Inverse Volume Rendering Approach to 3D Reconstruction from Multiple Images

Shuntaro Yamazaki, Masaaki Mochimaru, and Takeo Kanade

Digital Human Research Center, AIST,
Water Front 3F, 2-41-6, Aomi, Koto-ku, Tokyo 135-0064, Japan
{shun-yamazaki, m-mochimaru, t.kanade}@aist.go.jp
http://www.dh.aist.go.jp

Abstract. This paper presents a method of image-based 3D modeling for intricately-shaped objects, such as a fur, tree leaves and human hair. We formulate the imaging process of these small geometric structures as volume rendering followed by image matting, and prove that the inverse problem can be solved by reducing the nonlinear equations to a large linear system. This estimation, which we call *inverse volume rendering*, can be performed efficiently through expectation maximization method, even when the linear system is under-constrained owing to data sparseness. We reconstruct object shape by a set of coarse voxels that can model the spatial occupancy inside each voxel. Experimental results show that intricately-shaped objects can successfully be modeled by our proposed method, and the original and other novel view-images of the objects can be synthesized by forward volume rendering.

1 Introduction

Reconstruction of 3D scene information from multiple view-images is a major research topic in computer vision. Most of the existing methods of scene reconstruction attempt to create a model of the object as a solid, using boundary representation. Many real objects, however, have extremely intricate shapes, such as human hair and fur, on the surface. It is therefore difficult to represent their geometry using boundary-based representation.

It is difficult to model intricately-shaped objects for two reasons. Firstly, the boundary-based shape representation is not suitable for such objects as human hair. Secondly, the resolution of optical sensors, such as Charge-Coupled Devices (CCD), is usually much lower than that of object geometry. Hence, it is inherently impossible to reconstruct complete geometry from given images.

Although it is difficult to capture and reconstruct intricate shape on the object surface, the captured image can preserve the appearance of these objects in sufficient quality. This fact implies that photorealistic view can be synthesized from a reconstructed model even if the resolution of the model is not as high as that of object shape.

In this paper, we propose a method of volumetric scene reconstruction using the voxels that model the spatial occupancy and color of the object. In practice, the spatial occupancy is stored in the voxel as α value (opacity), and the synthetic view images are generated through conventional volume rendering techniques.

2   Related Work

2.1   Multi-view Reconstruction

The research on the method of 3D scene reconstruction from multiple view-images has a rich history. Here we briefly describe the related work.

One of the first attempts for image-based modeling of 3D scene in computer vision is two-view stereo reconstruction [1]. Okutomi et al. [2] extended the conventional two-view stereo reconstruction into the multiple-view problem and achieved convincing results. Kang et al. [3] discussed a method of multi-view stereo reconstruction from images with large occlusions. These methods are designed to reconstruct depth maps from particular viewpoints. Hence, they are not suitable for full 3D scene reconstruction from images obtained from multiple surrounding cameras.

Visual hull reconstruction [4] is another approach to 3D scene reconstruction from multiple view-images. The algorithm does not need to solve the correspondence problem. Instead, it simply calculates the convex hull of silhouettes in all view images. While the visual hull method works robustly when cameras surround the object, a concave object cannot be reconstructed using silhouettes alone. This problem was solved by Seitz et al. [5] in the voxel coloring method. The original voxel coloring has a limitation on the location of input view images, which is overcome in the space carving method proposed by Kutulakos et al. [6] The opacity hull [7] method proposed by Matusik et al. is another approach to this problem. They simply use a visual hull model as an rough geometric proxy, and map opacity images using view-dependent texture mapping. This method avoids the difficulty in geometric reconstruction, but requires a lot of input images to achieve photorealistic rendering.

Our proposed method is inspired by the space carving method, but has been extended so that it can deal with intricate shape within a framework of voxel modeling. Specifically, the geometrical structure within a voxel is represented as its spatial occupancy. Our method is similar to Roxels method [8] in that both can reconstruct spatial occupancy/opacity of voxels from images. The convergence of the Roxels method, however, is not proven, and the method cannot reconstruct the voxels in high resolution owing to the high computational cost.

2.2   Alpha Estimation

When a scene is captured by the digital optical sensors, such as CCD, what the device can record is not the light energy of a single light ray, but the averaged radiance incoming from a finite space in the scene. If a part of an object is observable from a single device cell, the recorded radiance is the combination of radiance coming from the corresponding foreground and background.

