Journal of Zhejiang University SCIENCE A
ISSN 1673-565X (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Stepwise approach for view synthesis*

CHAI Deng-feng†[1,2], PENG Qun-sheng[1]

([1]*State Key Lab of CAD and CG, Zhejiang University, Hangzhou 310027, China*)

([2]*Institute of Spatial Information Technique, Zhejiang University, Hangzhou 310027, China*)

†E-mail: chaidf@cad.zju.edu.cn

Received Dec. 31, 2006; revision accepted May 8, 2007

Abstract: This paper presents some techniques for synthesizing novel view for a virtual viewpoint from two given views captured at different viewpoints to achieve both high quality and high efficiency. The whole process consists of three passes. The first pass recovers depth map. We formulate it as pixel labelling and propose a bisection approach to solve it. It is accomplished in $\log_2 n$ (n is the number of depth levels) steps, each of which involves a single graph cut computation. The second pass detects occluded pixels and reasons about their depth. It fits a foreground depth curve and a background depth curve using depth of nearby foreground and background pixels, and then distinguishes foreground and background pixels by minimizing a global energy, which involves only one graph cut computation. The third pass finds for each pixel in the novel view the corresponding pixels in the input views and computes its color. The whole process involves only a small number of graph cut computations, therefore it is efficient. And, visual artifacts in the synthesized view can be removed successfully by correcting depth of the occluded pixels. Experimental results demonstrate that both high quality and high efficiency are achieved by the proposed techniques.

Key words: View synthesis, Occlusion, Graph cut
doi:10.1631/jzus.2007.A1218 **Document code:** A **CLC number:** TP391

INTRODUCTION

As an alternative to traditional geometry-based rendering techniques for image synthesis, image-based rendering (IBR) techniques have recently received much interests. Given images of the scene captured by cameras setting at some known viewpoints, IBR techniques synthesize views for arbitrary novel viewpoints as if the views are captured by a camera setting at those points.

Shum and Kang (2000) presented a comprehensive survey of IBR techniques and put them into a continuous rendering spectrum according to how many images of the scene are needed and how much geometry information on the scene is used.

At one end of the rendering spectrum, texture mapping methods (Heckbert, 1986) use a few images as texture of the scene. But they need accurate geometric models of the scene and are essentially geometry-based techniques.

At the other end of the spectrum, light field rendering (Levoy and Hanrahan, 1996) and lumigraph (Gortler *et al*., 1996) formulate the rendering as sampling and re-sampling of the plenoptic function (McMillan and Bishop, 1995). They need no geometric knowledge of the scene. However, they need a large number of images of the scene, which poses the problem of image acquisition and storage.

In-between these two ends, there are many representative methods, such as 3D warping (Mark *et al*., 1997), layered depth images (LDI) (Shade *et al*., 1998) and view interpolation (Chen and Williams, 1993), that render novel views based on the tradeoff between images and geometry information. These methods rely on the successful establishment of point correspondences between different views, which is a classic problem and ongoing topic in computer vision (Dhond and Aggarwal, 1989; Scharstein and Szeliski, 2002).

* Project (No. 2002CB312101) supported by the National Basic Research Program (973) of China

In this paper, we discuss how to synthesize novel view for a virtual viewpoint from two given views of the scene captured by cameras setting at different viewpoints. We address both quality of the synthesized image and efficiency of the synthesis, and focus on scene representation, stereo matching, occlusion detection and refinement.

Previous work

Scharstein and Szeliski (2002) presented a comprehensive taxonomy and evaluation of two-frame dense stereo algorithms. We review some related algorithms in this subsection.

As argued in (Scharstein and Szeliski, 2002), disparity space or 3D space representation is critical to stereo algorithms. Some algorithms work in disparity space; they use disparity space image with respect to a reference view or with respect to a "cyclopian" view (Bobick and Intille, 1999). It is difficult to deal with multiple views based on this representation. Some algorithms use voxel-based representation (Seitz and Dyer, 1999; Kutulakos and Seitz, 2000). To achieve good results, it is necessary to design the resolution of voxel space carefully. All these methods are developed to recover the true 3D scene, the results are optimal with respect to one or all of the input views, but not the virtual view as desired in the view synthesis.

