IBR-based Free-viewpoint Imaging of a Complex Scene using Few Cameras

Norimichi Ukita, Shohei Kawata, and Masatsugu Kidode
Nara Institute of Science and Technology, Japan
ukita@ieee.org

Abstract

This paper proposes a free-viewpoint imaging method that can be used in a complicated *scene such as an office room by using* sparsely *located cameras. In our method, a free-viewpoint image is generated from multiple image patches obtained by dividing observed images. The quality of the generated image strongly depends on how to divide the observed images. In an incorrect patch in the generated image, the images projected from different cameras differ significantly. With this property, the incorrect patches can be detected. These patches are then re-divided. We demonstrated the effectiveness of our method by generating free-viewpoint images from the real images observed by the cameras in an office room.*

1 Introduction

Free-viewpoint imaging with multiple cameras is useful for a number of real-world applications such as remote monitoring and distance learning. Many studies focus on how to fully represent the 3D appearance of a foreground-moving target object such as a human (see [1], for example). In addition to this technique, background scene representation is also required in order to improve the immersive reality. In terms of cost, the background representation should be generated by few cameras that are located sparsely in order to use a free-viewpoint system daily in commonplace areas.

It is possible to generate a free-viewpoint image of any object, including foreground and background objects, using 3D information (e.g., depth map in [2]). However, it is difficult to acquire sufficient information (e.g., distances in all pixels in [2]) using sparsely located cameras. Instead of such a 3D-based approach, Image-Based Rendering (IBR) with a few cameras is feasible to provide a natural free-viewpoint image. For example, view morphing[3] enables high-quality imaging if a few cameras are located near by. Although IBR-based imaging with sparse cameras for a simple large-scale space has been proposed (e.g., for a stadium[4] and for large objects[5]), these methods are not suitable for reconstructing a *complex scene*. Therefore, we propose a free-viewpoint imaging method that can be used in a *complex scene* by using *sparsely* located cameras.

2 Free-viewpoint Imaging with a Mesh

Our prototype system consists of two fixed cameras spaced 1.5m apart. The images observed by these cameras are shown in Figure 5. Lines and points superimposed on the images depict a *triangle mesh*, consisting of multiple triangle patches. A free-viewpoint image is generated by deforming and integrating the triangle patches of the observed images. The quality of the generated image strongly depends on how to divide the observed images into the image patches.

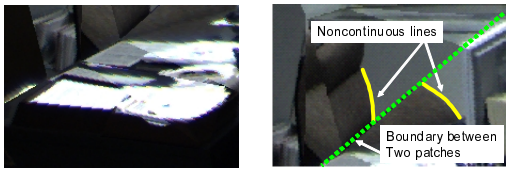
Our method is composed of an offline process for rectifying the triangle mesh and an online process for generating an image using it. We assume that a few correctly corresponding triangle patches in the input images are given in advance. In our experiments, only 15 corresponding points in the images were given manually. They are employed not only for generating an initial triangle mesh but also for estimating the epipolar geometry between the images. The epipolar geometry is used for rectifying the triangle mesh.

The basic scheme of our method is as follows.

1. The internal [6] and external camera parameters are estimated in advance.
2. Several corresponding points are given in two observed images.
3. A pair of triangle meshes in the images is generated from the corresponding points.
4. The xy coordinates of the given points in a free-viewpoint image are obtained via their reconstructed 3D positions.
5. A triangle mesh in the free-viewpoint image is produced from the obtained points.
6. The images inside the patches in the observed images are projected onto the free-viewpoint image using an affine transform.

As mentioned before, the patch configuration greatly affects the quality of a generated image. To confirm this effect, two typical examples are shown in Fig. 1, each of which shows a part of the generated image. Figure 1 (a) seems natural (i.e., no wrong noncontinuous line) because a patch is included within a 3D surface in a scene. The images that are transformed to this region

978-1-4244-2175-6/08/$25.00 ©2008 IEEE



(a) Natural region (b) Incorrect region

Figure 1. Examples in a generated image.

from the observed images are almost the same. In Fig. 1 (b), on the other hand, there are artificial noncontinuous lines. In this case, the images transformed from two camera images are quite different. With this property, an incorrect patch can be detected. The image patch is then re-divided so that the divided patch can generate a natural image. This is a basic scheme in mesh rectification for generating a natural free-viewpoint image.