Alpha estimation is the process that decomposes an RGB color C_p of each image pixel into three components: foreground color F_p, background color B_p, and foreground opacity A_p. The relationship between the variables can be described by *matting equation*.

$$C_p = F_p + (1 - A_p)B_p. \tag{1}$$

The foreground opacity, or simply, opacity $A_p \in [0, 1]$ represents the contribution of the foreground object color to the pixel. When the background color in the images is controllable, we can separate these three components perfectly by capturing two images with different background colors and computing F_p and A_p which are common in the images [9]. The natural image matting methods [10, 11, 12] can solve this problem even when the background cannot be controlled and only one image is available.

3 Inverse Volume Rendering

3.1 Assumption and Preprocess

In this paper, the scene is assumed to be static and the surface reflection follows the Lambertian law. Under this assumption, the radiance emanating from the scene can be observed as a single color. Our proposed method deals with the scene composed of foreground object O, which we are interested in, and background $\mathcal{B}$, which should be removed in the modeling process. The background component is removed beforehand by an appropriate method of alpha estimation described in Section 2.2.

The input to our modeling algorithm is a set of color images taken at N_{view} different viewpoints. The accuracy of reconstruction and robustness to noise and other factors not modeled in our assumption can be increased by using as many images as possible. About 30 images uniformly distributed on the upper hemisphere surrounding O could achieve good reconstruction in our experiments. Both intrinsic and extrinsic camera parameters are supposed to be known. The output from our algorithm is a set of voxels v_i which has both RGB color c_i and occupancy α_i.

3.2 Volume Rendering Equation

When the interest object O is captured and digitized into images by CCD, the intensity of pixel color C_p is in proportion to the sum of radiance emanating from the surfaces within a frustum spanning between the scene and the device cell. At each depth along viewing rays in the frustum, the transferred radiance is the sum of those emanating at the point ($\mathcal{S}_f$) and those coming from the behind ($\mathcal{S}_b$). Hence, the pixel colors in input images can be described by the following *volume rendering equation* [13].

$$C_p = \sum_{i \in \{\text{along a viewing ray}\}} c_i \alpha_i \prod_{j=0}^{i}(1 - \alpha_j), \tag{2}$$

c_i represents the radiance coming from light sources and reflected on the scene object at the depth i along viewing rays. $\alpha_i \in [0, 1]$ is the ratio by which the object located at i occludes others behind it. Thus, this ratio α can be regarded as equivalent to the spatial occupancy of the foreground object at the location. Supposing that c_i and α_i are view-independent, they can be parameterized by the location in 3D space. We voxelize the 3D space and assign c_i and α_i to each voxel.

3.3 Matting Equation

First the background components observed in input images are removed from each image pixel C_p, which is combination of radiance transferred from the foreground object

and the background scene. This process is essential in order to associate each voxel's values c_i and α_i with pixel intensity C_p to create a 3D model of the object from the observed images.

Suppose that a voxel grid spreads infinitely in 3D space. Then, we can define the mapping between the voxel coordinate (x, y, z) and the 1D coordinate s along viewing rays. For each viewing ray that goes through the foreground object O, there is a position that separates O and background $\mathcal{B}$. Dividing the 1D ray coordinate into the *front* and *back* parts, equation (2) is rewritten in

$$C_p = F_p + (1 - A_p)B_p, \tag{3}$$

where

$$A_p = 1 - \prod_{k \in front} (1 - \alpha_k) \tag{4}$$

$$F_p = \sum_{i \in front} c_i \alpha_i \prod_{j=0}^{i} (1 - \alpha_j) \tag{5}$$

$$B_p = \sum_{i \in back} c_i \alpha_i \prod_{j \in back}^{i} (1 - \alpha_j). \tag{6}$$

Intuitively, A_p is the contribution of the spatial occupancy of voxels along a viewing ray to an image pixel, F_p is the contribution of accumulated colors of the voxels, and B_p is the background color.

Compared with equation (1), it turns out that equation (3) is equivalent to the matting equation. The values A_p and F_p can be estimated from C_p before modeling voxels, by one of several methods of alpha estimation introduced in Section 2.2. Once the foreground components A_p and F_p in input images is associated with the voxel values c_i and α_i, we can estimate voxel values using equation (4) and equation (5) as constraints. We refer to the estimation of 3D voxel values from 3D pixels values composed of foreground components as *inverse volume rendering*.