Also demonstrated in (Scharstein and Szeliski, 2002), global optimization is necessary for stereo algorithms to achieve good performance. Dynamic programming (Ohta and Kanade, 1985) is an efficient technique for global optimization. It imposes constraints within each epipolar line independently and establishes pixel correspondences in a polynomial time. However, since the constraints are imposed for each epipolar line independently, it is difficult to keep consistency across epipolar lines. Recently, Criminisi *et al.*(2003) proposed a technique for filtering the cost space to improve inter epipolar lines consistency, but it does not impose inter epipolar line constraints explicitly at all. As an alternative, the constraints across epipolar lines can be modelled explicitly in a global energy minimization framework (Terzopoulos, 1986), which has a Bayesian interpretation in (Geman and Geman, 1984). Many approaches have been proposed to minimize this kind of energy; graph cut (Boykov *et al.*, 2001; Kolmogorov and Zabih, 2001; 2002) and belief propagation (Sun *et al.*, 2002) methods have received much interests in recent years and proved to be effective. But these methods are inefficient and do not suit practical applications.

Occlusion is an important but challenging topic in stereo matching. It usually leads to visual artifacts in the synthesized view and therefore must be dealt with specially. There are some algorithms to attack the occlusion problem. Some algorithms just use robust measure to reduce the sensibility of matching to occlusion, e.g. (Sara and Bajcsy, 1997). Some algorithms model occlusion explicitly in the matching process, e.g. (Bobick and Intille, 1999; Kolmogorov and Zabih, 2001). Some algorithms propose a post processing after matching to detect occlusion, e.g. (Silva and Santos-Victor, 2000). These methods do not take the virtual view into account. But, it is the virtual view that plays the most important role in the view synthesis. These methods do not estimate depth in the occluded regions, which is important for synthesized view to achieve high quality.

Overview of stepwise approach

In this paper, we choose virtual view associated depth map as the representation of the 3D scene and propose a stepwise approach for view synthesis. This approach consists of three components with each component doing one pass of the synthesis process: (1) depth recovery: to recover depth map associated with the virtual view; (2) occlusion detection and filling: to detect occluded pixels in the virtual view and recover their depth; (3) color assignment: to assign color for each pixel in the virtual view.

The global optimal depth map is recovered using our novel bisection approach for pixel labelling. The whole labelling is accomplished in $\log_2 n$ (n is the number of depth levels) steps. At each step, it involves an energy minimization that can be solved exactly via a single graph cut computation.

Occluded pixels are detected using the recovered depth map. Nearby foreground and background pixels are used to model foreground and background depth curve. Depth of occluded pixels are reasoned from modelled depth curve and refined using an optimization step, which is accomplished by a single graph cut computation. Visual artifacts can be removed with the help of recovered occluded pixels and their depth.

For each pixel in the novel view, the corresponding pixels in the input views can be determined with the help of the recovered depth map and occlusion map. Colors of pixels to be synthesized are computed from colors of their corresponding pixels in the input views. Occlusion is taken into account in the computation.

The rest of this paper is organized as follows, we first describe the virtual view associated depth map in Section 2. Then, we present the bisection approach for depth recovery in Section 3. Occlusion modelling, detection and filling are presented in Section 4. After that, color assignment is shown in Section 5. At last, we present some experimental results in Section 6 and draw conclusions in Section 7.

VIRTUAL VIEW ASSOCIATED DEPTH MAP

Since disparity space representation restricts the methods to deal with rectified images, we choose depth map as a representation, and choose depth map associated with the novel view to be synthesized as that of (Yang *et al*., 2002).

As shown in Fig.1, V is the virtual viewpoint for the view to be synthesized, L and R are viewpoints for two given views denoted as I_L and I_R. There are many existing techniques that can be used to obtain the projection matrix P_L and P_R of the two given views. The projection matrix for the virtual view can be computed from the knowledge of the viewing conditions (Hartley and Zisserman, 2000). In this paper, we assume that the projection matrix of the three views P_L, P_R and P_V are known. If the depth d_p for pixel p is known, its corresponding 3D point P_s that projects into I_V at p is determined, and its two corresponding points p_l in I_L and p_r in I_R can be computed easily.

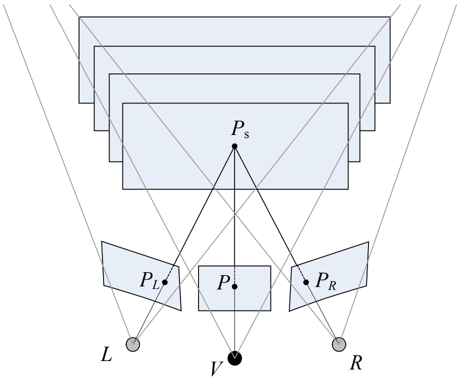


Fig.1 Virtual view associated depth map. The 3D scene is represented as depth map associated with the novel view indicated by black viewpoint *V*

The 3D space is split by a set of planes parallel with the virtual view plane, and it is carried out by a uniform sampling of the inverse depth space d', which is the inverse of the original depth d (i.e. $d'=d^{-1}$). Because the scene has limited range of depth, the set of sampled depth and inverse depth has finite elements. Assume that $D'=\{d_0',\ldots,d_n'\}$ contains all possible inverse depth d_i' ($i=0,1,\ldots,n$). Each pixel p in the virtual view (the set of all pixels is denoted as P) has a possible depth d_p from D (and an inverse depth d_p' from D').

