Virtual Viewpoint Disparity Estimation and Convergence Check for Real-Time View Synthesis

In-Yong Shin and Yo-Sung Ho

Gwangju Institute of Science and Technology (GIST)
261 Cheomdan-gwagiro, Buk-gu, Gwangju 500-712, Korea
{siy0808,hoyo}@gist.ac.kr

Abstract. In this paper, we propose a new method for real-time disparity estimation and intermediate view synthesis from stereoscopic images. Some 3D video systems employ both the left and right depth images for virtual view synthesis; however, we estimate only one disparity map at a virtual viewpoint. In addition, we utilize hierarchical belief propagation and convergence check methods to find the global solution rapidly. In order to use the virtual viewpoint disparity map for intermediate view synthesis, we build an occlusion map that describes the occlusion information in the virtual viewpoint region of the reference image. We have also implemented the total system using GPU programming to synthesize 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 a 3D video system as increasing interest in a 3D multimedia service. The 3D video system provides realistic multimedia services that offer 3D effects based on a binocular depth cue. It can be used in a wide range of multimedia applications such as immersive games, movies, presentations, video conferencing, 3D TVs and medical imaging. With the increasing demand of a 3D video display, MPEG has made an effort for a 3D audio-visual (3DAV) technology standardization [1]. The information of the 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 viewpoint images can be synthesized with respect to different virtual camera positions using the disparity map. Thus, disparity map estimation and virtual view synthesis are two most important parts in 3D video display.

Many disparity map estimation algorithms for stereo image pairs have been proposed in the past, and they can be classified into two types. One type emphasizes a low computational complexity for real time implementation. The block matching algorithm (BMA) provides a good example for this type. Due to the low complexity, a quality of the resulting disparity map is lower and the low quality disparity map affects a quality of synthesized virtual view. The other type attempts to get an accurate disparity map with a higher complexity. For example, global energy

Y.-S. Ho (Ed.): PSIVT 2011, Part I, LNCS 7087, pp. 121–131, 2011.

minimization algorithms are proposed in [2-4] for this purpose. Even though these methods can be used to synthesize high quality virtual viewpoint images, they demand a large amount of computation. So, their real time implementation is challenge.

Most virtual viewpoint image generation methods use two disparity maps (left and right viewpoints) or single disparity map at one of two reference viewpoints. First 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 disparity estimation, but synthesis accuracy is lower than first one due to occlusion regions.

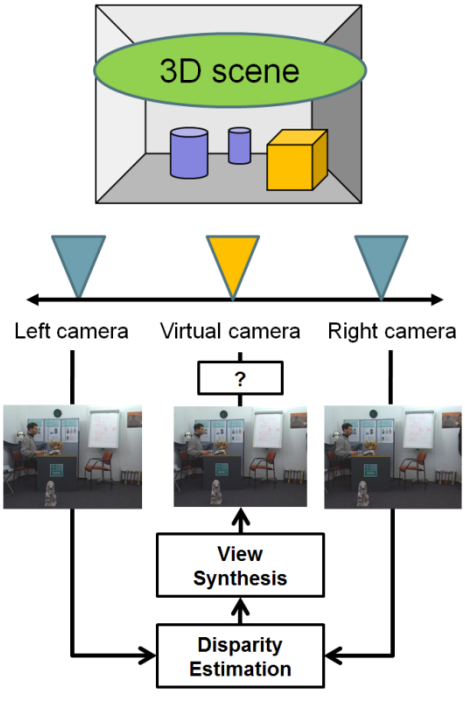


Fig. 1. Outline of view synthesis method

In this paper, we propose a real time virtual viewpoint synthesis method. In order to synthesize virtual viewpoint images in real time, we estimate disparity maps at the virtual viewpoint. Also we find convergence regions of the disparity map in the hierarchical belief propagation process. We cancel message updates at convergence regions to remove residual calculation by using a convergence map. After the disparity estimation process, we decide the occlusion map of the virtual viewpoint to select regions which can be back-projected. We synthesize the virtual viewpoint image using the virtual viewpoint disparity map and the occlusion map. Additionally, we implement the proposed method in real time using parallel programming called

CUDA. CUDA is the general purpose computing engine in NVIDIA GPUs that is accessible to software developers through industry standard programming languages.

This paper organized as follows. In Section 2, related work about view interpolation is explained. In Section 3, our proposed method is explained. In Section 4, the experimental results are given. The conclusion is presented in Section 5.

