# A DEPTH MAP REFINEMENT ALGORITHM FOR 2D-TO-3D CONVERSION

Yu-Lin Chang, Yu-Pao Tsai, Te-Hao Chang, Ying-Rui Chen, and Shawmin Lei

MediaTek Inc., Taiwan No. 1, Dusing 1st Rd., Hsinchu Science Park, Hsinchu City 30078, Taiwan

# ABSTRACT

For showing 2D video contents on 3DTVs, 2D-to-3D conversion is required to convert 2D contents to 3D ones. In general, the conversion process consists of depth map generation, which estimates the 3D geometry of the scene, and rendering, which produces output stereo images. We propose a depth map refinement algorithm which uses adaptive decimation and guided interpolation to refine the depth map. The proposed approach eliminates unnecessary textures and keeps object boundaries on the depth map. In the refined depth map, the transition of depth values within an object is smooth, the object boundaries on the depth map are aligned to those on the input image, and the foreground can be clearly separated from the background. These characteristics on the depth map help to achieve better 3D visual perception. Compared to the state-of-the-art approaches, the blurriness and object boundary alignment are easier to be adjusted by the proposed two-stage operations, and the computational complexity of the proposed algorithm is much lower.

*Index Terms*—2D-to-3D conversion, depth map generation, 3D reconstruction, 3DTV

# 1. INTRODUCTION

Since the success of the movie Avatar, which was released in 2009, 3D has enjoyed growing popularity. Almost all the TV manufacturers put 3D functionality into their high end TV products. One of the important required 3D techniques is 2D-to-3D conversion, which converts the traditional 2D videos into 3D ones. It is important because most contents are still in the traditional 2D format. A usual processing flow for 2D-to-3D video conversion is shown in Fig. 1. For a 2D monocular video input, objects and their geometry perspective information are estimated and modeled, and then a depth map can be generated. With the produced depth map, depth image based rendering (DIBR) can then convert the original 2D monocular video to stereoscopic videos for the left and right eyes, respectively. In this processing flow, the most important issue is how to generate the depth map.



Fig. 1 The usual processing flow for 2D-to-3D conversion

In order to correctly generate the depth map of the input 2D video, various cues are applied to estimate the depth information. Many depth generation methods were proposed to retrieve the depth information using different combinations of depth cues. Depth values should be similar within the same object, and the object/background separation regions should be aligned with object boundaries to provide better visualization effect. Machine learning algorithms were proposed to extract features to depth information [1] [3]. Segmentation based approaches were proposed to separate the objects from background. Color segmentation [4] and motion segmentation [5] were used to find out the objects and their possible depth map. The machine learning and segmentation based approaches give good object boundaries on the depth map but require too much computational power. It is also hard to keep the temporal stability for segmentation based approaches. They both produce noisy depth map due to the limitation of segmentation and machine learning based approaches.

Computed image depth (CID) approaches were proposed to convert the image characteristic such as contrast and sharpness to depth values [2]. The CID approaches produce a rough depth map from local statistics in an image. CID becomes a popular way of generating the depth values for 2D-to-3D conversion owing to its low complexity and smooth 3D effect. It eliminates unnecessary textures in the image through block-based operations, but the object boundaries are also blurred. With blurred object boundaries on the depth map, the depth perception is reduced. To enhance the stereoscopic effect, the object boundaries should be distinguishable on the depth map. To align the object boundary in the depth map, the bilateral filter based approaches were proposed. The bilateral filter was used as a post-processing procedure to make the depth values similar within the same object [6] [7]. The bilateral filter helps to refine the blurriness and object boundaries. However, it is hard to adjust the parameters to achieve both aggressive smoothing and edge preserving simultaneously. Moreover, the Gaussian filters in bilateral filter require too much computational power.

