Disparity Estimation at Virtual Viewpoint for Real-time Intermediate View Generation

In-Yong Shin and Yo-Sung Ho
Gwangju institute of Science and Technology (GIST)
261 Cheomdan-gwagiro, Buk-gu, Gwangju, 500-712, South Korea
{siy0808, hoyo}@gist.ac.kr

Abstract

In this paper, we propose a real-time view interpolation method using virtual viewpoint disparity estimation and parallel processing through a graphics processing unit (GPU). In order to make free viewpoint images, we need the geometry information. We can use left and right viewpoint disparity maps to generate virtual viewpoint images. However, the disparity map estimation operation takes a long time due to its iterative nature. In the proposed method, we estimate a disparity map at the virtual viewpoint from stereo images using the belief propagation method. This method needs only one disparity map, compared to the conventional methods that need two disparity maps. In the view synthesis part, we warp pixels from the reference images to the virtual viewpoint image using the disparity map at the virtual viewpoint. For real-time acceleration, we utilize a high speed GPU parallel programming, called CUDA. As a result, we can interpolate virtual viewpoint images in real-time.

Keywords---Stereo matching, belief propagation, CUDA, DIBR, GPU programming, view interpolation

1. Introduction

In recent years, various researches have been on 3D video system as increasing interest in 3D multimedia service. 3D video system provides realistic multimedia service that offers a 3D effect based on binocular depth cue. It can be used in a wide range of multimedia applications such as immersive games, movies, presentations, video conferencing, 3D TV and medical imaging. With the increasing demand of 3D video display, MPEG has made an effort for 3D audio-visual (3DAV) technology standardization [1]. The information of 3D video display is characterized by a disparity map that consists of disparity vectors (DVs) for pixel pairs between the left- and right-images. As shown in Figure 1, virtual views can be synthesized with respect to different virtual camera positions using disparity map. Thus, the disparity map estimation and virtual view synthesis are two most important parts in 3D video display.

Many disparity map estimation algorithms for a stereo image pair have been proposed in the past, and they can be classified into two types. One type emphasizes a low computational complexity for a real time implementation. The block matching algorithm (BMA) provides a good example for this type. Due to the low complexity, the quality of the resulting disparity map is lower, which in turn affects the quality of synthesized virtual view. The other type attempts to get accurate disparity map with a higher complexity. For example, a global energy minimization algorithm was proposed in [2-4] for this purpose. Even though these methods can be used to synthesize good virtual views, they demand a large amount of computation so that their real-time implementation is a challenge.

Most virtual viewpoint image generation methods use two disparity maps (left and right) or one side disparity map. The methods generate accurate synthesized image at a virtual viewpoint. However, it takes a long time to estimate two disparity maps. Second method needs half time for the disparity estimation, but synthesis accuracy is lower than first one.

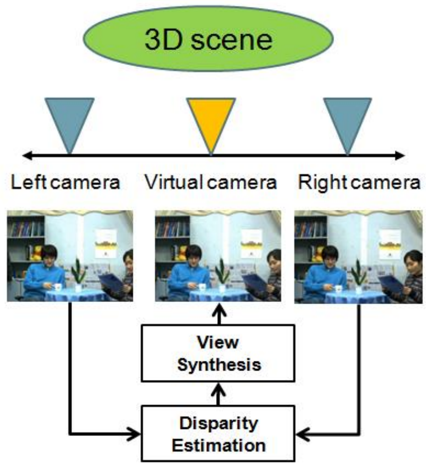


Figure 1: Outline of view synthesis method

In this paper, we propose a real-time virtual viewpoint synthesis method. In order to synthesize a virtual viewpoint image in real-time, we estimate a single disparity map at the virtual viewpoint. After the disparity estimation process, we decide an occlusion map of the virtual viewpoint to select regions which can be back-projected. We synthesize the virtual viewpoint image using the disparity map and the occlusion map.

Additionally, we implement our proposed method in real-time using a parallel programming. CUDA is the general purpose computing engine in NVIDIA GPUs that is accessible to software developers through industry standard programming languages.

2. Related work

There are many researches related to view interpolation techniques. Generally, left and right disparity maps are used for view synthesis [5]. As shown in Figure 2, this method makes left and right disparity maps and warp virtual images respectively. Then, two virtual images are summed by a weighting function. Although it has heavy complexity due to two disparity estimation processes, it generates virtual images which are respectable quality.