3.4   Derivation of Constraints

We reconstruct the color c_i and spatial occupancy α_i of voxels from the accumulated color F_p and occupancy A_p of image pixels in the following two-step procedure.

In the first step, we reconstruct only the voxel occupancy α_i using foreground opacity A_p. Taking the logarithm of equation (4) for each $A_p \neq 1$ and replacing opacity with transparency as $T_p = 1 - A_p$ and $t_i = 1 - \alpha_i$, we obtain the following equation.

$$\log(T_p) = \sum_i \log(t_i) \tag{7}$$

Since $\log(T_p)$ have already been estimated in preprocess, and therefore are regarded as constants, equation (7) comes down to a simple linear system in which $\log(t_i)$ are unknowns.

In the second step, we then reconstruct voxel color c_i from foreground color F_p. Now, the spatial occupancy α_i has been reconstructed in the first step. Thus, equation (4) can be reduced again into a linear system

$$F_p = \sum_{i \in front} \left[\alpha_i \prod_{j=0}^{i} (1 - \alpha_j) \right] c_i \qquad (8)$$

where $[\ldots]$ and F_p are constants, and c_i are the unknowns that we want to estimate.

4   Implementation

4.1   Iterative Back-Projection Based on EM

Various methods of solving linear systems have been proposed. When the coefficient matrix of the linear system is full-rank, we can solve the system either by using a direct method such as the Gauss-Jordan elimination, or an iterative method such as the conjugate gradient method. If the system is either under- or over-constrained, the solution that maximizes the certain likelihood measure is estimated, for instance, by singular value decomposition.

Our linear system, however, cannot be solved directly by these conventional methods owing to the gigantic size of the system. The number of unknowns in equation (7) and equation (8) is equivalent to the number of voxels that increases in a cubic order. On the other hand, the number of equations in the linear system is roughly equal to the number of image pixels. Thus, the computational cost of our problem can be extremely high. It is also the case that the coefficient matrix cannot be stored in the limited working memory of a standard computer.

In order to overcome these difficulties, we propose an algorithm that can solve such a gigantic linear system within a framework of the EM (Expectation Maximization) method [14]. This algorithm starts with an initial estimation of the solution, and iteratively improves the solution through the maximization of an objective function. The algorithm can improve the solution monotonically, and can reach the global optimum.

The EM estimation is composed of two steps, namely, E-step and M-step. In the E-step, the expectation of certain probabilistic phenomena is calculated using the current estimation of parameters. In the M-step, the parameters are modified so that the expectation is maximized. Repeating E-step and M-step alternatively can maximize the expectation function even when some parameters cannot be measured directly.

The parameters that we want to estimate are the color c_i and occupancy α_i of voxels. The observed data that we have is F_p and A_p. In the E-step of the inverse volume rendering, we simply perform forward projection of voxel values. This is equivalent to the volume rendering according to equation (2). In the M-step of the inverse volume rendering, we improve either the color or the occupancy of voxels using back-projection for each viewing ray. The expectation can be calculated as a linear combination of unknowns. Hence, the function is concave and has a single global optimum.

Let the n-th estimations of unknowns in a linear system be $\{x_j^{(n)}\}$, the coefficients of the system be $\{r_{i,j}\}$, the constants of the system be $\{c_i\}$, then the $n + 1$-th estimation of unknowns can be obtained by

$$x_j^{(n+1)} = \frac{x_j^{(n)}}{\sum_i r_{i,j}} \sum_i \frac{r_{i,j} c_i}{\sum_j r_{i,j} x_j^{(n)}} \tag{9}$$

where $\sum_j r_{i,j} x_j^{(n)}$ is the result of forward projection in the E-step, and the summation of projections with regard to $r_{i,j} c_i$ corresponds to the result of back projection. This relationship is illustrated in Fig. 1. It is worth noting that this EM estimation is a generic framework for solving linear systems.

The EM algorithm in our estimation can be accelerated by dividing the problem into several subsets. First, we divide the set of input images into several subsets. Then, the linear system is solved using one of the subsets. Once the algorithm has been converged, the linear system is solved using another subset using the previous solution as an initial estimation. This scheme is called OSEM (Ordered Subset EM) [15]. In our

Fig. 1. Interpretation of the update law in EM

experiments, we made subsets by choosing four images such that the distances between their viewpoints are as large as possible.