2 Related Work

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

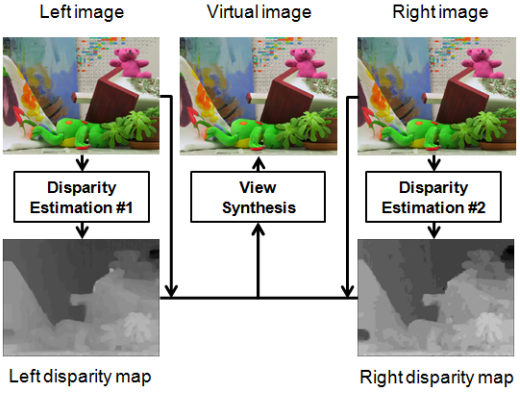


Fig. 2. Conventional view synthesis method

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

In the global disparity estimation methods, the belief propagation algorithm is frequently used [7]. Although it produces an accurate disparity map, it is too slow to be practical. So, the hierarchical belief propagation algorithm is proposed [8][9]. It runs much faster than the previous algorithms while maintaining comparable accuracy. The main difference between the HBP and the standard BP algorithm is that the HBP algorithm works in a coarse-to-fine manner. In other words, the HBP algorithm estimates the disparity map with a smallest resolution, then it estimates higher resolution disparity maps with a previously estimated disparity map. The basic steps are: (a) initialize the messages at the coarsest level to all zeros, (b) apply the BP

algorithm at the coarsest level to iteratively refine the messages, (c) use refined messages from the coarser level to initialize the messages for the next level. Specifically, if X is a pixel at a coarser level, and its corresponding pixels at the finer level are $X'_{i,i} \in [1, 4]$, as shown in Figure 3.

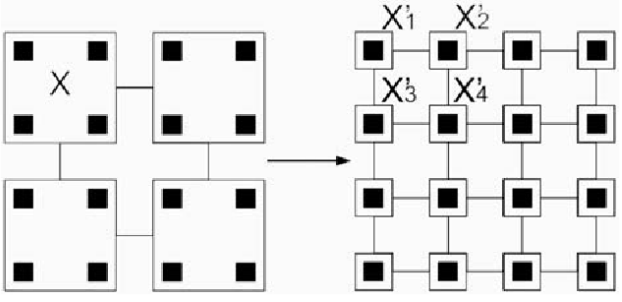


Fig. 3. Two levels in the coarse-to-fine method

Two main parameters S and T define the behavior of the HBP algorithm, S is the number of levels and T is the number of iterations at each level. Generally, we estimate disparity maps with five levels and ten iterations (S=5, T=10). Actually, we only compute beliefs (disparity map) at level 0 in the HBP algorithm.

3 Virtual Disparity Estimation and Convergence Check

In this section, we describe the proposed method for the real time virtual viewpoint image generation using the virtual viewpoint disparity estimation method and the convergence check method of the HBP algorithm. As shown in Figure 4, our method contains following steps.

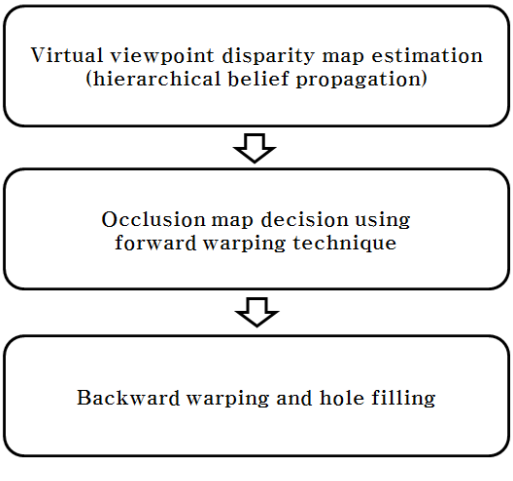


Fig. 4. View synthesis using virtual disparity estimation

3.1 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 similar disparity value).