In this paper, we proposed a depth map refinement algorithm using two-stage operations – adaptive decimation and guided interpolation. The low-pass filters used for decimation and adaptive filters used for interpolation are designed to control both the blurriness and object boundaries with low computational complexity. Each of the operation handles different type of tasks to produce a depth map aligned with object boundaries. This paper is organized as follows: The depth map refinement algorithm is described in Section 2. In Section 3, the experimental results of the proposed algorithm are shown. A complexity analysis is discussed in Section 4. Finally, a conclusion is given in Section 5.

## 2. DEPTH MAP REFINEMENT BY ADAPTIVE DECIMATION AND GUIDED INTERPOLATION

The proposed depth map refinement approach is shown in Fig. 2. The proposed approach is divided into two stages: adaptive decimation and guided interpolation. In the following sub-sections, the algorithm for obtaining initial depth map is described first. The adaptive low-pass filtering and decimation are then presented. Finally, the guided interpolation which reconstructs the decimated depth data with consideration to the object boundary is discussed.



Fig. 2 Flowchart of the proposed depth map generation approach



Fig. 3 An example: (a) Initial depth map I(x,y) (b) Adaptive decimated data h(x,y) (c) Guided interpolated data

## A. Initial Depth Map

An initial depth map can be generated by the previous approaches such as CID or segmentation based approaches. We use a CID-like depth mapping approach to map the depth values. CID generates rough depth values for objects [2]. One may use CID or other kinds of approach such as segmentation based approaches or machine learning approaches for the initial depth map.

## B. Adaptive Decimation

After the initial depth map is obtained, we use an adaptive low-pass filter to eliminate unnecessary details on the mapped depth values and to reduce the noise in the initial depth map. We designed several kinds of different low-pass filters which correspond to different texture complexities. The adaptation of the low-pass filters is based on the analysis of the texture complexity. The texture complexity is presented by the sum of Sobel edge gradients here. The edge gradient is defined as in the following equation:

$$G(p) = \sum_{q \in W} L(q)I(p+q),$$

where G(p) is the edge gradient at location p, L(.) is the edge filter coefficient function, W is the footprint of the edge filter, and I(p) is the luma value of pixel p. The texture complexity is classified by thresholding. When the texture complexity is high, the pass-band of the low-pass filter should be low to reduce noise in the depth map. When the texture complexity is low, the pass-band of the low-pass filter should be higher to preserve more details.

The adaptive low-pass filter procedure is combined with downsampling as shown in the following equation to form an adaptive decimation:

$$D_{deci}(p) = \sum_{q \in W} h(Mp+q) D_{init}(Mp+q),$$

where  $D_{deci}(p)$  denotes the values on decimated coarse depth map, M is the decimation scale, W is the footprint of the adaptive low-pass filter,  $D_{inid}(p)$  is the depth value on the original input depth map, and h(p) is the adaptive low-pass filter. In our implementation, the adaptation of M and lowpass filter selection is at frame level. M can be 8, 16, 32, and 64. Once the texture complexity is calculated, M and its corresponding low-pass filter are determined by a mapping function which is obtained from an offline training process. An example of the proposed approach is shown in Fig. 3. The initial noisy depth data in Fig. 3(a) is filtered and decimated by 4x4 (M=4) as shown in Fig. 3(b). After the decimation, a coarse depth map is obtained.

### C. Guided Interpolation

After the adaptive low-pass filter and decimation, we propose the guided interpolation to reconstruct a depth map that is consistent with object boundaries. The depth interpolation we used is a bilinear interpolation which has a triangular kernel [9]. One may also change the triangular kernel to other kernels such as Lanczos kernel to get better smooth areas. An example of the interpolation scheme is shown in Fig. 4. The interpolation output is defined as the following equation:

$$D_{final}(p) \approx \sum_{q \in N(p)} g(F(p,q)) l(p,q) D_{deci}(q),$$

where f(p) is the interpolated depth value at location p, N(p) is the four p-neighboring pixels on the decimated grid,  $D_{deci}(q)$  is the depth value at location q on the coarse depth map, and l(p,q) represents the reconstruction filter coefficient. The reconstruction filter coefficient is expressed