3 Mesh Rectification

3.1 Increasing points

New points added to an initial mesh should be located so that the following two conditions are satisfied:

C1: Located within unnatural regions.

C2: Located in a vertex of a 3D surface in a scene.

First of all, a triangle patch that satisfies C1 is selected by the following steps:

step-1-1 Select a pair of triangle patches between two images. The selected corresponding patches are transformed to the same shape using an affine transform. In our experiments, they are transformed to an isosceles triangle.

step-1-2 The difference between the two transformed patches is computed.

step-1-3 The difference values of all pixels are binarized using [7]. The difference pixels detected by the binarization process are counted. This number is denoted by N_{dif}.

step-1-4 Let one of the two images, whichever has the larger patch, be I_R. The counted number is divided by the original size of the patch in I_R for normalization. This number is denoted by $\hat{N}_{dif}$.

The above steps are applied to all patches. The patch with the highest $\hat{N}_{dif}$ is regarded as the most unnatural patch. This patch is therefore selected as a patch divided by the next new point. Compared with an algorithm that handles all pixels in an image simultaneously (e.g., [8]), our method is superior for rectifying small errors. Although the method proposed in [9] also evaluates the difference between corresponding transformed patches as does our method, the purpose of [9] is different from that of our method; [9] determines the best connections among given fixed points for generating triangle patches that are consistent with 3D surfaces.

Then, a new point that satisfies C2 is determined in the selected patch by the following steps:

step-2-1 The corresponding patches in the observed images are transformed, subtracted, and binarized by steps 1-1, 1-2, and 1-3 for detecting difference pixels.

step-2-2 The binarized patch is segmented so that neighboring difference pixels are connected and compose a block.

step-2-3 The max-size block is selected and retransformed onto the original-form patch in I_R.

step-2-4 A point is determined within the retransformed block in I_R using the Harris operator [10] in order to extract a point with the largest image-gradient in which a vertex of a 3D surface might be located.

3.2 Point Correspondence

The procedure described in Sec. 3.1 detects a new point P_R only in one image (i.e., I_R). A point corresponding to P_R must be found in the other image. The corresponding point is searched for along the epipolar line in the other image by employing the following evaluation function $F(P_C)$, where P_C is one of the points on the epipolar line $F(P_C) = F_{ncc}(P_C) \times F_{3D}(P_C) \times F_{im}(P_C)$, where $F_{ncc}(P_C)$, $F_{3D}(P_C)$, and $F_{im}(P_C)$ denote the following values, respectively:

$F_{ncc}(P_C)$ is the normalized cross correlation between P_R and P_C.

$F_{3D}(P_C)$ is $\exp(-dis)$ where dis denotes the distance between a 3D surface including the selected patch and a 3D position reconstructed from P_R and P_C using triangulation.

$F_{im}(P_C)$ is $1-\frac{N_{dif}^{new}}{N_{dif}}$, where N_{dif}^{new} denotes the number of difference pixels detected by the steps 1-1, 1-2, and 1-3 in new patches with a new point, P_C[1], if $N_{dif} > N_{dif}^{new}$. Otherwise, $F_{im}(P_C)$ is 0.

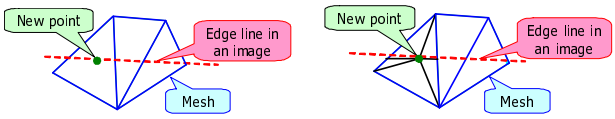
If $F(P_C)$ is the highest and larger than a threshold, P_C is considered to be the point corresponding to P_R.

Although several algorithms have been proposed for matching feature points observed by sparsely located cameras (see [11], for example), $F_{im}(P_C)$ is peculiar to the application addressed in this paper.

Note that wrong corresponding points result in unnatural regions in a generated free-viewpoint image. The above threshold should be, therefore, large in order to avoid wrong correspondence.

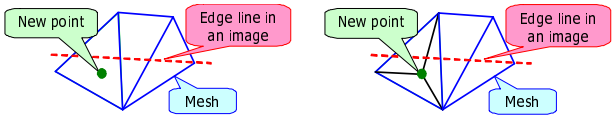
3.3 Increasing Patches

New patches should be generated so that their sides are along edges of a 3D object in a scene. Our



(a) Before rectification (b) After rectification

Figure 2. Rectifying patches for a new point on an edge.