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

As shown in Figure 5, we directly estimate the disparity map at the virtual viewpoint. For this case, we calculate data cost by using

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

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

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

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

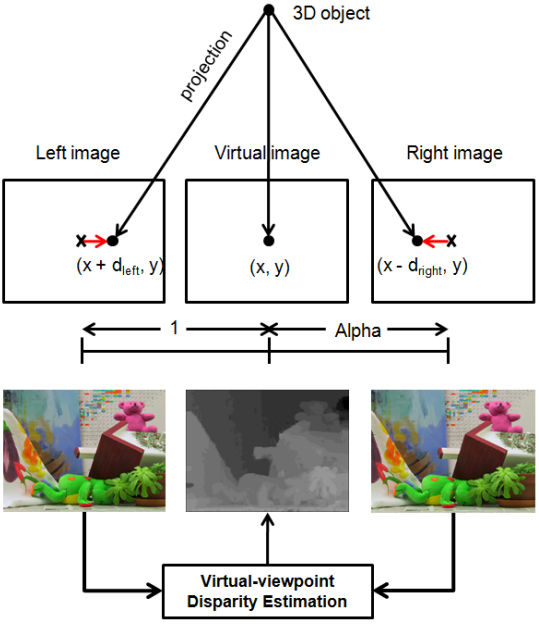


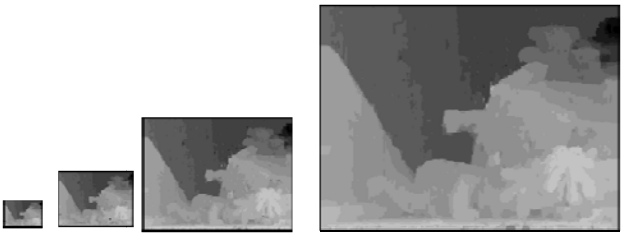
Fig. 5. Virtual viewpoint disparity estimation

The hierarchical belief propagation algorithm is used to minimize the energy function. It passes messages called belief around in four adjacency image grids. 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.

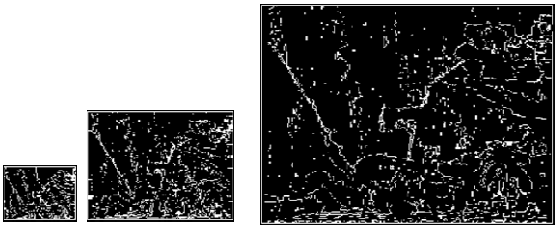
3.2 Acceleration of HBP Using Convergence Check

We can find convergence regions of message in the HBP. After each level shifting, we can find it by comparing midterm result of the HBP at each level. Then, we can stop message updates at convergence regions, if differential values of the message are lower than a predetermined threshold value. However, it takes non-slight additional time to check convergence regions due to many message paths (four directions, up, down, left, and right, message passing paths).

In order to obtain the convergence map efficiently, we compute beliefs (disparity map) for each stage and we check whether it stationary during a stage transition. Actually, beliefs are calculated only at a zero stage in the standard HBP algorithm. After we build the convergence map, we subtract converged nodes from the message update process. Figure 6(a) shows computed beliefs for each stage which is 'Teddy' which is given by the Middlebury web site. Figure 6(b) are convergence maps. In the convergence map, black and white pixels mean converged region and non-convergence region.



(a) Computed beliefs for each stage



(b) Binary convergence map (block: converged, white: non-converged)

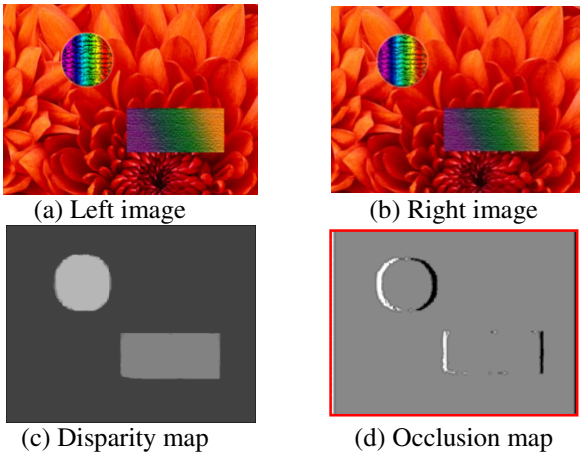
Fig. 6. Convergence check of HBP

3.3 Occlusion Decision and Backward Warping

If we have the virtual viewpoint disparity map, we can make the virtual viewpoint image using a backward warping process. Before we warp reference images to virtual viewpoint image plane, we have to consider possible errors due to occlusion regions. In order to avoid a occlusion problem of backward warping, we need to check occlusion regions of the virtual viewpoint from left and right viewpoints. So, we decide an occlusion map which is composed with four labels.

