Sparse IBR Using Range Space Rendering

Dan Kong
Department of Computer Engineering
University of California,Santa Cruz
kongdan@soe.ucsc.edu

Hai Tao
Department of Computer Engineering
University of California,Santa Cruz
tao@soe.ucsc.edu

Hector H. Gonzalez-Banos
Senior Computer Scientist
Honda R&D Americans, Inc.
hgonzalez@honda.hra.com

Abstract

In this paper, we propose a new solution to the sparse image-based rendering (IBR) problem. Given two images taken from different viewpoints, our algorithm can accurately generate images from new viewpoints in between the original two views. This paper contributes to the sparse IBR in the following aspects: (1) Direct range space matching and multiple depth maps rendering. (2) Good rendering can be achieved with incorrect depth values, especially those textureless regions.(3) An efficient coarse-to-fine strategy to accelerate rendering and avoid problems caused by repetitive patterns. (4) A weak scene consistency constraint to simultaneously refine the rendering results and detect the occlusions. The proposed algorithm has been implemented and tested on real images and promising experimental results are demonstrated in the paper.

1 Introduction

A key problem in Image-based rendering is how to represent the scene. Many methods have been proposed in the past few years to address this problem. Based on the geometric information used, these techniques can be classified into three categories [16]. Light Field [9] and Lumigraph [7] can be regarded as dense IBR approaches where no geometry information is needed. However, such an IBR system requires special capturing equipment and the image database is very large even for small objects. Another type of IBR techniques represents the scene geometry explicitly, either in the form of depth maps or 3D meshes. There are many methods fall into this category [5, 10, 15]. In between these two extreme approaches, there is another class of techniques that only requires the correspondence information across views. The rendering of the new views is based on these correspondences. View interpolation [4] and view morphing [13] are two representative techniques in this category. However, the establishment of the correspondence is also difficult, in particular for real images. The trade-off between the number of input images and the amount of geometric information is first systematically discussed in the plenoptic sampling paper by Chai et al. [3]. In this paper, we attempt to address the following problem: is there a way to synthesize virtual views from a set of sparsely sampled images without explicitly recovering a consistent and accurate depth representation.

Several methods have been proposed recently addressing the sparse IBR problem. Most of these algorithms depend on a single depth representation recovered using stereo algorithms. A good survey of early stereo algorithms can be found in [6]. Recently, several new scene representations have been proposed to solve the stereo problem. A Layered depth [1] representation describes the scene as a collection of planar layers. In [17] Szeliski proposed to associate a depth map with each key input image to facilitate new view generation. Volumetric method is another interesting approach that tries to reconstruct the scene in the global 3D coordinate system. In this approach, the scene is represented as a 3D volume. By enforcing the color consistency constraint, methods such as space carving [8] and voxel coloring [14] are developed to reconstruct the scene from sparse distributed cameras and to generate the new views. Recently a range space based depth recovery algorithm was proposed by Ng [11], in which a volumetric depth representation is recovered in each new view. In addition, in the past several years, global optimization algorithms have also been proposed to improve the depth recovery algorithms [2, 12, 18].

Based on the depth recovery results, the input images can be warped to the new viewpoints to generate the images. One problem with depth based sparse IBR is that it requires very accurate depth estimation. Errors in depth maps often cause annoying visual artifacts such as spikes or holes in the rendered images.

In this paper, based on the observation that image-based rendering is an easier problem compared to accurate depth computation, we propose a direct sparse IBR method called range space rendering. In this method, the depth information is implicitly determined in each new view through coarse-to-fine range space matching. By associating a depth map at each new view instead of a single depth map for the reference image, we can avoid the difficulty of recovering accurate depth. We also show that good rendering results can be achieved without accurate depth information. This is most noticeable for the smooth and textureless areas. Finally, we also integrate a weak scene consistency constraint into our algorithm to improve the visual quality of the rendered images and detect the occluded regions.

