traditional computer graphics based on geometric modelling would be unsatisfactory particularly with very complex three-dimensional scenes.

One natural approach to IBR is to explicitly compute a 3D representation of the scene for rendering by texture mapping from a collection of images utilizing computer-vision-based 3-D reconstruction techniques [5, 6]. Arbitrary views of the scene can be synthesized by reprojecting the reconstructed 3-D model [7]. Typical examples of this approach are stereo reconstructions [8] and volumetric techniques such as space carving [9, 10]. More recent work has shown that some implicit-geometry techniques [1, 11], which assemble the pixels of the synthesized view from the rays sampled by the pixels of the input images, are very efficient for rendering of complex scenes. All these approaches assume rigid 3-D scenes.

In this paper, an efficient image-based rendering method is proposed to deal with this problem in which the computation of scene geometry is implicit. The proposed algorithm comes of the recently proposed method in [11]. There are several differences between our algorithm and their proposed one. First we introduce the weighting factors based on the position and orientation of the view to be synthesized, deviated from positions and orientations of input images. Second we can find all local photoconsistency minima by explicitly sampling pre-defined depth searching range in the linear way and thus we can generate a new image in the faster speed. In [11], however, they isolated the minima by starting gradient descent from several randomly chosen starting points in an appropriate colorspace. Thus the whole optimization process is very slow and many minima would be lost. Finally, in the updating step for rendering of each pixel by incorporating texture priors, we reduce the space of searching the best matching texture patch so as to improve the speed.

2 Problem Formulation

Given a set of 2D input images $\mathcal{I}_1$ to $\mathcal{I}_n$ taken by cameras in different positions represented by 3×4 projection matrices $\mathtt{P}_1$ to $\mathtt{P}_n$, a new view from a new viewpoint represented by projection matrix $\mathtt{P}_v$ would be synthesized. Figure 1 demonstrates the view synthesis situation. Assumed that we are dealing with diffuse, opaque objects, and any deviation from this assumption may be considered as part of imaging noise. Both weighted photoconsistency constraint and texture priors on input images are utilized to complete the task of virtual view synthesis.

2.1 Photoconsistency constraint of one pixel

To render a particular pixel (x, y) in the synthesized view denoted by $V(x, y)$ as a 3-vector in an appropriate col-

[†]This work is partially supported by the Direct Grant 2050284.

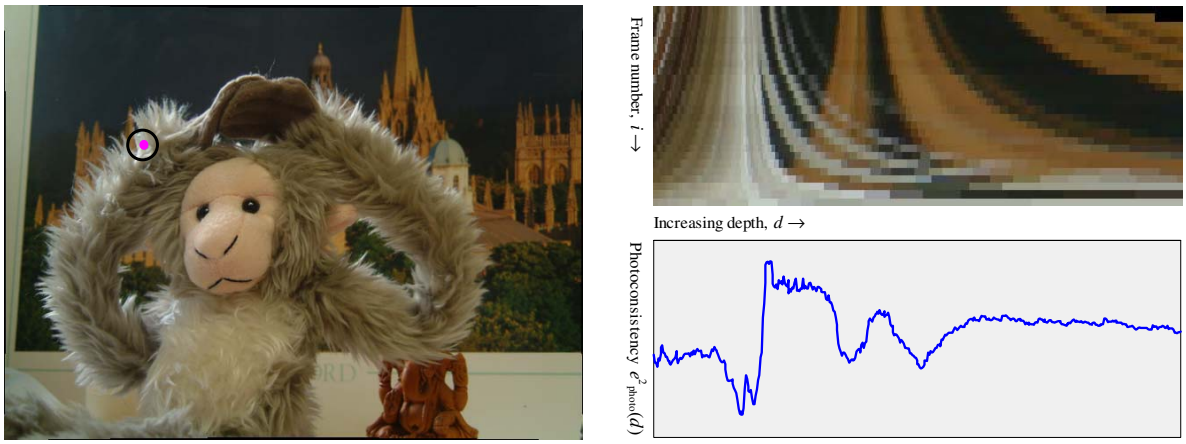

Figure 2. Weighted photoconsistency constraint. *One image is shown from a sequence of 27 captured by a hand-held camera. The circled pixel's photoconsistency with respect to the other 26 images is illustrated on the right. The upper right image shows the re-projected colour set $\mathcal{C}$. Below are shown photoconsistency errors $e^2_{\text{photo}}(d)$. The multi-modality of photoconsistency errors is the essence of the ambiguity in a new view to be synthesized, which we can utilized texture priors knowledge from input images to remove.*

