

# DEPTH FILTERING FOR AUTO-STEREOSCOPIC MOBILE DEVICES

Virginia Martín, Julián Cabrera and Narciso García

## ABSTRACT

In this work we address a scenario where 3D content is transmitted to a mobile terminal with 3D display capabilities. We consider the use of 2D plus depth format to represent the 3D content and focus on the generation of synthetic views in the terminal. We evaluate different types of smoothing filters that are applied to depth maps with the aim of reducing the disoccluded regions. The evaluation takes into account the reduction of holes in the synthetic view as well as the presence of geometrical distortion caused by the smoothing operation. The selected filter has been included within an implemented module for the VideoLan Client (VLC) software in order to render 3D content from the 2D plus depth data format.

**Index Terms** — auto-stereoscopic, depth-map, DIBR, disparity, Real3D, smoothing, streaming, video, virtual, VLC

## 1. INTRODUCTION

The spread of Wifi/3G over mobile terminals is one of the keys for the growing popularity of Internet video. The integration among the mobile devices, Internet and wireless communications allows the transmission of large amounts of information over the network. In addition, the latest advances in processing capacity, the new compression technologies and the broadband networks have led to the delivery of real-time multimedia services over Internet. Services based on video streaming and real-time applications are constantly demanded, for example, Internet TV, distance education (e-learning) or real-time meetings.

On the other hand, 3D technology has had an upward trajectory in recent times. This fact is endorsed by the large number of movies whose scenes have been captured or can be watched using 3D techniques, and televisions that include this technology. Even the smartphone market has bet on tridimensional contents, with the emergence of auto-stereoscopic displays for mobile devices. This kind of displays are a great chance to achieve a good user experience in 3D content viewing, without the need the use of 3D glasses.

There are different formats for 3D video broadcasting. As a first approach, frame compatible formats, such as, side by side or top and bottom were adopted with the aim of making compatible broadcasting 3D content with the existing deployed 2D equipment. Later on, other formats more appropriate for representing 3D contents have been adopted for specific technologies or applications, such as, 2D plus depth map (one texture view plus a depth map) used to feed autostereoscopic devices.

We address a scenario where 3D content is transmitted to a mobile terminal with 3D display capabilities. We consider the use of 2D plus depth format to represent the 3D content, since it

can offer a better compression efficiency compared to the use of frame compatible formats. In addition, the rendering of the 3D content allows the adjustment of the degree of perceived depth, minimizing the visual fatigue that is common in 3D content viewing, as well as it offers a greater flexibility to render images from different viewpoints. On the contrary, the rendering of the 3D content will require a larger processing capability to generate the corresponding synthetic views.

In this paper, we focus on the generation of synthetic views in mobile devices. More specifically, we tackle the preprocessing stage of the depth maps with the aim of improving the quality of the synthetic views, taking into account that these devices have a limited computational capacity. We analyze the behavior of different low complexity smoothing filters in terms of decreasing the number of holes that emerge in these virtual images after rendering process and the geometric distortions that appear as result of a smoothing process. Finally, the selected filter has been included within an added module for the VLC software for Android in order to allow the VLC player to render 3D content from the 2D plus depth data format. This implementation has been tested in a LG Optimus 3D Max P720.

The outline of the paper is as follows. Section II contains a brief introduction of the modules integrated into VLC application. In section III we describe the features of the filters. Section IV presents the results of the evaluation. Finally, section VI concludes the paper.

## 2. ADAPTATION OF VLC PLAYER

VLC media player [1] is a free, open source and cross-platform multimedia player that allows the playback of most multimedia files of 2D media content. Besides, it can be used as a server and as a client to stream and receive network streams using some of the following streaming protocols: HTTP, RTSP or MMS.

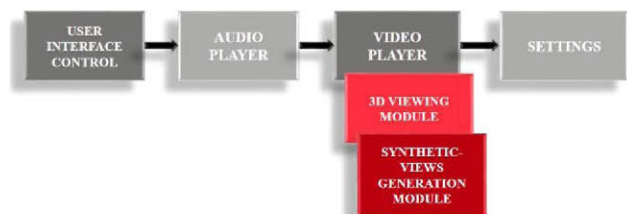


Figure 1. Block diagram of VLC player

In order to provide VLC with 3D rendering capability for 2D+depth content, the following modules have been implemented:

- A **3D viewing module**, whose function is the suitable configuration of the resources related to the presentation of 3D multimedia content in the mobile terminal selected: LG Optimus 3D Max P720.