This representation has many advantages. First, because this representation depends on the virtual view instead of the input views, it can be easily extended to deal with multiple input views. Secondly, since the depth map is associated with the virtual view, the recovered depth map is optimal with respect to the virtual view as desired. Third, occlusion detection and modelling can take virtual view into account. In fact, the occlusion modelling is based on the virtual view in this paper.

DEPTH RECOVERY

In this section, we formulate depth recovery as pixel labelling, then propose a bisection approach for it.

Depth recovery by labelling

As shown in Section 2, D contains all possible depth. To determine the depth of pixel p is to assign one label (depth) d in D to p. A labelling for the whole view is the assignment of labels d_p for all pixels P, which consists of all pixels in the virtual view. The depth recovery for the whole view can be formulated as:

"Find a labelling f from the set F of all possible labelling, to agree with the two given views and meet some prior knowledge about the scene as much as possible."

In the Bayesian framework based on Markov Random Fields (Geman and Geman, 1984), the above labelling involves finding a maximum *a posterior* estimate of f, which in turn can be formulated as minimizing the energy:

$$E(f)=E_\text{d}(f)+E_\text{s}(f), \tag{1}$$

where data term $E_d(f)$, which measures the agreement between the labelling and the given images, reflects the likelihood of the labelling; smooth term $E_s(f)$, which measures smoothness of the labelling, reflects the prior probability of the labelling.

Bisection approach for labelling

There are many methods developed to minimize the energy Eq.(1), among which graph cut methods are most effective. Unfortunately, when there are more than two labels, it is a multi-way cut problem and is NP-hard. α-β swap algorithm and α expansion algorithm adopt swap move and expansion move respectively to update the solution, and improve the solution iteratively to reach a local optimal one. But their complexities are $O(n^2)$ and $O(n)$, respectively (n is the number of labels), they are inefficient when n is large and are not suited for practical application. Here, we propose a bisection approach instead to improve the efficiency.

First, we map the depth set D to the label set $L=\{0,\ldots,n\}$. Then, each label $l\in L$ is written as a binary digit and a binary tree T is constructed for the label set. The root node denotes the whole L. Each node denotes one subset of L. T is constructed in this way: first, L is partitioned into two sets L_0 and L_1 denoted by the two child nodes. Whether L_0 or L_1 should $l\in L$ be classified into is decided by whether its corresponding bit is 0 or 1. Further, L_0 and L_1 are partitioned into two subsets respectively. This process is repeated until each final set contains only one label.

Fig.2 shows one example of the constructed binary tree. The indexes of bits are shown in the left column. The whole label set is $L=\{0,1,2,3\}$. The index of the highest bit corresponding to the root node is 2 and the partition of the whole set L is decided by bit 1. As shown, each leaf node denotes one subset that contains only one label.

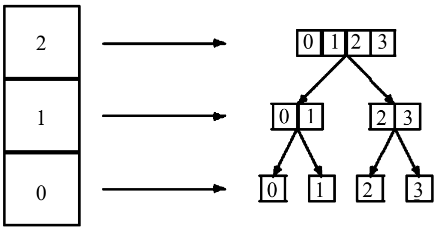


Fig.2 Binary decision tree for classification. All brother nodes are aligned horizontally to be one layer, whose number is shown in the left column. The classification is carried out layer by layer from top to bottom, one layer each time

Given a pixel p, its label can be determined by finding a leaf node, which can be accomplished by finding the corresponding "root to leaf" path on T. Each step towards this goal is a selection of left or right child node, which is denoted as labelling p with 0 or 1. When all pixels in P are labelled simultaneously, they are carried out from the root to the leaf step by step. Each step corresponds to one layer of T (i.e. one bit of the label). Therefore, the whole labelling can be accomplished in $\log_2 n$ (n is the number of depth levels) steps. In each step, the global optimal labelling can be achieved by minimizing the energy:

$$E(X)=\sum_{p\in P}E_p(x_p)+\sum_{p\in P}\sum_{q\in Q}E_{p,q}(x_p,x_q), \tag{2}$$

where $X=\{x_p,\ x_p\in\{0,1\},\ p\in P\}$, N_p is the set of neighbors of p, $x_p=0$ and $x_p=1$ respectively denote that left and right child node are selected.