The rest of this paper is organized as follows. Section 2 explains the main ideas of the proposed approach. Section 3 describes the implementation details. In Section 4, some experimental results on real image data are demonstrated. Discussions and conclusions can be found in Section 5

2 The Approach

2.1 Range Space Matching

As for image-based rendering, some interesting questions are: Do we really need a accurate depth map to synthesis a novel view? What are the benefits of recovering depth map for each new view in range space? Do all the depth maps across each view need to be consistent? These questions are discussed in this section

2.1.1 The Basic Idea

The basic idea of this approach is illustrated in Figure 1. A 3D range space is discretized to constant depth layers at each new viewpoint. For two view cases, we can rectify the two images to obtain a standard stereo pair. The rectification process will greatly simplify the range space matching algorithm if the new viewpoints are on the baseline. For such configuration, the depth steps in the range space correspond to sub-pixel shifts along the horizontal epipolar line in the image space. For each depth layer, the transformation between the original image and the new view can be described using a 2D homography. We use two homographies H_l and H_r to warp the left and right views to the new view. Matching scores are computed for each depth layer based on the two warped images. The matching scores of all depth layers form a 3D matching volume. A unique layer that gives the best match is picked for each pixel in the new view.

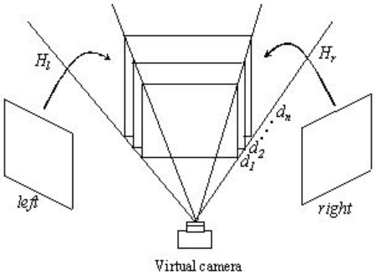

Figure 1: The 3D range space representation.

2.1.2 Multiple Depth Maps Versus Single Depth Maps

Traditional depth-augmented IBR approach synthesis virtual views by first recovering a single depth map for one of the reference images, and then warp the reference image to the novel viewpoint based on the depth. There are two main problems with this approach. First, it is hard to recover an accurate depth map. Second, the depth-based image warping may cause holes and spikes in the new views.

In our approach, by matching two images at each new viewpoint, we implicitly associate a depth values for every pixel in the new view. Thus, we can avoid the holes and spikes in the new view. Another advantage of computing multiple depth maps is more scene information can be acquired to improve rendering result. Basically, a 3D matching volume is constructed at each new viewpoint. If we do not consider the arrangement of each cell in this volume, we can regard all the volumes as the same and the only difference is to search along different direction to find a best match. Thus, we can represent the 3D matching volume in the left view. This is illustrated in Figure 2. Suppose the two input images have been rectified, each cell $(x+d,y)$ in the matching volume will correspond to pixel (x,y) in the left image and $(x+d)$ in the right image. Since this matching volume is represented in the left view, the search area for a pixel in the left image is always along

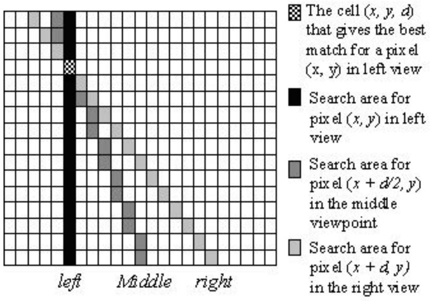
▩ The cell (x, y, d) that gives the best match for a pixel (x, y) in left view
■ Search area for pixel (x, y) in left view
▨ Search area for pixel (x + d/2, y) in the middle viewpoint
□ Search area for pixel (x + d, y) in the right view
left
Middle
right

Figure 2: A 2D slice of the 3D matching volume with y held constant.

vertical direction. However, for pixels in the right view and other viewpoints in between, the search direction is different. Thus, for a surface point, a depth is computed differently in each view. If we insist that all the depth corresponding to the surface point in each view should be the same, we enforce a strict scene consistency in our approach. If we are concerned with 3D reconstruction, this should be fine. But as we will see next, for image-based rendering, benefits can be gained by computing multiple depth maps.

2.1.3 Smooth Surface And Occluded Regions

Smooth surface will cause matching ambiguity in stereo algorithm. As a result, the recovered depth map is very noisy, making the rendered new views visually uncomfortable. In our approach, the rendering results are not affected by the textureless areas even the depth is not correctly computed. This can be seen from Figure 3(a).