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

The occlusion map can be labeled by forward warping of a virtual viewpoint disparity map to left and right viewpoints. Label *A* can be selected when pixel information of the virtual viewpoint is only occluded from the left viewpoint. Label *B* can be selected when pixel information of the virtual viewpoint is not occluded from any viewpoints. Label *C* can be selected when pixel information of the virtual viewpoint is occluded only from the right viewpoint. In the case of label *D*, virtual viewpoint pixel information is occluded both viewpoints. Figure 7 shows an input stereo image pair, the virtual disparity map, and the occlusion map which has four labels (255-A, 128-B, 0-C, 1-D). Artificial images are used to see occluded regions clearly.



(a) Left image (b) Right image

(c) Disparity map (d) Occlusion map

Fig. 7. Occlusion map decision

3.4 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 fill the hole with 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.

3.5 GPU Implementation

For the real time implementation, we use the GPU parallel programming which executed on the GPU. 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 a data independency between all data executed simultaneously. We implement whole process with the parallel GPU programming while maintaining a data independency.

4 Experimental Results

In order to evaluate performance of proposed algorithm, we have implemented three methods (method A, method B, and proposed method) on CPU and additionally applied GPU parallel programming to proposed method. Because fast processing time and acceptable visual quality are key points of our algorithm, we measured processing time and visual quality by calculating PSNR value between original and output images. Furthermore, we check these measurements with other two methods. Method A and B are conventional methods. Method A interpolates the virtual viewpoint image using left and right disparity maps. Method B uses only a 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 to verify synthesis quality by comparing original images.

Table 1. Specification of the test stereo image set

Sequence	Teddy	Poster	Cones	News papers	Book arrivals
Size	640x480	480x416	480x416	640x480	640x480
max disparity	30	20	20	50	50

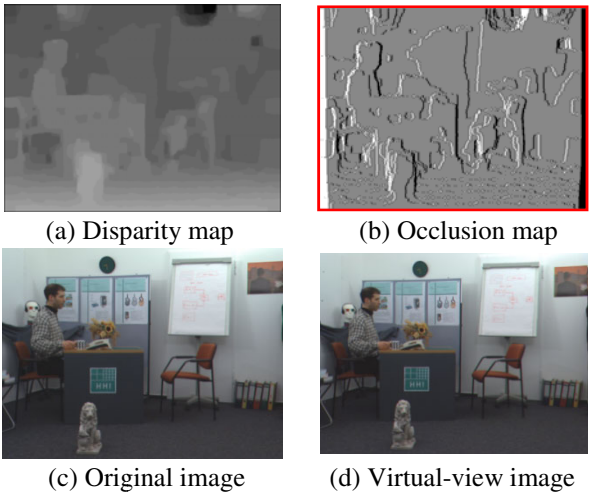


Fig. 8. Input and output images

Figure 8 shows input and output images of the proposed method. In order to investigate synthesis quality, we calculate PSNR values with synthesized images and original intermediate viewpoint images. Figure 9 and 10 shows performance comparison of the 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 11. We implement same algorithm on the CPU and GPU. However GPU execution speed is better than CPU because GPU execute multiple threads at the same time. As a result, GPU accelerates execution speed up to 30 times by comparing CPU execution time.