1. Data term

When p is labelled with 0, the left child node denoted by L_p^0 is selected, and p can take any label in L_p^i (i=0, 1), i.e.

$$f_p\in\begin{cases}L_p^0, & p=0,\\ L_p^1, & p=1.\end{cases} \tag{3}$$

$E_p(x_p)$ is estimated by adopting "winner takes all" schema, i.e.

$$E_p(x_p)=\min(D_p(f_p), f_p\in L_p^{x_p}). \tag{4}$$

$D_p(f_p)$ measures the agreement between labelling p with f_p and the given images. It can be specified by

$$D_p(f_p)=|I_L(p_l)-I_R(p_r)|, \tag{5}$$

where p_l and p_r are corresponding pixels of p when p has depth f_p, $I_L(p_l)$ and $I_R(p_r)$ are their colors.

2. Smooth term

Similarly, we have

$$f_q\in\begin{cases}L_q^0, & q=0,\\ L_q^1, & q=1.\end{cases} \tag{6}$$

$E_{p,q}(x_p,x_q)$ is taken as the expectation of $V_{p,q}(f_p,f_q)$, i.e.

$$E_{p,q}(x_p,x_q)=\sum_{f_p\in L_p^{x_p}}\sum_{f_q\in L_q^{x_q}}V_{p,q}(f_p,f_q)P(f_p,f_q),\quad(7)$$

where $V_{p,q}(f_p,f_q)$ is a penalty for p and q having different labels (Boykov *et al.*, 2001), and $P(f_p,f_q)$ is approximated as:

$$P(f_p,f_q)=\frac{\exp(-V_{p,q}(f_p,f_q))}{\sum_{f_p\in L_p^{x_p},f_q\in L_q^{x_q}}\exp(-V_{p,q}(f_p,f_q))}.\quad(8)$$

It can be proved that

$$E_{p,q}(0,0)+E_{p,q}(1,1)\le E_{p,q}(0,1)+E_{p,q}(1,0).\quad(9)$$

Therefore, Eq.(2) can be minimized via graph cut method (Kolmogorov and Zabih, 2004).

OCCLUSION DETECTION AND FILLING

Occlusion is common in binocular or multicular vision. Since it is very difficult to deal with occlusion, some algorithms just ignore this phenomenon. But this may results in visual artifacts in the synthesized view. In this section, the second pass that detects occluded pixels and recovers their depth is presented. With the help of detected occluded pixels and their depth, visual artifacts can be removed successfully.

Occlusion detection

Although there are many methods to attack occlusion, they only take input views into account. Since the virtual view plays an important role in view synthesis, it must be taken into account. If possible, occlusion modelling should be based on the virtual view. So, we define the occluded pixel as:

"Occluded pixel is the one that is visible in the virtual view, but visible in one and only one of the given two views. Occluded pixels that are not visible in left view are called 'left occluded pixels' and those that are not visible in the right view are called 'right occluded pixels'."

There may be some scenes that can produce pixels visible in the virtual view but visible in none of the given two views, but no information about these pixels can be obtained from the given views, so, they are ignored to simplify the occlusion detection and filling. We assume that the scene point visible in the virtual view is visible in at least one given view.

Fig.3 (see P1224) is an illustration of the above definition of occlusion. F is foreground while A~E and G~K are background. A and K are out of view at the virtual viewpoint. E and G are occluded by F at the virtual viewpoint. They are not presented in the virtual view and are not discussed here. D is visible from virtual and left viewpoints but not the right one. H is visible from virtual and right viewpoints but not the left one. They produce right and left occluded pixels.

If the depth map associated with virtual view is recovered accurately, the corresponding pixel p_l and p_r of pixel p can be determined easily. If some pixels with smaller depth have the same corresponding point p_l, then p is detected as left occluded pixel. If some pixels with smaller depth have the same corresponding point p_r, then p is detected as right occluded pixel. All left and right occluded pixels can be detected in this way.

Fig.4 (see P1224) shows one example of the recovered depth map and detected occlusion map. In practice, the depth map is not recovered precisely, each occluded region in each scanline can broaden some pixels in both left and right directions.

Occlusion filling

Because the pixels in the occluded regions are visible in only one given view, $D_p(f_p)$ calculated by the way described in Section 3.2.1 does not measure the agreement between labelling p with f_p and the given views. In turn, it does not reflect the likelihood of the labelling. Occluded regions usually lie on the boundary of the foreground, and they may not meet the smooth condition that energy minimization method is built on. This leads to incorrect depth in the resultant depth map and visual artifacts in the synthesized view at the occluded region. Fig.5 shows the recovered inverse depth map of one scanline. The recovered depth of occluded pixels is incorrect as shown, these pixels have been assigned an intermediate depth between foreground depth and background depth. We propose a schema for reasoning about the depth of these pixels.