as the inverse of the distance to the decimated grid, i.e. the phase of the bilinear interpolation here. g(F(p,q)) is a guided weighting function which may be defined as the guided image filter in [8], Gaussian function, or trained Wiener filter. Here we use a Gaussian function  $e^{-a|F(p,q)|} / w$  from part of the bilateral filter as g(F(p,q)). F(p,q) denotes the image characteristics at location p and q. We use the difference of intensity values in the original image, I(p)-I(q), as F(p,q) in our implementation. w is a normalization factor for the guided weighting function. This guided weighting function generates similar depth values for the neighboring pixels with similar intensities.



Fig. 4 Guided interpolation scheme, p=(x,y),  $q_{11}=(x_1,y_1)$ ,  $q_{12}=(x_1,y_2)$ ,  $q_{21}=(x_2,y_1)$ ,  $q_{22}=(x_2,y_2)$ 

Owing to the decimation and guided interpolation, the computational complexity is quite low compared to the bilateral filter based approaches. The results of the proposed guided interpolation of the example in Fig. 3(b) are shown in Fig. 3(c). The noise in the depth data in Fig. 3(a) is reduced, and the depth map is aligned with object boundary.

After the depth map is generated, the stereoscopic video is rendered by depth image based rendering (DIBR).

### **3. EXPERIMENTAL RESULTS**

We use two video clips and two images from Kodak photo database to demonstrate the results of the depth map refinement. The low-pass filter footprint for M=32 is 48x48. We set 32x32 as the Gaussian window size in the bilateral filter based approach for comparison. The first video clip shown in Fig. 5(a) is the opening video of a cartoon – One Piece from YouTube. The result of the proposed approach in Fig. 5(d) shows that our depth map refinement works well even for synthetic video. The object separation effect is clearer than the bilateral filter based approach in Fig. 5(c). Our proposed approach produces similar depth values for the whole sail, while the false line structures of the sail on the ship can be observed in Fig. 5(c). The second video clip is a movie trailer - Red Cliff from YouTube. The soldiers in the same row are on the same depth layer for the proposed approach.

In the results of Kodak photo database shown in Fig. 7 and Fig. 8, we can find that the bilateral filter based approach, as shown in Fig. 7(c) and Fig. 8(c), produces false depth values on the tides of the sea and the roof of the

house. The textures are not blurred due to their high luma difference. We had tried many different correlation factors for the bilateral filter. Even if we use lower luma correlation for the bilateral filter as shown in Fig. 7(b) and Fig. 8(b), the depth values are still wrong since the luma difference is too high. The results from the proposed approach shown in Fig. 7(d) and Fig. 8(f) provide better results which make objects more distinguishable from each other compared to the blurred results produced by the bilateral filter. We change the M from 32 to 16 and 8 to preserve more details as the results shown in Fig. 8(d) and Fig. 8(e). No matter what M is chosen, the objects on the depth map are always consistent with their shape boundary. By using the proposed approach, one can adjust the pass-band of the low-pass filter in the first stage to eliminate unnecessary textures and recover the object boundaries in the second stage easily.

#### 4. COMPLEXITY ANALYSIS

The complexity of the proposed approach is analyzed by operations for one pixel. As for window size 32x32, the pixel operation required for the bilateral filter based approach is as following:

$$t_{total} = t_{G32x32} + t_{G32x32} + 32 \cdot 32 \cdot t_{mul} + 32 \cdot 32 \cdot 2 \cdot t_{add} + t_{div}$$

where the  $t_{G32x32}$  is the time used by Gaussian filter with window size 32x32. The  $t_{mul}$ ,  $t_{add}$ , and  $t_{div}$  denote for the time used by multiplication, addition, and division. For our proposed approach with decimation scale M=32, the required pixel operation is as following:

$$t_{total} = \frac{1}{32 \cdot 32} t_{L48x48} + t_{G2x2} + 4 \cdot t_{mul} + 3 \cdot 2 \cdot t_{add} + t_{div},$$

where the  $t_{L48x48}$  is the time used by the corresponding lowpass filter with footprint 48x48. The reduction of the complexity is quite large due to the decimation in the first stage of the proposed approach. The operations of Gaussian filter and multiplication are highly reduced. The experiment shows that there is at least 8 times of run time difference between the two approaches. With the proposed approach, real-time and parallel operations for both software and hardware are more feasible.