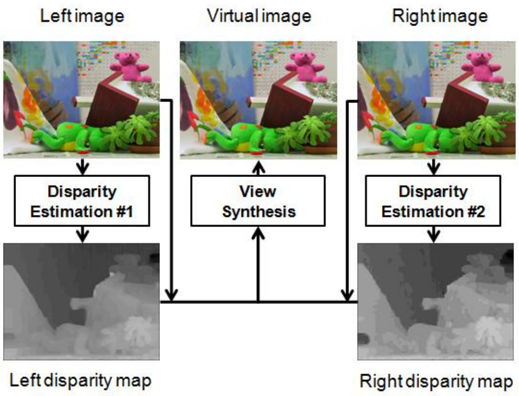


Figure 2: Conventional view synthesis method.

Also, single disparity map estimation process can be used to interpolate a virtual viewpoint image. For instance, Oh et al. [6] use single disparity map at left or right viewpoint. Omitting a disparity map estimation part of the other viewpoint leads it to fast execution. Qualities of view synthesis outputs are, however, lower than outputs of the first method due to occlusion regions which should only refer pixel information from the other viewpoint.

3. Proposed method

In this section, we describe the proposed method for a real-time virtual viewpoint image generation using a virtual viewpoint disparity estimation method and an occlusion map decision. Our method contains following steps.

Step 1: We directly estimate the virtual viewpoint disparity map using a new energy function. For the optimization, we utilize the hierarchical belief propagation method which produces a global solution effectively.

Step 2: An occlusion map of the virtual viewpoint is decided through virtual disparity map warping to right and left viewpoints. Then, we can properly warp left and right images to the virtual viewpoint using the disparity map and the occlusion map.

Step 3: If there are some hole-regions which are occlusion area from left and right viewpoints, we fill it with neighboring pixels.

A. Virtual viewpoint disparity estimation

Global stereo matching methods find corresponding points using iterative energy minimization algorithms. An energy function E considers photo-consistency (a corresponding pixel should have the same intensity value) and piecewise smoothness (neighboring pixels are likely to have the same disparity value).

$$E(x,y,d) = E_{data}(x,y,d) + E_{smooth}(x,y,d) \quad (1)$$

As shown in Figure 3, we directly estimate a disparity map at a virtual viewpoint. For this case, we calculate data cost by using

$$\Sigma_{x,y}\left|I_R\left(x + d_{V\_L}(x,y), y\right) - I_R\left(x - d_{V\_R}(x,y), y\right)\right| \quad (2)$$

where $d_{V\_L/R}$ are $I_{L/R}$ disparity map of virtual viewpoint and input stereo image. Relationship between disparity values of d_{V_R} and d_{V_L} is

$$d_{V\_R}(x,y) = Alpha \times d_{V\_L}(x,y) \quad (3)$$

where $Alpha$ is a relative distance from the virtual viewpoint to the right viewpoint when a distance between left viewpoint and virtual viewpoint is one.

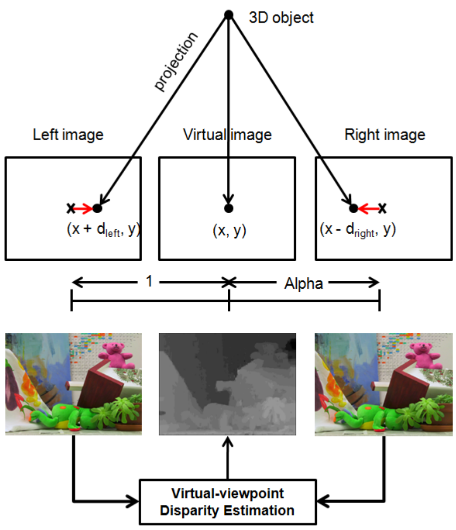


Figure 3: Virtual viewpoint disparity estimation.

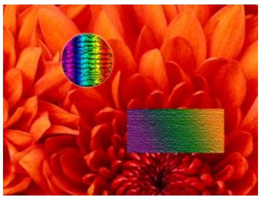
The hierarchical belief propagation algorithm is used to minimize the energy function. It passes messages called belief around in a four adjacency image grid. Message updates are in iterations. At one iteration step, each pixel of the adjacency graph computes its message based on the message to all the adjacent pixels. We set a number of stages and iterations of the hierarchical belief propagation as five. After iteration process, we compute beliefs (disparity map) for the virtual viewpoint synthesis.