For each scanline as shown in Fig.6, we first recover a foreground depth curve and a background depth curve for each occluded segment (e.g. occ1) by applying curve fitting using depth of nearby foreground and background pixels. We assume that the scene is planar and that the depth curve is a straight line. As shown in the figure, two straight lines are recovered in each occluded region, one denotes foreground depth curve and the other denotes background depth curve.

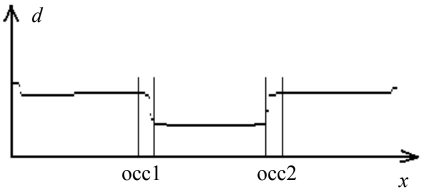


Fig.5 Depth map of one scanline. Each pixel denoted by *x* coordinate has its own depth *d*. Foreground has small depth while background has larger depth. occ1 and occ2 are two occluded regions

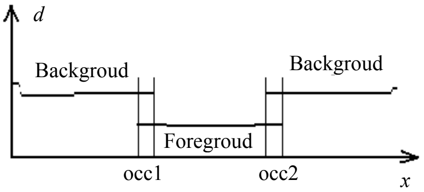


Fig.6 Occlusion modeling. In the occluded regions, foreground and background depth curves (straight line) are fitted using depth of nearby foreground and background pixels respectively

The next step is to decide which pixels belong to foreground and which pixels belong to background. This can be formulated as labelling the occluded pixels with 0 or 1, the pixels labelled with 1 belong to foreground and those labelled with 0 belong to background. Global optimal labelling can be achieved by minimizing the energy:

$$E(X)=\sum_{p\in P_{\mathrm{o}}}E_p(x_p)+\sum_{p\in P_{\mathrm{o}}}\sum_{q\in N_p}E_{p,q}(x_p,x_q), \quad (10)$$

where X={x_p, $x_p\in\{0,1\}$, $p\in P_{\mathrm{o}}$}, P_{o} is the set of occluded pixels. The data term is specified as:

$$E_p(x_p)=\begin{cases}\cos(t_{\mathrm{occ}}), & x_p=0,\\ D_p(f_p^1), & x_p=1,\end{cases} \quad (11)$$

where $\cos(t_{\mathrm{occ}})$ is the cost of labelling p as background pixel (i.e. occluded pixel), $D_p(f_p)$ is specified by the the means presented in Section 3.2.1, and $f_p^{\ 1}$ is the depth of p when p is labelled as foreground pixel and calculated using foreground depth curve.

The smooth term is specified as:

$$E_{p,q}(x_p,x_q)=\begin{cases}0, & x_p=x_q,\\ 1, & x_p\neq x_q,\end{cases} \quad (12)$$

where $f_p^{\ 0}$ and $f_p^{\ 1}$ are calculated using foreground and background depth curves when p are labelled as foreground and background pixels, respectively. The same notation is applied to q. Eq.(10) can be minimized exactly by a single graph cut computation.

COLOR ASSIGNMENT

Once the depth map and occlusion map associated with the virtual view are recovered, the corresponding pixels p_{l} and p_{r} can be easily determined for each pixel p in the virtual view. The color of p is assigned by:

$$I(p)=\begin{cases}I(p_{\mathrm{l}})\cdot w_{\mathrm{l}}+I(p_{\mathrm{r}})\cdot w_{\mathrm{r}}, & p\notin P_{\mathrm{o}},\\ I(p_{\mathrm{l}}), & p\in P_{\mathrm{o}}^{\mathrm{r}},\\ I(p_{\mathrm{r}}), & p\in P_{\mathrm{o}}^{\mathrm{l}},\end{cases} \quad (13)$$

where, $I(p)$ is color of pixel p, w_{l} and w_{r} are weights of two colors from left and right views respectively, P_{o} is the set of all occluded pixels, $P_{\mathrm{o}}^{\ \mathrm{l}}$ and $P_{\mathrm{o}}^{\ \mathrm{r}}$ are set of left and right occluded pixels respectively. w_{l} and w_{r} can be determined by the distances between the virtual viewpoint and the two known viewpoints. For example, when virtual viewpoint lies on the middle point, $w_{\mathrm{l}}=w_{\mathrm{r}}=0.5$ can be applied, when it lies near left viewpoint, $w_{\mathrm{l}}>w_{\mathrm{r}}$ should be adopted.

EXPERIMENTAL RESULTS

The above approach is applied to view synthesis for a variety of images. In this section, we present some examples to illustrate the procedure of synthesis and show some results of synthesis.