#### 5. CONCLUSIONS

We proposed a depth map refinement algorithm for 2Dto-3D conversion, which includes adaptive decimation and guided interpolation. The adaptive decimation helps to generate smooth depth map and to greatly reduce computations. The guided interpolation makes the depth of the objects consistent with their shape boundary. The subjective quality of this 2D-to-3D conversion is comparable to the state-of-the-art approaches [6][7], and the whole algorithm is flexible to be adjusted to different styles. The computational complexity of the proposed approach is much lower than the bilateral filter based approaches. The algorithm can also be used to refine the depth map generated by depth sensor or stereo matching.

#### 6. REFERENCES

- P. Harman, J. Flack, S. Fox, and M. Dowley, "Rapid 2d to 3d conversion," in *Proc. SPIE Stereoscopic Displays and Virtual Reality Systems IX* 4660, pp. 78~86, May 2002.
- [2] H. Murata, Y. Mori, S. Yamashita, A. Maenaka, S. Okada, K. Oyamada, S. Kishimoto, "A Real-Time 2-D to 3-D Image Conversion Technique Using Computed Image Depth, " *SID SYM*, Vol.29, pp.919-922, 1998
- [3] Mahsa T. Pourazad, Panos Nasiopoulos, and Ali Bashashati, "Random Forests-Based 2D-to-3D Video Conversion," in *IEEE International Conference on Electronics, Circuits, and Systems (ICECS)*, Greece, Dec. 2010.
- [4] S. Battiato, S. Curti, M. L. Cascia, M. Tortora, and E. Scordato, "Depth map generation by image classification," *Proceedings of SPIE*, *Three-Dimensional Image Capture and Applications VI* 5302, pp. 95~104, Apr. 2002.
- [5] K. Moustakas, D. Tzovaras, and M. G. Strintzis, "Stereoscopic video generation based on efficient layered structure and motion estimation from a monoscopic image sequence," in *IEEE Transactions on Circuits and Systems For Video Technology* 15, Aug. 2005.
- [6] Ludovic J. Angot, Wei-Jia Huang and Kai-Che Liu, "A 2D to 3D video and image conversion technique based on a bilateral filter," in *Proc. of SPIE-IS&T Electronic Imaging*, Vol. 7526, 2010
- [7] Chao-Chung Cheng, Chung-Te Li, and Liang-Gee Chen, "A novel 2D-to-3D conversion using edge information, " in *IEEE Transactions on Consumer Electronics*, Vol. 56, No. 3, Aug. 2010
- [8] Kaiming He, Jian Sun, and Xiaoou Tang, "Guided Image Filtering, " in the 11th European Conference on Computer Vision (ECCV 2010)
- [9] "Bilinear Interpolation, "*Wikipedia, The Free Encyclopedia.* Wikimedia Foundation, Inc. Web. 20 Sep. 2011













Fig. 6 (a) The original image of Red Cliff (b) the red-cyan stereoscopic video results of the proposed approach (c) the depth map by the bilateral filter based approach (d) the depth map by the proposed approach





(c)

(d)

Fig. 7 (a) The original image from Kodak photo database (b) the depth map generated by bilateral filter with low luma correlation (c) the depth map generated by bilateral filter with high luma correlation (d) the depth map of the proposed approach



Fig. 8 (a) The original image from Kodak photo database (b) the depth map generated by bilateral filter with low luma correlation (c) the depth map generated by bilateral filter with high luma correlation (d) the depth map of the proposed approach with M=8 (e) the depth map of the proposed approach with M=16 (f) the depth map of the proposed approach with M=32