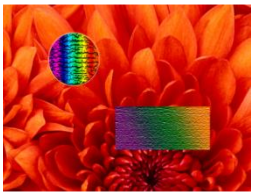
B. Occlusion decision and backward warping

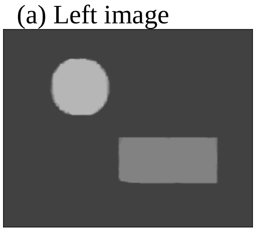
If we have a virtual viewpoint disparity map, we can make a virtual viewpoint image using backward warping. Before we warp reference images to virtual viewpoint, we have to consider a possible error due to occlusion region. In order to avoid occlusion region backward warping, we need to check occlusion regions of virtual viewpoint from left and right viewpoints. So, we decide an occlusion map value with four labels.

$$O_v(x,y) = \begin{cases} A, \text{Occluded from } I_L \\ B, \text{No occlusion} \\ C, \text{Occluded from } I_R \\ D, \text{Occluded from } I_L \text{ and } I_R \end{cases} \quad (4)$$

Occluded pixels are labeled by using virtual view warping to left and right viewpoints. Figure 4 shows occlusion map which has four labels (255-A, 128-B, 0-C, 1-D).


(a) Left image


(b) Right image


(c) Disparity map


(d) Occlusion map

Figure 4: Occlusion map decision.

C. Hole filling

Although warping process fills up proper pixel values from the reference images, there are still unknown hole-regions which cannot find a same fetch from the reference images due to a occlusion problem. Thus, we have to find the most plausible value by using surrounding pixel information. Most of the presented hole-filling methods use image interpolation or in-painting algorithm. In order to get best quality hole-filled images, neighboring background pixel values and their geometric information should be used. The reason why we use generally background region information is that background pixels rather than the foreground ones as the disoccluded area is more reasonable by definition of the disocclusion [10,11]. Thus, we fill up hole-regions with neighboring pixel values which have background disparities.

D. GPU implementation

In order to achieve real-time implementation, we use a parallel programming which executed on the GPU called CUDA. The architecture of CPU and GPU are very different. Although GPU has a small number of instruction control unit, it has a lot of cores capable of calculating floating points operation. Thus, GPU has a Single Instruction Multiple Threads (SIMT) structure [12]. So, image processing algorithm is very suitable for GPU programming due to that all of image pixels may have same operation. There is an important condition of the SIMT parallel processing. It is data independency between all data executed simultaneously. We implement whole process with parallel GPU programming.

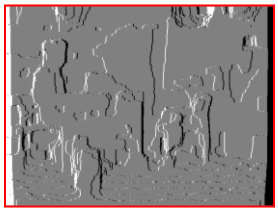
4. Experimental results

We have implemented three methods (method A, method B, and proposed method) and applied GPU parallel programming to proposed method. Method A interpolates the virtual viewpoint image using left and right disparity maps. Method B uses only left disparity map. For the experiment, we performed tests on several rectified stereo images which listed in Table 1. Test images are obtained from Middlebury stereo website and MVD test materials. Test stereo set includes not only stereo images, but also intermediate viewpoint images.

Table1: Specification of the test stereo image set.

sequence	Teddy	Poster	Cones	News paper	Book arrivals
size	640 x480	480 x416	480 x416	640 x480	640 x480
max disparity	30	20	20	50	50


(a) Disparity map


(b) Occlusion map


(c) Original image


(d) Virtual-view image

Figure 5: Input and output images.

Figure 5 shows input and output images of proposed method. In order to investigate synthesis quality, we calculate PSNR values with synthesized images and original intermediate viewpoint images. Figure 6 and 7 shows performance comparison of three methods. Results prove that our proposed method is accurate and faster than others. Moreover, implementation of GPU parallel programming carries additional speed up as shown in Figure 8. GPU accelerates execution speed up to 60 times by comparing CPU execution time.

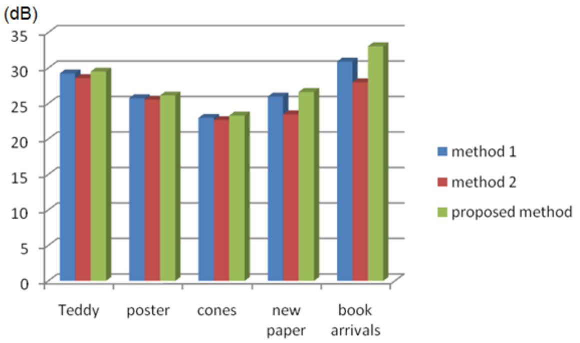


Figure 6: PSNR comparisons of three methods.