Bisection approach for stereo matching

Fig.7 shows how bisection approach recovers the depth map. The top row shows two input views. The rest shows the recovered depth map associated with the virtual view, which has viewpoint setting at the middle point of the baseline of the two input views

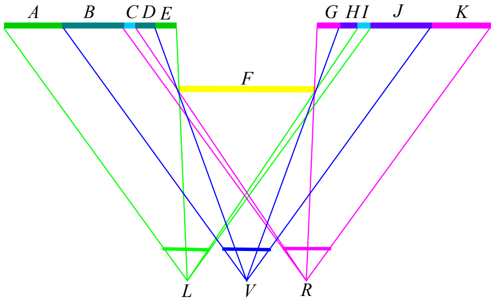
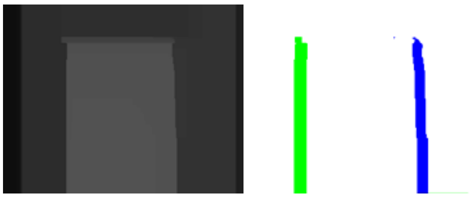


Fig.3 Illustration of occlusion. *L*, *R*, and *V* denote left, right and virtual viewpoints respectively. *F* is foreground while *A*~*E* and *G*~*K* are background



(a) (b)

Fig.4 Inverse depth map and occlusion map. (a) One inverse depth map; (b) Its corresponding occlusion map. Blue areas indicate left occlusion region and green areas indicate right occlusion regions

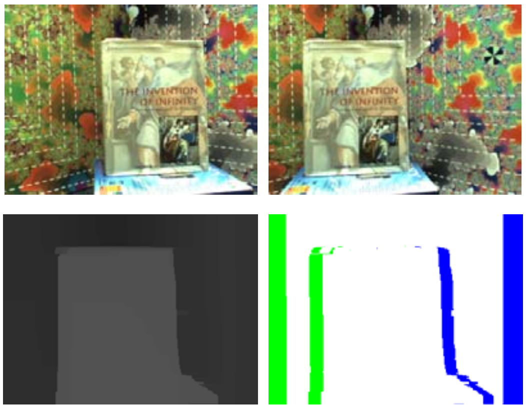


Fig.8 Occlusion detection. The top row shows left and right views respectively. The bottom row shows recovered depth map and occlusion map

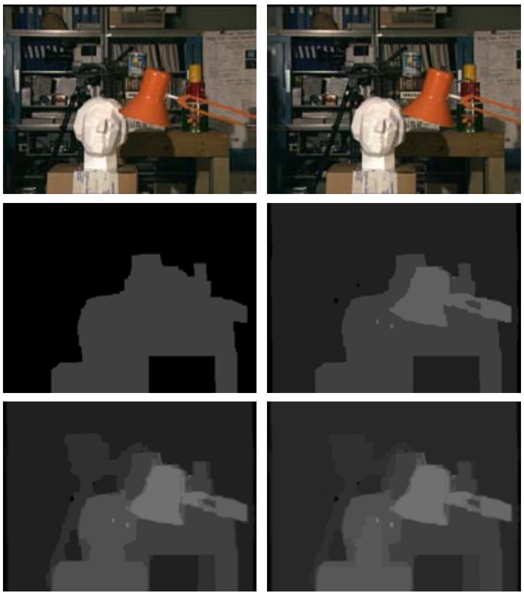


Fig.7 Depth recovery. Top row shows left and right views respectively. The rest rows show depth images recovered step by step

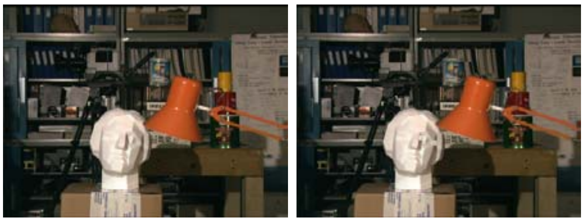


Fig.9 True and synthesized view. Left and right ones are true and synthesized images, respectively