- A **synthetic-views generation module**, whose diagram is presented in Figure 2. This system provides the left and right views from each video frame encoded in the 2D plus depth map format that the streaming server sends the client. These two views form the side by side frames that the LG device needs as 3D input content.



Figure 2. Functional Flow Block Diagram of the module of generation of synthetic views

View synthesis methods allow to locate virtual cameras in different positions in order to obtain images of the same scene from new perspectives. In particular, the DIBR method ([2],[5]) carries out the generation of the virtual stereoscopic images from a centered image and its associated depth map

Inpainting techniques [6] are applied with the aim of filling the dis-occluded regions. During the rendering process, a mask is generated in order to identify which pixels must be reconstructed, since no value has been assigned them. The unknown regions are restored using data from the region neighborhood in the process of hole-filling. This method handles the holes that emerge in the synthetic image successfully. The bigger the size of the region neighborhood is, the higher the computational cost is.

Finally, the side by side image is formed using the received texture image as the left view and the synthetic view that has been generated by DIBR method as the right view. Although it is possible to generate one or more virtual images from an original centered image and its depth map, we have considered that the image captured with a real camera will be of better quality than any other generated synthetic image.

### 3. PREPROCESSING OF THE DEPTH MAP

The most significant problem of DIBR methods used in the rendering process is the emergence of dis-occluded regions in the virtual images. These regions were hidden from the original viewpoint, but they become visible from the new viewpoint and no data are available for them.

The smoothing of the depth map [2] allows to reduce the artefacts resulting from the block effect, decrease the distortions included in the depth map and smooth the regions where the holes will emerge predictably in the synthetic images. Therefore, the objective of this process is the minimization of the number and size of these newly exposed areas. For that, the filters implemented smooth the sharpness of the depth edges, where many of these areas without information are located. Below we describe each one of the filters considered and the parameters used in their implementation.

#### 3.1 Symmetric Gaussian filter

This filter performs a smoothing operation with the same smoothing strength in the vertical and horizontal directions. According to the documentation [2], pixels that are located beyond three times the value of the standard deviation of the Gaussian filter from the center pixel have an insignificant influence in the value of the smoothed pixel. Therefore, the relationship between the filter's window size and the standard deviation is  $\omega = 3 \cdot \sigma$ , where  $\omega$  is previously fixed.

The smoothing strength is chosen between a medium effect (with window sizes equal to 11 or 13) and a strong effect (with values close to 61), in accordance with the distance between the

real and virtual cameras, known as the baseline distance. Medium strength will be chosen, if we take a value of the baseline distance close to the human inter-ocular distance (approximately 65 mm); while we will choose a higher smoothing strength, if the distance is quite larger. This is due to that an increase of the baseline distance involves a rise of the number of geometric distortions and dis-occluded areas.

Figure 3 shows the original depth map and the depth map after symmetric Gaussian filter.

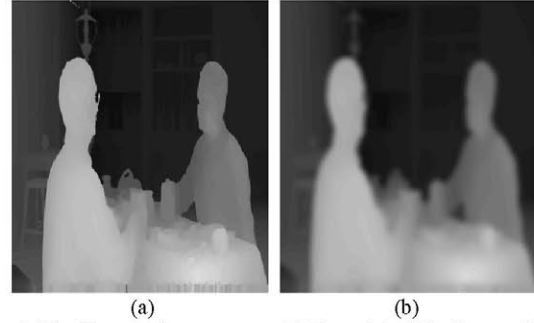


Figure 3. The Beer-garden sequence: (a) The original depth map; (b) the depth map after symmetric Gaussian filter  $\sigma_\mu = \sigma_\nu = 4$  and  $\omega_\mu = \omega_\nu = 13$ .

#### 3.2 Asymmetric Gaussian filter

The larger dis-occluded regions and the geometric distortions appear more frequently in the horizontal direction, because the virtual images are generated through a horizontal shift of the pixels of the original texture image. In order to improve the virtual image quality, we have performed an asymmetric Gaussian smoothing ([2], [3]), taking in account that the smoothing strength of depth maps in the horizontal direction should be less than the smoothing in the vertical direction. In this way, the vertical objects of the scene will have similar depth values along the vertical dimension after this process of smoothing of the depth map.

Generally, the standard deviation value in the vertical direction is three times the standard deviation applied in the horizontal direction:  $\sigma_\nu = 3 \cdot \sigma_\mu$ . In addition the previous relation between the filter's window size and the standard deviation, which is used in the symmetric Gaussian filter, remains correct for this smoothing:  $\omega_\mu = 3 \cdot \sigma_\mu$  and  $\omega_\nu = 3 \cdot \sigma_\nu$ . The filtered depth map using the asymmetric Gaussian filter is shown in Figure 4.