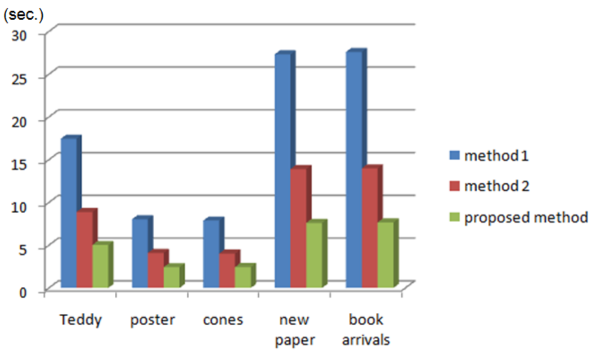


Figure 7: Execution time of three methods.

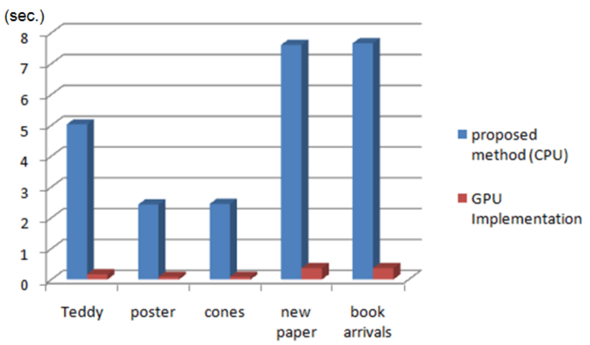


Figure 8: Execution time in CPU and GPU.

5. Conclusions

In this paper, we present a real-time view interpolation method. In order to make it more rapidly, we applied the virtual viewpoint disparity estimation method and GPU parallel programming. Previous methods estimate some duplicated and unnecessary disparity values for certain virtual viewpoint synthesis. Thus, our proposed method reduces complexity and makes accurate synthesis results by eliminating redundant calculations. We designed the data cost function for the virtual viewpoint disparity map. The hierarchical belief propagation algorithm is used to minimize the energy function. In the view synthesis part, we warp pixels from reference images to the virtual viewpoint using the virtual viewpoint disparity map. For a great real-time acceleration, we utilize a high speed GPU parallel programming called CUDA. As a result, we can synthesize the virtual viewpoint images in real-time.

Acknowledgements

This research was supported in part by MKE under the ITRC support program supervised by NIPA (NIPA-2011-(C1090-1111-0003)).

References

[1] ISO/IEC JTC1/SC29/WG11, “Survey of algorithms used for multi-view video coding,” N6909, January 2005.

[2] Y. Boykov, O. Veksler, and R. Zabih, “Fast approximate energy minimization via graph cuts,” IEEE Trans. Patt. Anal. Machine Intell., vol. 23, pp. 1222–1239, November 2000.

[3] V. Kolmogorov and R. Zabih, “Multi-camera scene reconstruction via graph cuts,” Proc. European Conf. Computer Vision, vol. 3, pp. 82–96, May 2002.

[4] J. Sun, N. N. Zheng, and H. Y. Shum, “Stereo matching using belief propagation,” IEEE Trans. Patt. Anal. Machine Intell., vol. 25, pp. 787–800, July 2003.

[5] ISO/IEC JTC1/SC29/WG11, "Contribution for 3D Video Test Material of Outdoor Scene," M1537, April 2008.

[6] J. D. Oh, S. Ma, and C. C. J. Kuo, “Disparity estimation and virtual view synthesis from stereo video,” in Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS), pp. 993–996, New Orleans, La, USA, May 2007.

[7] M. Tappen and W. Freeman, “Comparison of Graph Cuts with Belief Propagation for Stereo,” Proc. IEEE Int’l Conf. Computer Vision, vol. 1, pp. 508-515, 2003.

[8] P. F. Felzenszwalb and D. P. Huttenlocher, “efficient Belief Propagation for Early Vision,” CVPR, vol. 1, pp.261-268, 2004.

[9] Q. Yang, L. Wang, R. Yang, H. Stewenius, and D. Nister, “Stereo matching with color-weighted correlation, hierarchical belief propagation and occlusion handling,” CVPR, June 2006.

[10] K. Oh, S. Yea, and Y. Ho, “Hole Filling Method using Depth Based In-painting for View Synthesis in Free Viewpoint Television and 3-D Video,” Picture Coding Symposium, pp. 39 (1-4), 2009.

[11] M. Oliveira, B. Bowen, R. McKenna, Yu-Sung Chang, “Fast Digital Image Inpainting”, Proceedings of the International Conference on Visualization, Imaging and Image Processing (VIIP 2001), Marbella, Spain, September 3-5, 2001

[12] nVIDIA Corporation, “CUDA 3.2 Programming Guide,” inhttp://www.nvidia.com/cuda_develop.html, 2010