Suppose the scene object is s. P is a surface point that is visible both in left and right views. In the virtual camera position, considering the virtual ray r that hits the surface point p. The depth information along r can be determined by analyzing the matching volume. As shown in Figure 3(a), ideally, r has the best match in depth layer d_i. However, this may not be true for untextured regions. Suppose the surface between the points a and r has the same surface color. When the two images are warped to the depth layer d_j, they will also match well because the surface point a and r have the same color. As a result, we will incorrectly assign depth d_j to virtual ray r. From this example, we observed that correct color is indeed selected, even with the wrong depth value. Though flat areas usually cause serious problems in depth recovery algorithms, in image-based rendering, since our only concern is whether the correct color is picked for each virtual ray, incorrect depth recovery may not cause any rendering problems.

By computing multiple depth maps at new viewpoint, occluded regions may also be detected. This is illustrated in Figure 3(b). Since surface point p is occluded in the right view, spurious but relatively good matches will assign incorrect depth to virtual ray that hits p. For example, at new viewpoint i, ray r_1 and r_2 intersect at a, giving the best match score along virtual ray r_i. While at new viewpoint j, the best match is found at depth layer d_j , which is the intersection of ray r_2 and r_3. Thus, for a same scene point, inconsistent depth in computed at each new view. As we will show in 2.3, we can utilize this inconsistency to refine the rendering results and detect the occluded regions.

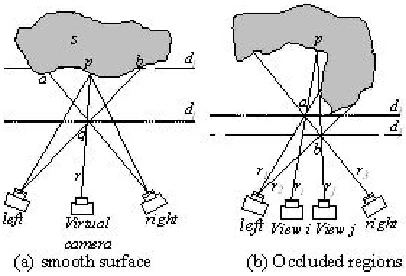

Figure 3: Smooth surface and occluded regions.

2.2 Coarse-to-Fine Scheme

The proposed algorithm is computational expensive for image pairs with large disparities, making it impractical in real applications. Another problem with the direct implementation of this algorithm is the repetitive pattern, which is illustrated in Figure 4.

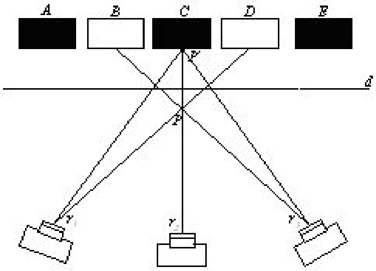

Figure 4: Repetitive Pattern

Suppose the scene contains the repetitive pattern A-B-C-D-E. The correct match for a virtual ray r_2 at the new viewpoint should be at p'. However, because of the repetitive pattern, ray r_1 and r_3 intersect at p also give a high match score because the B and D are the same pattern. If the match score at p is higher than the match score at p', incorrect color will be picked for virtual ray r_2. If we apply our algorithm directly, this problem is inevitable since the match and search is done at each layer and a winner is picked for each pixel. However, if we have a rough estimate of the layer for each ray, for example, layer d for ray r_2 in Figure 4, we can restrict our search within a range from that layer, which will help to solve this problem.

A coarse-to-fine scheme is proposed to address both the computational and repetitive pattern problems. Basically, we construct the pyramid from two input images. The algorithm is first applied to the low-level of the pyramid to get an initial depth for each virtual pixel. The initial depth is then propagated from the low-level to the high-level. The same algorithm is carried on at the high-level, but the match and search are restricted within a range from the depth layer estimated from the low-level. Figure 5 compares the result before and after using coarse-to-fine scheme, we can see that the repetitive pattern appeared near the ladder is eliminated after the coarse-to-fine method is used. More experimental results can be found in section 4.

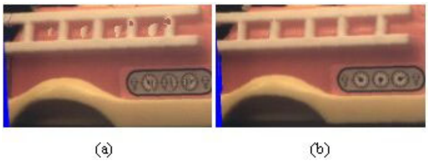

Figure 5: The virtual view rendered at the middle viewpoint. (a) Direct implementation of the approach. (b) Using coarse-to-fine scheme.