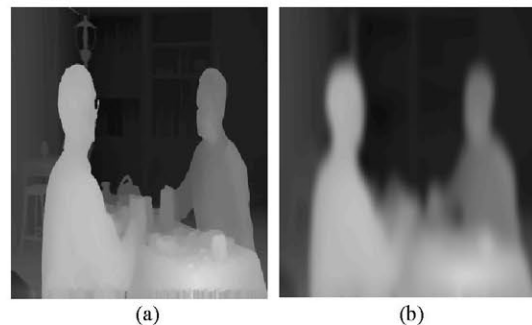


Figure 4. (a) The original depth map; (b) the depth map after asymmetric Gaussian filter with  $\sigma_\mu = 4$ ,  $\sigma_\nu = 12$ ,  $\omega_\mu = 13$  and  $\omega_\nu = 41$ .

#### 3.3 Edge dependent filter

This filtering method aims at preserving depth information in regions of interest, unlike the previous methods, in which the entire depth map is smoothed. This filter performs a smoothing operation only on the edges of the depth maps, keeping the depth value in the rest of areas which are not detected like object boundaries [4]. In that way, it produces a lower loss of the details of the depth maps and these high frequencies will enhance the desired 3D effect.

Figure 5 shows the block-diagram of the edge dependent filtering:



Figure 5. Functional Flow Block Diagram of the edge dependent filter

In the first stage we have computed a mask for the edges of the depth maps using the canny edge detector. This intermediate mask will be used as a reference to avoid an excessive filtered in those areas where there is no boundary. Then, a dilatation operation is applied to the previous mask in order to enlarge the size of the regions identified as edges. Finally, the modified depth map is computed as follows: the pixels located in the object boundaries will take the value of the corresponding pixels in the preprocessed depth map with a symmetric Gaussian filter and the rest of pixels will have the values of the original depth map. Figure 6 shows the intermediate images and the filtered depth map resulting from this smoothing process:

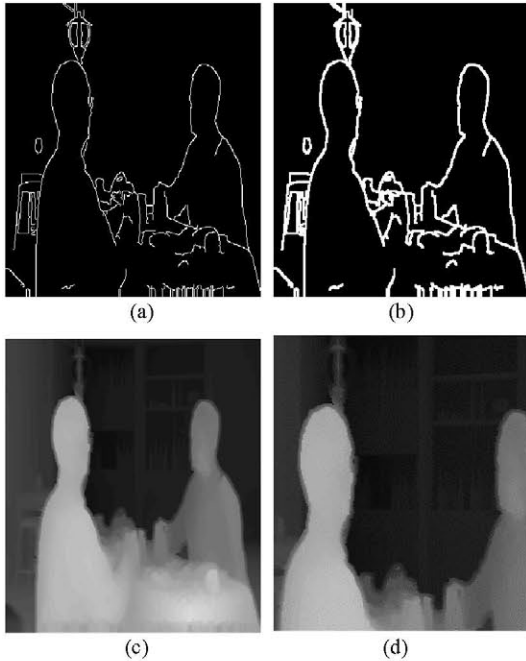


Figure 6. (a) The mask with the object boundaries; (b) the previous mask after dilatation operation; (c) the depth map after edge dependent filter; (d) Magnified image of a certain region of the previous depth map.

#### 4. EVALUATION

First of all, the chosen scenario involves some constraints. Each video frame encoded in the 2D plus depth map format is processed properly and its corresponding synthetic view is generated in real time. Since this process involves a high computational cost, and due to the low computational power of the mobile device, the frame-rate of the videos that have been used to evaluate the different filters is low: for the Beer-garden sequence is 1 fps and for the Pinocchio sequence is 5 fps.

In order to evaluate the implemented process and their integration into VLC application, we considered two aspects: the efficiency of the filters to decrease the number of dis-occluded regions and the emergence of geometric distortions in the synthetic images.

For the evaluation, we have used a medium smoothing strength since the synthetic views are rendered with a value of the

baseline distance equal to 65 mm. The standard deviation and the filter's window are the following:  $\sigma_v = \sigma_\mu = 4$  and  $\omega_\mu = \omega_v = 13$  for the symmetric Gaussian filter;  $\sigma_v = 4$ ,  $\sigma_\mu = 12$ ,  $\omega_\mu = 13$  and  $\omega_v = 41$  in the case of the asymmetric Gaussian smoothing and lastly, the Gaussian filter that is part of the edge dependent filter has the same parameters that the symmetric Gaussian filter.