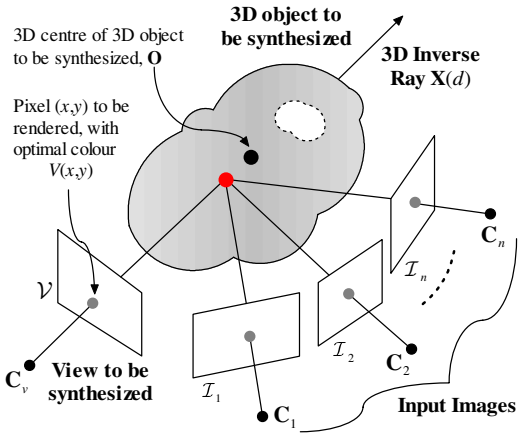

Figure 1. View synthesis system configuration.

orspace, we firstly construct a colour set consisting of re-projection onto the input images of some 3D points on the back-projection ray at (x, y) emanating from the camera center $\mathbf{C}_v$ of the synthesized view. Let the unit direction of this back-projection ray be denoted as $\mathbf{u}(x, y)$ which can be computed as follows:

$$\mathbf{u}(x,y) = \frac{\varpi(\mathtt{P}_v^+\mathbf{x}) - \mathbf{C}_v}{\|\varpi(\mathtt{P}_v^+\mathbf{x}) - \mathbf{C}_v\|}, \tag{1}$$

where $\mathtt{P}_v^+$ is the pseudo-inverse of $\mathtt{P}_v$, $\mathbf{x} = (x, y, 1)^\top$ which a homogenous 3-vector, the inhomogenous 3-vector $\mathbf{C}_v$ is the camera center of $\mathtt{P}_v$, and $\varpi(\mathtt{X},\mathtt{Y},\mathtt{Z},\mathtt{T}) = (\mathtt{X}/\mathtt{T},\mathtt{Y}/\mathtt{T},\mathtt{Z}/\mathtt{T})^\top$. Let a 3D point along the ray be given by the function:

$$\mathbf{X}(d) = ((\mathbf{C}_v + d\mathbf{u}(x,y))^\top, 1)^\top \tag{2}$$

where d is the depth of this 3D point and it ranges between preset values $d_{\min}$ and $d_{\max}$. We can first obtain the minimal and maximal depth values $\tilde{d}_{\min}$ and $\tilde{d}_{\max}$ of a set of reconstructed 3D points corresponding to input images with respective to $\mathtt{P}_v$. To obtain a feasible depth searching range $[d_{\min}, d_{\max}]$ for each pixel on the view $\mathcal{V}$, we suitably broaden the depth range $[\tilde{d}_{\min}, \tilde{d}_{\max}]$ and assume that all re-projected points on all input images are available or visible. For a given depth d, we can compute the set of re-projected pixels on the input images

$$C(i,d) = I_i(\mathbf{X}(d)) = I_i(\pi(\mathtt{P}_i\mathbf{X}(d))), \tag{3}$$

where $\pi(x, y, w) = (x/w, y/w)$ and $I_i(\mathbf{X}(d))$ denote the pixel in the i-th image to which 3D point $\mathbf{X}(d)$ projects. For saving the computing time, we compute the re-projected point as follows:

$$\begin{aligned} \mathtt{P}_i\mathbf{X}(d) &= [\mathtt{M}_i|\mathbf{m}_i]((\mathbf{C}_v + d\mathbf{u}(x,y))^\top, 1)^\top \\ &= (\mathtt{M}_i\mathbf{C}_v + \mathbf{m}_i) + d(\mathtt{M}_i\mathbf{u}(x,y)), \end{aligned} \tag{4}$$

where $\mathtt{M}_i$ and $\mathbf{m}_i$ denote the first 3×3 submatrix and the last column of $\mathtt{P}_i$ respectively. So we only compute $\mathtt{M}_i\mathbf{C}_v + \mathbf{m}_i$ once for all the pixels to be rendered. However, $\mathtt{M}_i\mathbf{u}(x, y)$ is computed once only for each pixel.