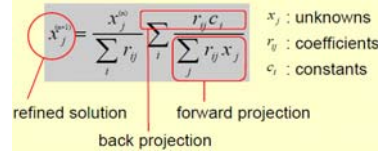### 4.2   Shell Voxels

The voxels in which no foreground object exists ($\alpha_i = 0$) do not affect the estimation of other voxels along the viewing rays that pass through the empty voxel. Similarly, the voxels that are completely occupied by foreground object ($\alpha = 1$) neither affect the estimation of the voxels along the viewing rays. We can reduce computational cost by just omitting the computation for these rays.

After the alpha estimation for each input image has been completed, the voxels are classified into three types according to the opacities of corresponding image pixels.

1. background voxel: $A_p = 0$ in at least one image
2. internal voxel: $A_p = 1$ in all images
3. shell voxel: otherwise

The classification of voxels is performed as follows. Firstly, we classify as background the voxels whose projection is completely transparent (the corresponding pixels are all $A_p = 0$) at least in one of input images. Secondly, we construct the visual hull [4] of completely opaque pixels ($A_p = 1$), and classify the voxels enclosed by the hull as internal. The rest are shell voxels.

4.3   Optimization of Voxel Traversal

In each EM estimation, a set of coefficients in the left hand of equation (7) has to be prepared. This calculation requires the voxel traversal along arbitrary viewing rays and therefore is computationally expensive. Therefore, we precompute the set of voxels along every viewing ray beforehand, and store the result into LDI structures [16] for each input pixel. We can omit the LDI entry for the pixels where $A_p = 0$ and $A_p = 1$.
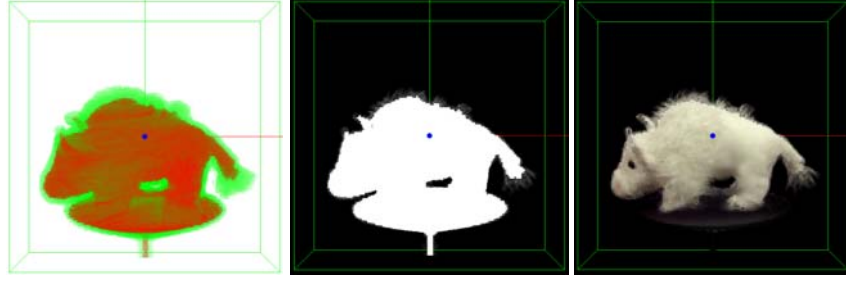
Fig. 2. Example of voxel reconstruction. (left) shell voxels in green and internal voxels in red, (center) reconstructed spatial occupancy, and (right) reconstructed color distribution.

5 Experimental Results

We have implemented the proposed method of inverse volume rendering and conducted some experiments on a standard PC with Pentium4 3.4GHz CPU and 2G byte main memory. Input images are captured from 36 viewpoints around the object that we want to model. The size of input images are 320×240, and the voxel resolution is set to 64^3 and 128^3.

5.1 Alpha Estimation

We adopted the multi-background scheme proposed by Smith et al.[9] for alpha estimation from input images. The background color of images is controlled by a liquid crystal projector. For each viewpoint, two images with different background color, C_{k1} and C_{k2} in RGB color space, were taken. Let the observed image color at the same pixel be C_{m1} and C_{m2} respectively, then the foreground opacity A_p of the pixel can be estimated by the following equation.