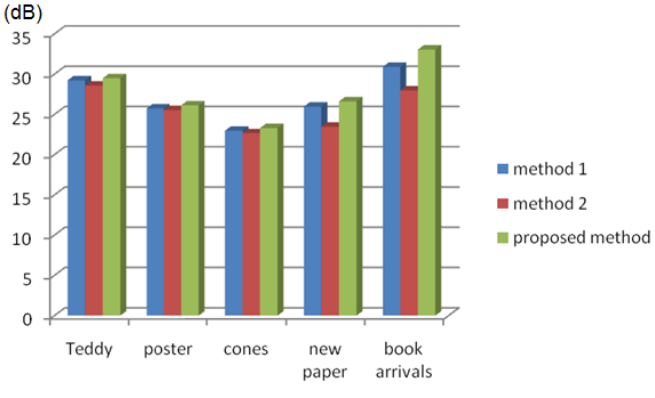


Fig. 9. PSNR comparisons of three methods

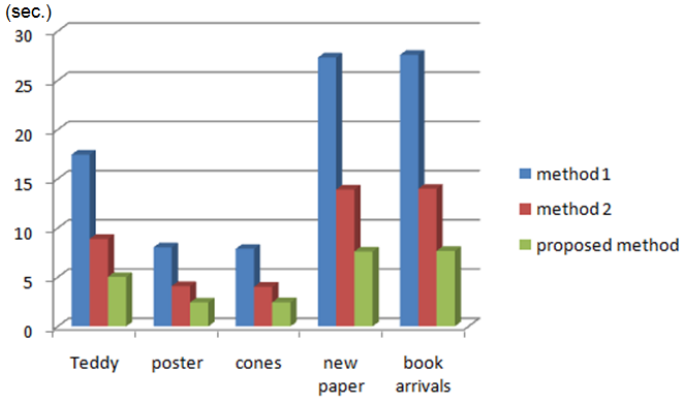


Fig. 10. Execution time of three methods

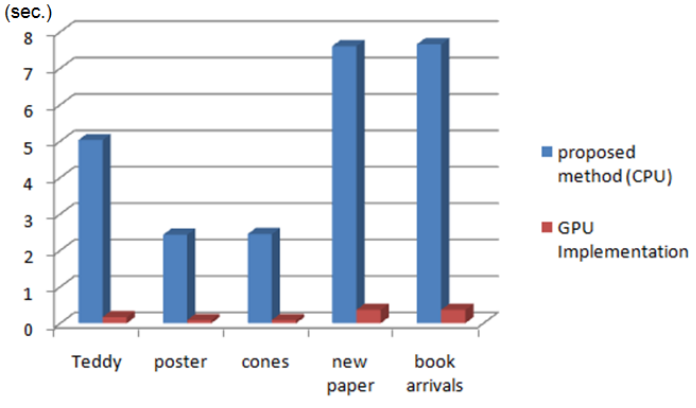


Fig. 11. Execution time in CPU and GPU

5 Conclusions

In this paper, we present the real time view interpolation method. In order to make it more rapidly, we apply the virtual viewpoint disparity estimation method and GPU parallel programming. Previous methods estimate some duplicated and unnecessary disparity values for the certain viewpoint. Thus, our proposed method reduces complexity and makes accurate synthesized images by eliminating surplus calculation. 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. In order to check a synthesized image quality, we calculate PSNR values by comparing original images and synthesized images. Our results are generally 0.3dB higher than previous method. For the real time implementation, we utilize the high speed GPU parallel programming

called CUDA. As a result, we can synthesize the virtual viewpoint image at a rate of 30 frames per second at most.

Acknowledgments. This research was supported by the MKE (Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agency) (NIPA-2011-(C1090-1111-0003)).

References

1. ISO/IEC JTC1/SC29/WG11 N6909: Survey of algorithms used for multi-view video coding, MVC (2005)
2. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE Transactions on Pattern Analysis and Machine Intelligence 23, 1222–1239 (2000)
3. Kolmogorov, V., Zabih, R.: Multi-Camera Scene Reconstruction via Graph Cuts. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2352, pp. 82–96. Springer, Heidelberg (2002)
4. Sun, J., Zheng, N., Shum, H.: Stereo matching using belief propagation. IEEE Transactions on Pattern Analysis and Machine Intelligence 25, 787–800 (2003)
5. ISO/IEC JTC1/SC29/WG11 M1537: Contribution for 3D Video Test Material of Outdoor Scene (2008)
6. Oh, J., Ma, S., Kuo, C.: Disparity estimation and virtual view synthesis from stereo video. In: Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS), New Orleans, LA, USA, pp. 993–996 (2007)
7. Tappen, M., Freeman, W.: Comparison of Graph Cuts with Belief Propagation for Stereo. In: Proc. IEEE Int'l Conf. Computer Vision, vol. 1, pp. 508–515 (2003)
8. Felzenszwalb, P., Huttenlocher, D.: Efficient Belief Propagation for Early Vision. In: CVPR, vol. 1, pp. 261–268 (2004)
9. Yang, Q., Wang, L., Yang, R., Stewenius, H., Nister, D.: Stereo matching with color-weighted correlation, hierarchical belief propagation and occlusion handling. IEEE Transactions on Pattern Analysis and Machine Intelligence 31, 492–504 (2009)
10. Oh, K., Yea, S., Ho, Y.: Hole Filling Method using Depth Based In-painting for View Synthesis in Free Viewpoint Television and 3-D Video. In: Picture Coding Symposium, pp. 39 (1-4) (2009)
11. Oliveira, M., Bowen, B., McKenna, R., Chang, Y.: Fast Digital Image Inpainting. In: Proceedings of the International Conference on Visualization, Imaging and Image Processing, Marbella, Spain (2001)
12. NVIDIA Corporation, CUDA 3.2 Programming Guide (2010), `http://www.nvidia.com/cuda_develop.html`