2.3 Local Depth Voting And Occlusion Detection

In order to eliminate the false matching and get better rendering result, we further propose a method to refine the local depth map for each new view. This method is based on a weak constraint called *scene consistency*. As discussed in section 2.1, by using rang space matching, the depth computed for smooth area and occluded regions may not be correct and consistent across each view, but for other scene point that is visible in each view, we assume they should be in the same 3D position when seen from different viewpoint. We use a local depth voting strategy to enforce this weak consistency constraint. Basically, for every pixel (x,y) at each new view, the corresponding depth information at other viewpoint is collected. A function *LocalVote(x, y)* is applied to these values to check the scene consistency. We modify the depth at that pixel to the voting result if the return value of this function is not empty, which means that the scene consistency is satisfied for those pixels. Otherwise, we keep the original depth for that pixel. The scene consistency is a constraint that we add to our algorithm, which is similar to the radiance constraint enforced in the space carvin [8]. In reality, the scene consistency constraint is more reasonable since radiance constraint assume the Lambertian model. Another problem we can simultaneously solve by using the local depth voting is occlusion detection. Here, we combine the criterion used in the cooperative stereo [20] with our local voting result to explicitly detect the occluded regions in the novel view. Thus, our occluding mask function is defined as:

$$H(i,j) = \begin{cases} 1 & \mathit{Match(i,j)} < \mathit{threshold} \; \&\& \; \mathit{LocalVote(i,j)} == \mathit{Empty} \\ 0 & \mathit{else} \end{cases}$$

Once the occluded regions at each new view are detected, we can use an iterative hole filling method to determine the depth for each pixel that is marked as hole.

2.4 Rendering

Once the local depth map is recovered for each new viewpoint, the rendering is just performed by weighted combination of corresponding pixel of the original views according to the depth. The weights are a function of the angle between the novel viewpoint and the original viewpoint. To handle occlusion, the z-buffer algorithm is used to record the visibility information for each new view pixel in the original image. Based on the visibility map for each view, we could correctly pick the pixels from the input images for the occluded region.

3 Implementation

The proposed algorithm can be applied to any image pair taken with a hand-held camera when the overlapping areas between the two images are big enough. Some implementation details are discussed in this section.

3.1 Preprocessing

Currently, we only consider the case which the input is two views. To make the implementation easier, we use two rectified images so that the warping and range space matching are totally operated along the scanline. Such a preprocessing is similar to pre-warp stage in view morphing [14]. To rectify the two images, we need to know the camera intrinsic and extrinsic parameters. In our implementation, we use Zhang's algorithms [19] which use the planar calibration object to compute the camera matrix. Since the algorithm also compute the relative 3D pose of each camera with respect to the calibration object, it is possible to use the same algorithm to compute the relative poses between cameras. In a real-time IBR system, the camera parameters should be obtained using the self-calibration method. Thus, the whole procedure of this algorithm can be fully automatic, which is a requirement for real-time application.

3.2 Image Warping and Computing Of Match Score

The range space is divided into several layers. The number of the layers is proportional to disparity between the two views. In our implementation, we set the layer number equal to the difference of the maximal and minimal disparity of the two images. Since the two images are rectified in the preprocessing, the warping is simply a shift along the epipolar line. To avoid holes and folds in the warped image, we use backward warping, and bilinear resampling along the scanline. The matching is a process similar to the stereo vision. The similarity of the two corresponding image pixels is measured by the matching cost which can be defined as absolute squared intensity difference (SAD) or sum of squared difference (SSD). Alternatively, matching cost can also be defined as correlation. In our implementation, we use the SSD to compute the matching cost. A 7 by 7 window is used to aggregate the local evidence. After the matching process, we use the winner-take-all principle to select the best layer.

4 Experimental Results

Based on this algorithm, we implement a prototype IBR system on a PC machine using MATLAB. Currently, our system can only solve the two view problems. However, the discussion in this paper provides us a basis to extend to general multiple camera configurations. The computation time to generate a new view is less than 15 seconds. We believe that a real-time implementation of this method is possible on a modern workstation. Our algorithm is tested on several real image pairs and some results are presented in this section.

We first ran our algorithm on two images of a Tonka toy truck. The transformation between both views was recovered with an eight-point algorithm using 30 manually-selected pairs of image features. The images were then rectified. We use 60 depth layers to discretize the range space at the original image resolution. But only 15 layers were used at the lowest resolution for the course-to-fine method. A new virtual view was rendered at positions 0.4 and 0.6 between the two original camera optical centers. Figure 6 shows the results.