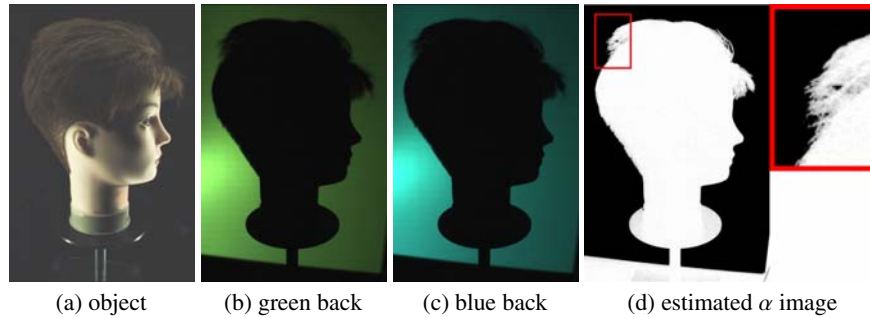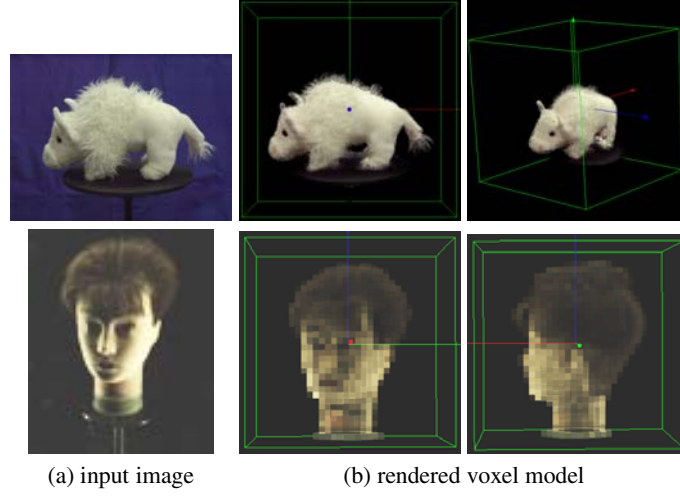$$A_p = 1 - \frac{(C_{m1} - C_{m2}) \bullet (C_{k1} - C_{k2})}{(C_{k1} - C_{k2}) \bullet (C_{k1} - C_{k2})} \tag{10}$$

(a) object   (b) green back   (c) blue back   (d) estimated α image

Fig. 3. Multi-background matting

|(a) input image|(b) rendered voxel model|

Fig. 4. Results of volume rendering

where an operator ($\bullet$) represents dot-product of RGB vectors. The foreground color F_p of the pixel is calculated as follows.

$$F_p = (C_{m1} + C_{m2} - (1 - A_p)(C_{k1} + C_{k2}))/2 \tag{11}$$

An example of input images in alpha estimation and obtained alpha image is shown in Fig. 3.

5.2   Results of Volume Rendering

The reconstructed voxel model is rendered in Fig. 4. The voxels are rendered using the volume rendering equation (equation (2)) with the viewpoints not included in the input images.

5.3   Convergence

In Fig. 6, the convergence of our proposed method is illustrated. The upper and lower rows show the process of estimating α_i and c_i respectively. The resolution of reconstructed voxels is 128^3. The figure indicates that the visually sufficient result can be obtained in 10 iterations.

Fig. 5 is the plot of reprojection error in the EM estimation for the data shown in Fig. 6. The lines indicate the decreases in error for two different voxel resolutions. The error decreases rapidly within 5 iterations, and then gradually converges into the minimum value.
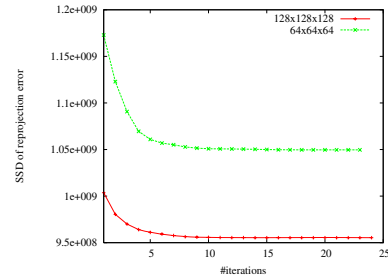
Fig. 5. Convergence
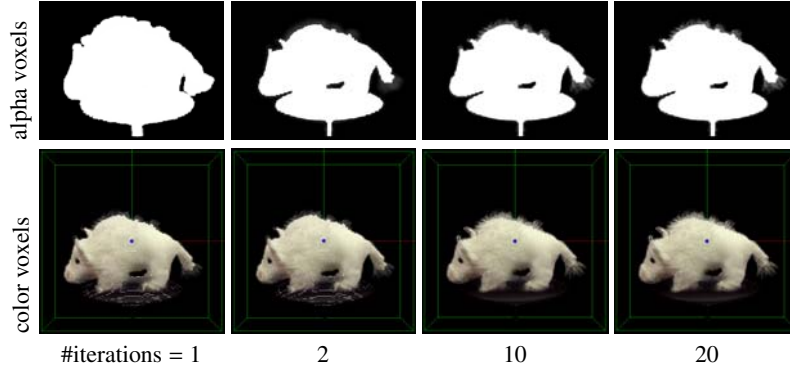
Fig. 6. Iterative optimization

5.4   Computational Cost

Table 1 shows the figures of memory usage and computational time. We have recorded these figures in the experiments using voxel resolutions of 64^3 and 128^3. Owing to the limitation of computer hardware, we could not conduct experiments with larger resolution. For example, reconstruction of a mannequin object with the resolution of 128^3 failed because of limited memory space on 32 bit computer.