As can be seen in Figure 7, the asymmetric Gaussian filter obtains the minimum area with holes that emerge in the virtual view, closely followed by the symmetric Gaussian filter. Regarding the edge dependent filter, it produces the bigger number of holes because some dis-occluded areas are located in regions that have not been considered as discontinuities or object boundaries and consequently, they have not been previously filtered. The virtual images after rendering process and before hole-filling process are presented in Figure 7:

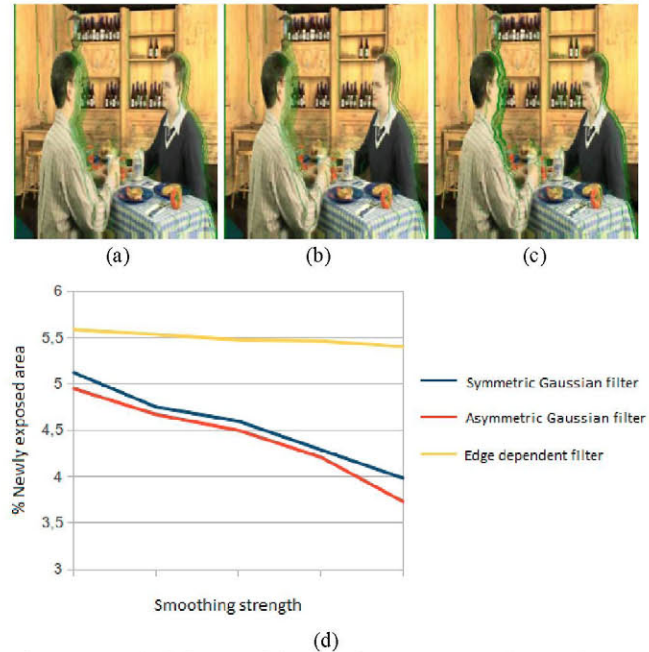


Figure 7. Synthetic images with holes after: (a) symmetric Gaussian filter; (b) asymmetric Gaussian filter; (c) edge dependent filter; (d) Relation between depth smoothing strength and newly exposed areas as a percentage of the total area.

Then we evaluated the emergence of geometric distortions for the different filters. The symmetric Gaussian filter produces a loss or blurring of the image details that penalizes the 3D effect. In addition, there are some geometric distortions in the vertical edges of the synthetic image that present a curved appearance. The reason of this curvature is that the vertical boundaries do not have consistent depth values along their vertical dimension. The asymmetric Gaussian filter provides virtual images with a greater level of detail, and in consequence the sense of depth is reasonably good and stable. However, the problem of the curved appearance of the vertical objects is not resolved, although it is less noticeable. Therefore, the drawbacks of this filter are the curved appearance of the vertical objects and the remarkable deformation of the depth map. And lastly, the edge dependent filter generates a synthetic image with the lowest presence of geometric distortions. This is due to the fact that edges are only smoothed and therefore, the rest of regions present a higher level of detail. The distortions in the vertical edges do not resolve completely, although with this filtering, this problem is minimized.

Figure 8 shows the different synthetic images resulting from a different pre-processing of the depth map and the same rendering process. Some vertical objects are highlighted in these images, since they are important due to their curved contour.

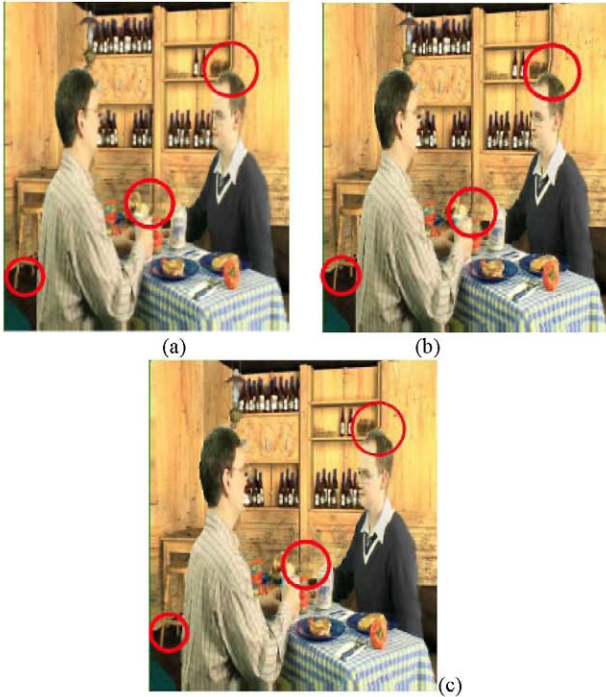


Figure 8. Synthetic images with some vertical edges highlighted: (a) symmetric Gaussian filter; (b) asymmetric Gaussian filter; (c) edge dependent filter.