From the above computation, we get the colour set:

$$\mathcal{C} = \{C(i,d) | 1 \le i \le n, d_{\min} \le d \le d_{\max}\}. \tag{5}$$

In order to choose the colour $V(x, y)$, we shall search the depth range $[d_{\min}, d_{\max}]$ and the whole RGB colorspace by minimizing the weighted photoconsistency average error function as follows:

$$e^2_{\text{photo}}(V(x,y),d) = \sum_{i=1}^{n} w_i \|V(x,y) - C(i,d)\|^2, \tag{6}$$

where w_i is a weighting factor for the i-th image. In general, we set the weighting factors $\{w_i\}$ based on the combination of both the relative deviations of camera directions of input images $\mathcal{I}_1$ to $\mathcal{I}_n$ with respective to the view $\mathcal{V}$ to be synthesized and relative distances of camera centers $\mathbf{C}_1$ to $\mathbf{C}_n$ deviated from the center of 3D object to be synthesized with respective to the camera center $\mathbf{C}_v$ as follows:

$$w_i = e^{-\alpha\left(\beta \frac{\|\boldsymbol{\theta}_i - \boldsymbol{\theta}_v\|}{\max_i \|\boldsymbol{\theta}_i - \boldsymbol{\theta}_v\|} + (1-\beta)\frac{\|\mathbf{C}_i - \mathbf{O}\|/\|\mathbf{C}_v - \mathbf{O}\|}{\max_i \|\mathbf{C}_i - \mathbf{O}\|/\|\mathbf{C}_v - \mathbf{O}\|}\right)}, \quad (7)$$

where $\boldsymbol{\theta}_i = (\theta_x^i, \theta_y^i, \theta_z^i)$ and $\boldsymbol{\theta}_v = (\theta_x^v, \theta_y^v, \theta_z^v)$ denote the rotation angles of cameras of the i-th input image and the view $\mathcal{V}$ to be synthesized respectively with respective to the world frame. $\mathbf{C}_i$ and $\mathbf{C}_v$ denote camera centers of the i-th input images and the view $\mathcal{V}$ to be synthesized respectively. $\mathbf{O}$ is an approximative 3D center of the 3D object to be synthesized. The coefficients α and β are two adjustable nonnegative parameters. Generally, the rotation angles deviated from the view $\mathcal{V}$ affect the final synthesized view more than the distances deviated from the center of 3D object to be synthesized.

For a particular depth d, we compute the optimal colour $V_{\text{opt}}(x, y, d)$:

$$V_{\text{opt}}(x, y, d) = \frac{\sum_{i=1}^{n} w_i C(i, d)}{\sum_i^n w_i}. \quad (8)$$

Thus we can find the optimal depth value by minimizing the following error function:

$$e^2_{\text{photo}}(d) = \min_d \sum_{i=1}^{n} w_i \|V_{\text{opt}}(x, y, d) - C(i, d)\|^2. \quad (9)$$

In the implementation, the minimum over d is computed by explicitly sampling, typically using 500 values. Figure 2 shows an example of the colour set $\mathcal{C}$ and photoconsistency error $e^2_{\text{photo}}(d)$ at one pixel in a real sequence.