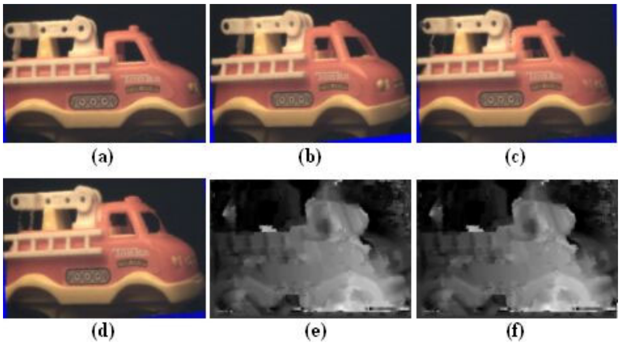

Figure 6: (a)(b): Two input 320x240 RGB images of a toy truck. (c)(d): New views rendered at viewpoint 0.4, 0.6 and (e)(f): The corresponding local depth maps.

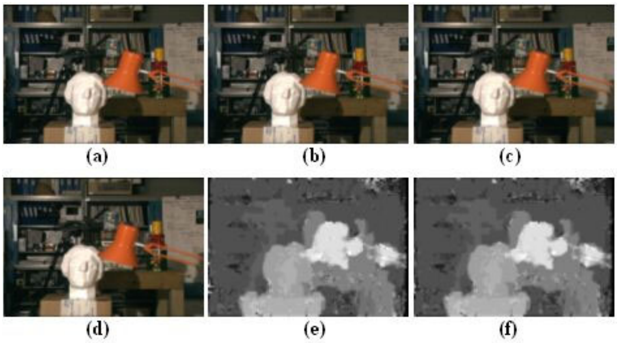

Figure 7: (a)(b):Left and right original images. (c)(d): New views rendered at viewpoint 0.4, 0.6 and (e)(f): The corresponding local depth maps.

Except for the occluded areas, the overall rendering result is very good. This is true not only for textured areas, but also for smooth surfaces, such as the bumper and the body of the truck. As a comparison, we generate the local depth maps for these viewpoints; Notice that good renderings are produced even when the computed depth is not correct. This is a major contribution of our method.

We next apply our algorithm to a benchmark stereo image pair Tsukuba. Since the two images have been rectified, we can directly use them as the input to our algorithm. We use 16 layers for the original image and 4 layers for the low resolution image. The rendering result and the corresponding depth map are shown in Figure 7.

In our third experiment, we apply the range-space method to render the images of a

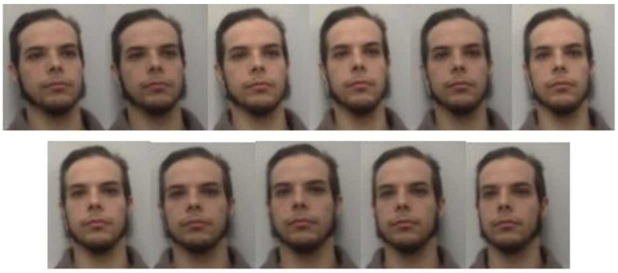

Figure 8: The face sequence generated using our method.

human face. Given two side-views of a human face, a frontal view would be generated which gives the user a sense of eye contact. Such a situation has many applications, especially in the teleconferencing. We use a hand held camera to collect the data. The camera is calibrated and the relative motion is estimated using Zhang's algorithm [19]. Figure 8 shows the rendered sequence from the left view to the right view. Because of occlusion near the ear of this man, we can see some artifacts around the ear. The overall result is very good.

This experiment highlights the advantage of our method over previous stereo-based IBR approaches. The traditional stereo algorithm works well for densely textured scene and suffer in the low texture area and large occlusions. However, we see from this experiment that our method can handle the smooth surface very well and properly solve the occlusion problem

5 Discussions and Conclusions

We have proposed a direct range space method to address the sparse IBR problem. Our algorithm synthesizes a new view at the virtual camera position from two images. The proposed algorithm avoids the scene recovery problem by using range space matching. Our method differs from previous depth-augmented IBR approaches in the following: (1) few depth layers are required. (2) Multiple depth maps for each novel view instead of a single depth map. (3) Images tend to be rendered correctly even if the depth calculations are inaccurate, especially for smooth surface. These properties allow our system to be sparse with potential fast run-times using optimized code.