In order to illustrate and analyze the problem of the curved appearance of the vertical objects, Figure 9 shows two magnified images of a certain region of the synthetic images after asymmetric Gaussian filter (left) and edge dependent filter (right). In the left image, the bottles present a distorted contour, since their pixels have suffered a shift towards the left direction. Moreover, one side of the shelf present discontinuities that will be annoying for the user. On the other hand, the virtual image corresponding to the edge dependent filter presents a higher definition in object boundaries and a more smoothed aspect than in the other image.



Figure 9. Magnified images of a certain region of the synthetic images after: (a) asymmetric Gaussian filter; (b) edge dependent filter.

For the video frame encoded in the 2D plus depth map format that is presented in Figure 10, the perceived differences in the synthetic images resulting from the different pre-processing of the depth map are limited due to the low definition that presents the background of this video sequence.



Figure 10. The Pinocchio sequence.

However when we enlarge them, it is possible to distinguish a higher definition in the object boundaries of the virtual image using the edge dependent filter again, as can be seen in Figure 11. In particular, the differences can be seen in the left side of Pinocchio's hat.

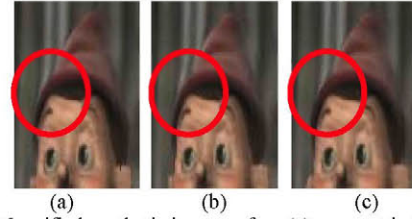


Figure 11. Magnified synthetic images after: (a) symmetric Gaussian filter; (b) asymmetric Gaussian filter; (c) edge dependent filter.

## 5. CONCLUSIONS

We have evaluated different filters used in the stage of pre-processing of the depth map in the real-time generation of synthetic views for a mobile device, with the aim of carrying out the implementation of the best filter. To accomplish this evaluation, we have performed an adaptation of VLC player that has consisted of the design and implementation of a 3D viewing module and a synthetic-views generation module.

Their behavior have been evaluated by measuring the efficiency in the reduction of the dis-occluded areas and the minimization of the geometric distortions.

On the one hand, the asymmetric Gaussian smoothing is the filter that better resolves the dis-occlusions problem; on the other hand, the edge dependent filtering generates a synthetic image with the least geometrical distortion of the implemented filters. In conclusion, the edge dependent filter has been chosen for its implementation into the synthetic-views generation module, since the presence of geometric distortions is significantly lower with this filter, while the difference between the number of holes reduced by this filter and the amount reduced by the asymmetric Gaussian filter is not very significant.

## 6. ACKNOWLEDGEMENTS

This work has been partially supported by the Ministerio de Economía y Competitividad of the Spanish Government under project TEC2010-20412 (Enhanced 3DTV).

## 7. REFERENCES

- [1] VLC Documentation. [Online]. Available: [www.videolan.org/developers/vlc/doc/doxygen/html](http://www.videolan.org/developers/vlc/doc/doxygen/html).
- [2] L. Zhang and T. Wa James, "Stereoscopic image generation based on depth images for 3D TV", *IEEE Transactions on Broadcasting*, vol.51, no.2, pp.191, 199, June 2005.
- [3] M. Koppel, M. Ben Makhoulouf and P. Ndjiki-Nya, "Optimized adaptive depth map filtering", in *Proc. of IEEE International Conference on Image Processing (ICIP)*, September 2013.
- [4] C. Hyun-Woong, C. Soon-Wook, S. Moon-Kyu Song and S. Woo-Jin, "Depth-image-based 3D rendering with edge dependent preprocessing", in *Proc. of 54th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pp.1,4, 7-10, August 2011.
- [5] C. Jung, L. Jiao, Y. Oh and J.K. Kim, "Depth-preserving DIBR based on disparity map over T-DMB", *IET Electronics Letters*, vol. 46, p. 628 – 629, April 2010.
- [6] L. Azzari, F. Battisti, A. Gotchev, M. Carli and K. Egiazarian "A modified non-local mean inpainting technique for occlusion filling in depth-image-based rendering", in *Proc. of SPIE Stereoscopic Displays and Applications XXII*, 78631C, February 2011.