2.2 Incorporating Texture Priors

Utilizing the weighted photoconsistency constraint, we find the optimal colour $V(x, y)$ by minimizing (6). From Figure 2, we observe that multiple local minima are generally generated, due firstly to some physical factors such as occlusion and partial pixel effects and secondly to deficiencies in the image-formation model, such as not modelling specular reflections or having an inaccurate model of imaging noise [11]. Thus the photoconsistency error $e^2_{\text{photo}}(d)$ at the true depth value may often be larger than the errors $e^2_{\text{photo}}(d)$ at other spurious depth values. To remove this ambiguity, local texture statistics similar to the input images are utilized to choose a best matching texture patch most likely corresponding to the rendered colour $V(x, y)$ by minimizing the following error function over d:

$$e^2_{\text{texture}}(\mathcal{P}_i(x, y, d)) = \min_{i,d} \|\mathtt{M}(\mathcal{P}_i(x, y, d) - \mathcal{N}(x, y))\|^2, \quad (10)$$

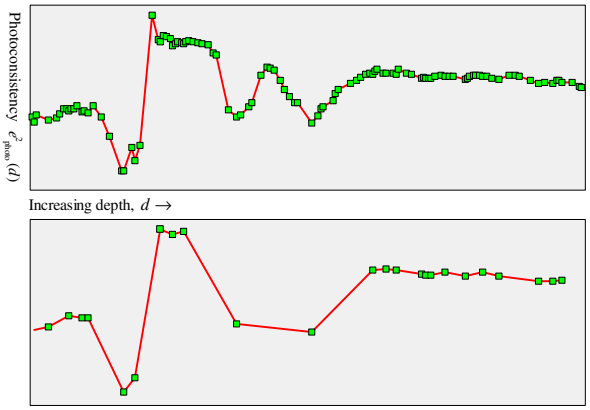

Figure 3. Depth sets with local minimal photoconsistency errors for the circled pixel in Figure 2. *The upper image shows all the valid depth values with local photoconsistency minima $e^2_{\text{photo}}(d)$. Below is shown the final set of plausible depth values for minimizing (10).*

where the function $\mathcal{N}(x, y)$ is the set of colours of neighbours of (x, y) in the view $\mathcal{V}$. $\mathcal{P}_i(x, y, d)$ is the set of colours of beighbours of re-projected pixel of $\mathbf{X}(d)$ on the i-th image for a given depth d, and $\mathtt{M}$ is a mask ignoring the centre pixel. Here we use 5×5 neighbourhoods, so

$$\mathcal{N}(x, y) = \{V(x+\Delta_x, y+\Delta_y) | -2 \leq \Delta_x, \Delta_y \leq 2\}. \quad (11)$$

The set $\mathcal{P}_i(x, y, d)$ consists of the similarly bounding neighbourhoods of the re-projected pixel $I_i(\pi(\mathtt{P}_i\mathbf{X}(x, y, d)))$.

Furthermore, we update the rendered colour for pixel (x, y) using the following equation [11]:

$$V'(x, y) = \frac{V(x, y) + \lambda T}{1 + \lambda}, \quad (12)$$

where T is the value at the centre pixel of the best matching texture patch and λ is a nonnegative adjustable parameter. Finally, $V'(x, y)$ will be replaced by the closest mode with local photoconsistency minima $e^2_{\text{photo}}(d)$ at (x,y).

To save the best matching texture patch searching time, we only deal with a small set of depth values with local minimal photoconsistency errors $e^2_{\text{photo}}(d)$ at (x, y). We find these depth values as follows. First we scan all the depth samples and find a set of valid depth values at which photoconsistency errors $e^2_{\text{photo}}(d)$ is less than ones at adjacent two depth samples. Again, we scan the above obtained depth set and find the final set of depth values in the same way for use in the best matching texture patch searching process. An example is shown in Figure 3.

For each pixel (x, y) on the view $\mathcal{V}$, we yield an initial estimate $V(x, y) = V_{\text{opt}}(x, y, \tilde{d})$ where $\tilde{d}$ corresponds to most likely mode with global photoconsistency mimina using (9). Then we iteratively find the best matching texture

(a) (b) (c) (d)

Figure 4. Leave-one-out test. *Using 26 views to render a missing view allows comparison to be made between the rendered view and ground truth. (a) Ground-truth view. (b) View with global photoconsistency minima $e^2_{\text{photo}}(d)$. (c) View synthesized using texture priors. (d) Difference image between (b) and (c).*

patch using (10) and further update the rendered colour at (x, y) using (12) until no change of optimal rendered colour for each pixel (x, y) on the view $\mathcal{V}$.