A coarse-to-fine method was also developed to accelerate the basic algorithm and to eliminate artifacts in the rendered image caused by repetitive patterns in the scene. A weak scene consistency constraint is enforced across each view to refine the rendering results and detect the occlusion in each view. We obtained promising experimental results even with a relatively straightforward implementation.

Along the course of the development of this algorithm, we found several interesting research problems in sparse IBR such as range space representation, rendering without correctly geometry, occlusion detection, and scene consistency. Our future research in sparse IBR will focus on these problems.

References

[1] S. Baker, R. Szeliski, and P. Anandan. A layered approach to stereo reconstruction. In *IEEE Computer Society Conference and Pattern Recognition*, pages 434–441, 1998.

[2] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. In *Seventh International Conference on Computer Vision*, 1999.

[3] J.-X. Chai, X. Tong, S.-C. Chan, and H.-Y. Shum. Plenoptic sampling. In *Computer Graphics (SIGGRAPH'2000)*, 2000.

[4] S. Chen and L. Williams. View interpolation for image synthesis. In *Computer Graphics (SIGGRAPH'93)*, pages 279–288, 1993.

[5] P.E. Debevec, C.J. Talyor, and J. Malik. Modeling and rendering architecture from photographs: A hybrid geometry and image-based approach. In *Computer Graphics (SIGGRAPH'96)*, pages 11–20, 1996.

[6] U.R. Dhond and J.K. Aggarwal. Structure from stereo-a review. *IEEE Transactions on Systems, Man, and Cybernetics*, pages 1489–1510, 1989.

[7] S.J. Gortler, R. Grzeszczuk, R. Szeliski, and M.F. Cohen. The lumigraph. In *Computer Graphics (SIGGRAPH'96)*, 1996.

[8] K.N. Kutulakos and S.M. Seitz. A theory of shape by space carving. *International Journal of Computer Vision*, pages 199–218, 2000.

[9] M. Levoy and P. Hanrahan. Light field rendering. In *Computer Graphics (SIGGRAPH'96)*, 1996.

[10] L. McMillan. *An image-based approach to three-dimensional computer graphics*. PhD thesis, UNC Computer Science, 1999.

[11] K.C. Ng, M. Trivedi, and H. Ishiguro. Generalized multiple baseline stereo and direct virtual view synthesis using range-space search, match and render. *International Journal of Computer Vision*, pages 131–147, 2002.

[12] S. Roy and I.J. Cox. A maximum-flow formulation of the n-camera stereo correspondence problem. In *International Conference on Computer Vision*, 1998.

[13] S. Seitz and C. Dyer. View morphing. In *Computer Graphics (SIGGRAPH'96)*, pages 21–30, 1996.

[14] S. Seitz and C. Dyer. Photorealistic scene reconstruction by voxel coloring. *International Journal of Computer Vision*, pages 151–173, 1999.

[15] J. Shade, S. Gortler, L.-W. He, and R. Szeliski. Layered deph images. In *Computer Graphics (SIGGRAPH'96)*, pages 231–242, 1998.

[16] H. Shum and S.B. Kang. A review of image-based rendering techniques. In *IEEE/SPIE Visual Communications and Image Processing*, pages 2–13, 2000.

[17] R. Szeliski. A multi-view approach to motion and stereo. In *IEEE Computer Society Conference and Pattern Recognition*, 1999.

[18] H. Tao, H.S. Sawheny, and R. Kumar. A global matching framework for stereo computation. In *International Conference on Computer Vision*, 2001.

[19] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transcation on Pattern Analysis and Machine Intelligence*, pages 1330–1334, 2000.

[20] C.L. Zitnick and T. Kanade. A cooperative algorithm for stereo matching and occlusion detection. *IEEE Transcation on Pattern Analysis and Machine Intelligence*, 2000.