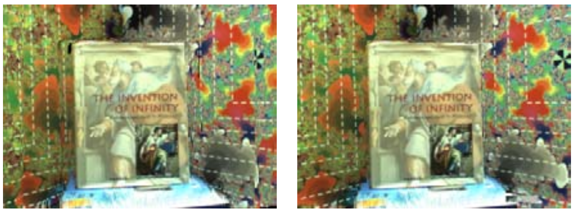
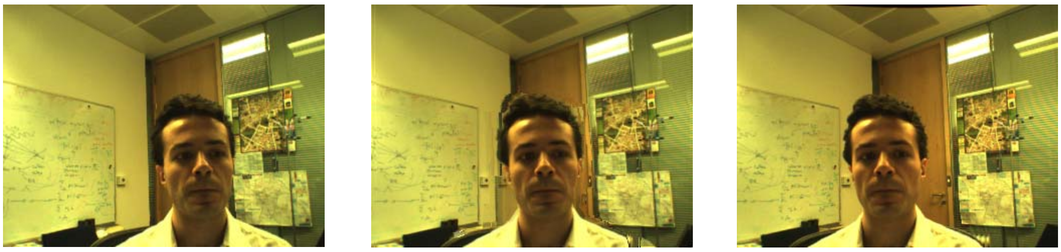


Fig.10 Artifacts removal. Left and right ones are synthesized images with and without dealing with occlusion, respectively



(a) (b) (c)

Fig.11 View synthesis. (a) Left image; (b) Virtual image; (c) Right image

and the same viewing direction as the input views. In this example, the depth level is 16. The whole process of depth recovery consists of 4 steps. The depth maps recovered step by step are shown from left to right and top to down in the figure. The depth is magnified 16 times in the figure to assist illustration. Bisection approach is tested against Middleburry Stereo dataset and good results are achieved, the complete results can be found in (Chai, 2006).

Occlusion detection and filling

The top row of Fig.8 (see P1224) shows two views of a scene consisting of a book (foreground) in front of the two planar backgrounds. As shown, the disparity between the two views is much larger than the example presented in Fig.7. In this case, the occlusion can be easily observed and must be detected. We also set the virtual viewpoint at the middle point of the baseline. The bottom row shows the recovered depth map for the virtual view and its corresponding occlusion map detected. As shown, the occluded pixels are detected successfully.

Quality

Fig.9 (see P1224) shows synthesized result corresponding to Fig.7. The left image is the view captured by a real camera setting at the virtual viewpoint. The right one is the view synthesized using the method presented in this paper. Since occlusion is not obvious, the second pass is not applied. As shown, there is little difference between the true and synthesized views.

Fig.10 (see P1224) shows synthesized views corresponding to Fig.8. The left and right images show views synthesized without and with recovery of depth of occluded pixels respectively. As shown, the artifacts in the occluded regions are removed by taking occluded pixels and their depth into account.

Fig.11 (see P1224) shows an application of our method to gaze manipulation for one-to-one teleconferencing. Figs.11a and 11c are left and right image captured by two cameras placed on the two sides of the computer monitor respectively. They lack eye contact and may lead to undesired effects on the interaction. Fig.11b is virtual image for viewpoint at center of the baseline, it is synthezied by our method. As shown, the gaze has been corrected.

Efficiency

Most of the computation time is used to recover the depth. If there are n depth levels, bisection approach needs $\log_2 n$ graph cut computation while state of the art graph cut methods need $k\times n$ ($k>1$) graph computation. Therefore, high efficiency is achieved by our method. In our experiments, to synthesize one image of size 320×240 takes only about 1 s while previous methods take many minutes.

CONCLUSION

This paper presents some techniques for synthesizing novel view from two given views. It has proposed a bisection approach for pixel labelling to recover the depth map. It has also proposed occlusion detection and filling techniques to detect occluded pixels and recover their depth.

Using bisection approach, the global optimal depth map can be recovered in $\log_2 n$ (n is the number of depth levels) steps, each step involves only one single graph cut computation. This assures high quality and high efficiency of the depth recovery. Depth of occluded pixels is recovered by a single graph cut computation. With the help of depth of occluded pixels, visual artifacts can be removed successfully. This assures high quality of the synthesized view.

Modification of the presented method is interesting. First, it is easy to extend the proposed method to deal with multiple input views based on our representation. Multiple input views can improve the accuracy of the recovered depth map since more information is presented. But similarity measurement and occlusion involving multiple views are more complicated. We plan to investigate these topics in the near future. Second, it is easy to simplify the proposed method to deal with two rectified views. Much higher efficiency can be achieved in this case. Further, graph cut algorithm can be implemented in a parallel way. It is our next work to implement the proposed method in the GPU to achieve high speed to support real-time application.

References

Bobick, A., Intille, S., 1999. Large occlusion stereo. *Int. J. Computer Vision*, **33**(3):181-200. [doi:10.1023/A:1008150329890]

Boykov, Y., Veksler, O., Zabih, R., 2001. Fast approximate energy minimization via graph cuts. *IEEE Trans. on Pattern Anal. Machine Intell.*, **23**(11):1222-1239. [doi:10.1109/34.969114]

Chai, D.F., 2006. Stereo Matching for Three Dimensional Visual Communication. Ph.D Thesis, Zhejiang University (in Chinese).