(a) Before rectification (b) After rectification

Figure 3. Rectifying patches for a new point not on any edge.

method has two ways of mesh rectification depending on whether or not a new point is on the edge line.

First of all, the edge image of the observed image is created. If the new point is on any of edge lines, new patches are generated so that the new point becomes one of their vertices and the edge line overlaps one of their sides as illustrated in Fig. 2.

If the new point is not on any edge line, the new patches are generated so that each of them consists of the new point and two of three vertices of the selected patch as illustrated in Fig. 3.

The procedures described in Sec. 3 are repeated until the number of difference pixels detected by steps 1-1, 1-2, and 1-3 in all patches are less than the threshold.

4 Free-viewpoint Image Generation from a Rectified Mesh

4.1 Camera Selection

In our method, each patch in a free-viewpoint image is generated from one of two observed images. In general, the camera that is near the virtual viewpoint should be selected. In some cases, however, a 3D surface visible from the virtual viewpoint is invisible from the selected camera as illustrated in Fig. 4. To avoid this inappropriate camera selection, the geometric configuration of three vertices of a patch is checked as follows. Each 3D point is projected onto the image plane of the virtual viewpoint; the 3D position of each vertex is known because point correspondence between two input cameras is given. If the rotation direction (clockwise or counterclockwise) of three vertices of a patch P in the virtual image is different from that in the image of the selected camera, the rotation direction is also checked in the image of the other camera. If it is the same as that in the

[1]Sec. 3.3 will describe how to produce new patches using P_C.

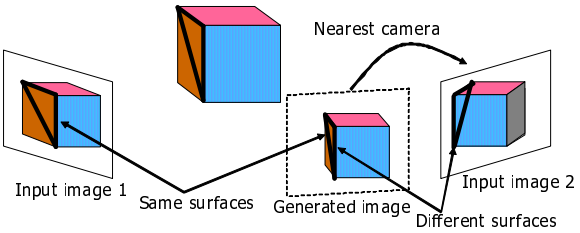
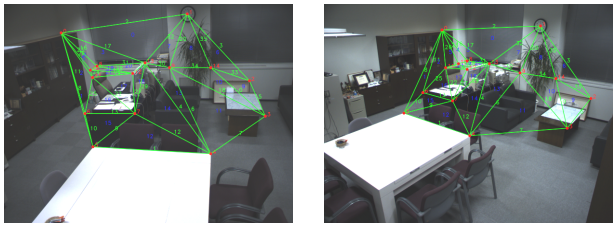


Figure 4. Different visible 3D surfaces depending on the viewpoint; thick triangles depict the same patch.



(a) Camera 1 (b) Camera 2

Figure 5. Initial triangle mesh.

virtual image, the other camera is selected for generating the image in the patch P.

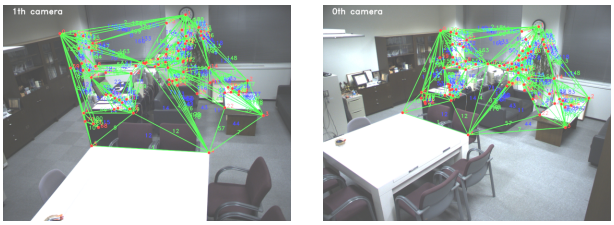
4.2 Projection-order Selection

Depending on the 3D positions of the vertices of the patches, the projection regions of the patches may overlap in a free-viewpoint image. In our method, the image patches are overwritten in the order of projection from input images. These overlapping patches have different images. That is, the generated image changes significantly depending on the order of projection, in particular, if a patch includes two or more noncontinuous 3D surfaces erroneously. To obscure such patches with errors, therefore, the patches are projected in descending order of unnaturalness. The unnaturalness is evaluated by the number of difference pixels detected by steps 1-1, 1-2, and 1-3.

5 Experiments

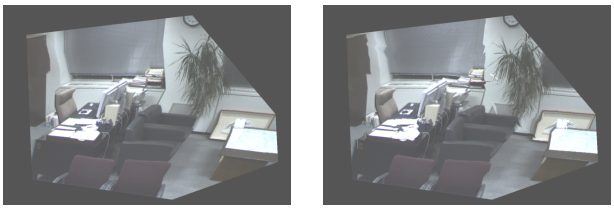
Our experiments were conducted with two IEEE1394 cameras (PointGrey Scorpion). Figure 5 shows the images observed by the cameras and the initial meshes in them (15 points and 22 triangle patches in each image).