3 Experiments

To test the validity of our algorithm, image sequences captured using a hand-held camera will be utilized. First the image sequences were calibrated using a commercial camera tracking software [12]. A set of corresponding points was firstly reconstructed in 3D space. Then we locate an approximative 3D centre point **O** of the scene by minimizing the root-mean-square (RMS) of distances between the reconstructed 3D points and centre point **O**. Finally a suitable depth searching range for a new view to be synthesized will be found by the 3D reconstructed points and the approximative 3D centre point **O**.

In our experiments, we utilize the same image sequences used in [11]. The complete MPEG sequences, camera projection matrices, and the reconstructed 3D points may be found at `http://www.robots.ox.ac.uk/~awf/ibr`. We do the same leave-one-out test as in [11] and the recovered results are shown in Figure 4. Compared with the recovered results in [11], there are no existing obvious high-frequency artifacts in the rendered scene because we can find global photoconsistency minima by explicitly sampling the depth searching range.

4 Conclusions

In this paper, we propose an efficient method to synthesize new images from new viewpoints based on the statistics of input images with known viewpoints. First we utilize the weighted photoconsistency constraint of one pixel to find the optimal colour for rendering for each pixel on the view to be synthesized. Considering the existing of partial occlusion or deficiencies in the image-formation model, we further update the rendered colour by assuming that the generated view has the similar local texture statistics to input images. The experimental results show that the proposed algorithm can efficiently synthesize a new image from a new viewpoint.

References

[1] M. Lhuillier and L. Quan, "Image-Based Rendering by Joint View Triangulation," *IEEE Trans. Circuits Syst. Video Technol.*, Vol. 13, No. 11, pp.1051-1063, Nov. 2003.

[2] C. Zhang and T. Chen, "Spectral Analysis for Sampling Image-Based Rendering Data," *IEEE Trans. Circuits Syst. Video Technol.*, Vol. 13, No. 11, pp.1038-1050, Nov. 2003.

[3] M.M. Oliveira, "Image-Based Modeling and Rendering Techniques: A Survey," *RITA - Revista de Informática Teóricae Aplicada*, Vol. IX, No. 2, pp.37-66, Oct. 2002.

[4] R.J. Radke, P.J. Ramadge, S.R. Kulkarni, and T. Echigo, "Efficiently Synthesizing Virtual Video," *IEEE Trans. Circuits Syst. Video Technol.*, Vol. 13, No. 4, pp.325-337, Apr. 2003.

[5] M. Magnor, P. Ramanathan, and B. Girod, "Multi-View Coding for Image-Based Rendering Using 3-D Scene Geometry," *IEEE Trans. Circuits Syst. Video Technol.*, Vol. 13, No. 11, pp.1092-1106, Nov. 2003.

[6] A.Y. Mulayim, U. Yilmaz, and V. Atalay, "Silhouette-Based 3-D Model Reconstruction From Multiple Images," *IEEE Trans. Syst., Man, Cybern. B*, Vol.33, No.4, pp.582-591, Aug. 2003.

[7] S. Yaguchi and H. Saito , "Arbitrary Viewpoint Video Synthesis From Multiple Uncalibrated Cameras," *IEEE Trans. Syst., Man, Cybern. B* (Accepted).

[8] P.J. Narayanan, P.W. Rander, and T. Kanade, "Constructing virtual worlds using dense stereo," in *Pro. 5th Eur. Conf. Comp. Vision,* Freiburg, Germany, pp.3-10, 1998.

[9] A. Broadhurst and R. Cipolla, "A Statistical Consistency Check for the Space Carving Algorithm," In *Proc. ICCV*, 2001.

[10] S.M. Seitz and C.R. Dyer, "Photorealistic Scene Reconstruction bhy Voxel Coloring," In *Proc. CVPR*, pp.1067-1073, 2002.

[11] A. Fitzgibbon, Y. Wexler, and A. Zisserman, "Image-based rendering using image-based priors," In *Proc. ICCV*, Vol.2, 2003.

[12] 2d3 Ltd. http://www.2d3.com, 2002.