Table 1. Performance

object	#shell vxl		memory(byte)		time(min)	
#voxels	64^3	128^3	64^3	128^3	64^3	128^3
cow	23379	185013	~256M	~2.1G	~30	~186
mannequin	32531	254123	~510M	—	~45	—

6   Discussion and Future Work

In this paper we proposed a novel method of voxel reconstruction that can deal with an object with intricate shapes such as a fur and hairs. We formulate our reconstruction process as the *inverse volume rendering* problem, and show how to solve it. We also present an effective implementation and conduct experiments on real objects to demonstrate the usefulness of the proposed method.

In Fig. 4, we see artifacts in the fur where the spatial occupancy seems higher than the real value. The reason for this is that computing $\log(1 - A_p)$ and $\log(1 - \alpha_i)$ become erroneous when α_i and A_p is close to 1, and therefore the small errors in alpha estimation for input image drastically affect the estimation of voxel occupancy.

We implemented some measures to reduce the computational costs in the inverse volume rendering. However, the cost is still high, and therefore we cannot reconstruct the object in a proper spatial resolution. We are planning to adopt adaptive voxel structures, such as octree and k-d tree, and to extend our algorithm so that it can be executed on parallel computers.

References

1. Marr, D.C., Poggio, T.: A computational theory of human stereo vision. Proceedings of the Royal Society of London **B 204** (1979) 301–328
2. Okutomi, M., Kanade, T.: A multiple-baseline stereo. IEEE Transactions on Pattern Analysis and Machine Intelligence **15** (1993) 353–363
3. Kang, S.B., Szeliski, R., Chai, J.: Handling occlusions in dense multi-view stereo. In: Proc. Computer Vision and Pattern Recognition 2001. (2001) I:103–110
4. Laurentini, A.: The visual hull concept for silhouette-based image understanding. IEEE Transactions on Pattern Analysis and Machine Intelligence **16** (1994) 150–162
5. Seitz, S.M., Dyer, C.M.: Photorealistic scene reconstrcution by voxel coloring. In: Proc. Computer Vision and Pattern Recognition '97. (1997) 1067–1073
6. Kutulakos, K.N., Seitz, S.M.: A theory of shape by space carving. In: Proc. International Conference on Computer Vision '99. (1999) 307–314
7. Matusik, W., Pfister, H., Ngan, A., Beardsley, P., Ziegler, R., McMillan, L.: Image-based 3D photography using opacity hulls. In: Proc. SIGGRAPH 2002. (2002) 427–437
8. Bonet, J.S.D., Viola, P.A.: Roxels: Responsibility weighted 3d volume reconstruction. In: Proc. International Conference on Computer Vision '99. (1999) 418–425
9. Smith, A.R., Blinn, J.F.: Blue screen matting. In: Proc. SIGGRAPH '96. (1996) 259–268
10. Ruzon, M.A., Tomasi, C.: Alpha estimation in natural images. In: Proc. Computer Vision and Pattern Recognition 2000. (2000) 24–31
11. Chuang, Y.Y., Curless, B., Salesin, D.H., Szeliski, R.: A bayesian approach to digital matting. In: Proc. Computer Vision and Pattern Recognition 2001. Volume 2. (2001) 264–271
12. Sun, J., Jia, J., Tang, C.K., Shum, H.Y.: Poisson matting. ACM Transactions on Graphics **23** (2004) 315–321
13. Lacroute, P., Levoy, M.: Fast volume rendering using a shear-warp factorization of the viewing transformation. In: Proc. SIGGRAPH '94. (1994) 451–458
14. Lange, K., Carson, R.: EM reconstruction algorithms for emission and transmission tomography. Journalof Computer Assisted Tomography (1984) 306–316
15. Hudson, H.M., Larkin, R.S.: Accelerated image reconstruction using ordered subsets of projection data. IEEE Transactions on Medical Imaging **13** (1994) 601–609
16. Shade, J., Gortler, S., wei He, L., Szeliski, R.: Layered depth images. In: Proc. SIGGRAPH '98. (1998) 231–242