Figure 6 shows rectified triangle meshes, each of which consists of 76 points and 167 patches, in two input images. Figure 7 shows examples of free-viewpoint images generated using (a) the initial meshes and (b) the rectified meshes. The viewpoint was at the center of two cameras.



(a) Camera 1 (b) Camera 2

Figure 6. Rectified triangle mesh.



(a) Before rectification (b) After rectification

Figure 7. Generated images.

To confirm the quality of the generated images, they were compared with an image observed by a real camera that was manually located at the viewpoint of the generated image. The difference between this real image and each generated image was computed and binarized; the same threshold was applied to two difference images. The free-viewpoint images were generated in 10 different virtual viewpoints. The means of the differences in the images obtained from the initial mesh and the rectified mesh were 3.2×10^{-3} and 1.9×10^{-3} (pixels/pixel), respectively.

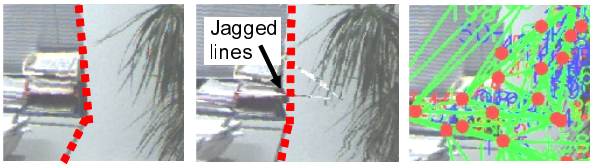
Although the difference pixels decreased, straight lines became uncomfortable in some regions as a result of rectification. Figure 8 shows an example. While a 3D straight line became more straight as depicted by red lines, jagged lines in the result of rectification (i.e., (b)) arose due to a number of narrow patches that crossed a boundary between noncontinuous 3D objects (i.e., a wall and a window).

We can make the following observations:

- While our method can rectify large deformations, small errors arise and make unnatural regions.
- Although a new point and one side of a new patch must be located on the boundary, our method cannot achieve that completely:
  - To locate a new point on the boundary, the Harris operator is not enough.
  - Although some new points were generated on the boundary, no new patch has its sides on the line.

6 Concluding remarks

This paper proposed a free-viewpoint imaging method that can be used in a complicated scene by using



(a) Before (b) After (c) Rectified Mesh

Figure 8. Disadvantage of rectification.

sparsely located cameras. In order to improve the quality, the following future work should be undertaken:

- Robust edge detection is needed to find boundaries.
- Boundaries should be more important in generating new points and patches.

References

[1] W. Matusik, C. Buehler, R. Raskar, S. Gortler, and L. McMillan,"Image Based Visual Hulls," In SIGGRAPH, pp.369–374, 2000.

[2] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality video view interpolation using a layered representation," ACM Trans. on Graphics, pp.600–608, 2004.

[3] S. M. Seitz and C. R. Dyer, "View Morphing," In SIGGRAPH, pp.21–30, 1996.

[4] I. Kitahara and Y. Ohta, "Scalable 3D Representation for 3D Video Display in a Large-scale Space," In IEEE Virtual Reality, pp.45–52, 2003.

[5] D. D. Morris and T. Kanade, "Image-consistent surface triangulation," In CVPR, Vol.1, pp.332–338, 2000.

[6] Z. Zhang, "A Flexible New Technique for Camera Calibration," IEEE Trans. on PAMI, Vol.22, No.11, pp.1330–1334, 2000.

[7] N. Otsu, "A threshold selection method from gray-level histograms," IEEE Trans. on SMC, Vol.SMC-9, No.1, pp.62–66, 1979.

[8] G. Vogiatzis, P. Torr, and R. Cipolla, "Bayesian Stochastic Mesh Optimisation for 3D Reconstruction," In BMVC, pp.711–718, 2003.

[9] A. Nakatuji, Y. Sugaya, and K. Kanatani, "Mesh optimization using an inconsistency detection template," In ICCV, pp.1148–1153, 2005.

[10] C. Harris and M. Stephens, "A combined corner and edge detector," In Alvey Vision Conference, pp.189–192, 1988.

[11] Y. Kanazawa and K. Kanatani, "Robust image matching preserving global consistency," In ACCV, Vol.2, pp.1128–1133, 2004