Chen, S.E., Williams, L., 1993. View Interpolation for Image Synthesis. Proc. SIGGRAPH'93, p.279-288. [doi:10.1145/166117.166153]

Criminisi, A., Shotton, J., Blake, A., Torr, P., 2003. Gaze Manipulation for One-to-one Teleconferencing. Proc. Int. Conf. on Computer Vision, p.939-946.

Dhond, U., Aggarwal, J., 1989. Structure from stereo—a review. *IEEE Trans. on Systems, Man, and Cybern.*, **19**(6):1489-1510. [doi:10.1109/21.44067]

Geman, S., Geman, D., 1984. Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images. *IEEE Trans. on Pattern Anal. Machine Intell.*, **6**(6):721-741.

Gortler, S.J., Grzeszczuk, R., Szeliski, R., Cohen, M.F., 1996. The Lumigraph. Proc. SIGGRAPH'96, p.43-54. [doi:10.1145/237170.237200]

Hartley, R., Zisserman, A., 2000. Multiple View Geometry in Computer Vision. University Press, Cambridge, UK.

Heckbert, P.S., 1986. Survey of texture mapping. *IEEE Computer Graphics and Applications*, **6**(11):56-67.

Kolmogorov, V., Zabih, R., 2001. Visual Correspondence with Occlusions Using Graph Cuts. Proc. Int. Conf. on Computer Vision, p.508-515.

Kolmogorov, V., Zabih, R., 2002. Multi-camera Scene Reconstruction via Graph Cuts. Proc. European Conference on Computer Vision, p.82-96.

Kolmogorov, V., Zabih, R., 2004. What energy functions can be minimized via graph cuts? *IEEE Trans. on Pattern Anal. Machine Intell.*, **26**(2):147-159. [doi:10.1109/TPAMI.2004.1262177]

Kutulakos, K., Seitz, S., 2000. A theory of shape by space carving. *Int. J. Computer Vision*, **38**(3):199-218. [doi:10.1023/A:1008191222954]

Levoy, M., Hanrahan, P., 1996. Light Field Rendering. Proc. SIGGRAPH'96, p.31-42. [doi:10.1145/237170.237199]

Mark, W., McMillan, L., Bishop, G., 1997. Post-rendering 3D Warping. Proc. Symposium on Interactive 3D Graphics, p.7-16. [doi:10.1145/253284.253292]

McMillan, L., Bishop, G., 1995. Plenoptic Modeling: An Image-based Rendering System. Proc. SIGGRAPH'95, p.39-46. [doi:10.1145/218380.218398]

Ohta, Y., Kanade, T., 1985. Stereo by intra- and inter-scanline search using dynamic programming. *IEEE Trans. on Pattern Anal. Machine Intell.*, **7**(2):139-154.

Sara, R., Bajcsy, R., 1997. On Occluding Contour Artifacts in Stereo Vision. Proc. IEEE Conf. on Computer Vision and Pattern Recognition, p.852-857. [doi:10.1109/CVPR.1997.609427]

Scharstein, D., Szeliski, R., 2002. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. Computer Vision*, **47**:7-42. [doi:10.1023/A:1014573219977]

Seitz, S., Dyer, C., 1999. Photorealistic scene reconstruction by voxel coloring. *Int. J. Computer Vision*, **35**(2):151-173. [doi:10.1023/A:1008176507526]

Shade, J., Gortler, S., Hey, L., Szeliski, R., 1998. Layered Depth Images. Proc. SIGGRAPH'98, p.231-242. [doi:10.1145/280814.280882]

Shum, H.Y., Kang, S.B., 2000. A Review of Image-based Rendering Techniques. Proc. IEEE/SPIE Visual Communications and Image Processing, p.2-13.

Silva, C., Santos-Victor, J., 2000. Intrinsic Images for Dense Stereo Matching with Occlusions. Proc. European Conference on Computer Vision, p.100-114.

Sun, J., Shum, H., Zheng, N., 2002. Stereo Matching Using Belief Propagation. Proc. European Conference on Computer Vision, p.510-524.

Terzopoulos, D., 1986. Regularization of inverse visual problems involving discontinuities. *IEEE Trans. on Pattern Anal. Machine Intell.*, **8**(4):413-424.

Yang, R., Welch, G., Bishop, G., 2002. Real-time Consensus-based Scene Reconstruction Using Commodity Graphics Hardware. Proc. 10th Pacific Conf. on Computer Graphics and Applications, p.225-235. [doi:10.1109/PCCGA.2002.1167864]