

# Optimal Distribution of Computational Power in Free Viewpoint Interpolation by Depth Hypothesis Density Adaptation in Plane Sweeping

Patrik Goorts, Steven Maesen, Maarten Dumont,  
Sammy Rogmans, Philippe Bekaert

Hasselt University - tUL - iMinds, Expertise Centre for Digital Media  
Wetenschapspark 2, 3590 Diepenbeek, Belgium

**Abstract.** In this paper, we present a system to redistribute computational power in plane sweeping. Plane sweeping allows the generation of novel viewpoints of a scene by testing different depth hypotheses across input cameras. Typical plane sweeping approaches incorporate a uniform depth plane distribution to investigate different depth hypotheses to generate a depth map. When the scene consists of a sparse number of objects, some depth hypotheses do not contain objects and can cause noise and wasted computational power. Therefore, we propose a method to adapt the plane distribution to increase the quality of the depth map around objects and to reduce computational power waste by reducing the number of planes in empty spaces in the scene. First, we generate the cumulative histogram of the depth map of the scene. This depth map can be obtained from the previous frame in a temporal sequence of images, or from a depth camera with lower resolution or quality. Next, we determine a new normalized depth for every depth plane by analyzing the cumulative histogram. Steep sections of the cumulative histogram will result in a dense local distribution of planes; a flat section will result in a sparse distribution. The results, performed on controlled and on real images, demonstrate the effectiveness of the method over a uniform distribution and the possibility of using a lower number of depth planes, and thus a more performant processing, for the same quality.

## 1 Introduction

View interpolation is an important technique for computational video and photography, allowing the generation of novel viewpoints from a scene. A number of real cameras are capturing a scene. Using view interpolation, it is possible to generate images from non-existing camera views using the images from the real cameras. This can increase the user experience, for example in sports broadcasting [1, 2] and video conferencing [3].

View interpolation is typically achieved by using 3D reconstruction or by image-based rendering. 3D reconstruction estimates the geometry of the scene

and can choose the novel viewpoint accordingly. The most notable methods are visual hull [4, 5], photo hull [6] and space carving [7]. While 3D reconstruction allows a large range of novel viewpoints, the reconstruction is typically slow and the quality is limited to the quality of the reconstructed 3D models.

Image-based rendering, on the other hand, does not use geometry-based models of the scene. Instead, only the images are used to generate the novel image directly. The most known approach is the generation of depth maps for small baseline setups using stereo matching [8, 9] and plane sweeping [10, 3], including depth-selective plane sweeping for two views [11] or for large scenes [2].

In the plane sweep approach, the scene is divided in planes, all representing a depth hypothesis. The input cameras are projected to a plane, and backprojected to the virtual image plane. By comparing the photoconsistency of every depth plane for every pixel of the virtual image, an optimal virtual image with a depth map can be created. We propose a system to reduce the waste of computational power for the plane sweep approach.

Typically, the planes for the depth hypotheses are distributed evenly in the scene space, thus allocating uniform computational power to all depth hypotheses. Because the scene typically does not have a uniform distribution of objects, wasted performance can be perceived by considering depth values where no objects are present. Therefore, we present a system where the distribution of the planes is adapted to the scene. A histogram is calculated of the resulting depth map to determine the plane distribution for the next temporal frame. This will redistribute computational power to the more dense regions of the scene, and consequently increase the quality of the interpolation by reducing mismatches and noise.

The used histogram is obtained from the depth map of the previous frame, or from the depth map generated by an active depth camera. When using the depth map from the previous frame, no extra hardware is required. However, the depth hypotheses adaptation lags one frame. When using an active depth camera, no lag is present, but hardware considerations may limit the working range of the setup. We used the Microsoft Kinect [12] as active depth camera, limiting the depth range to 5 meters. Furthermore, the resolution and the image quality are low, such that the depth map cannot be used directly for view interpolation. However, the depth map is adequate to be used in our plane redistribution method, thus effectively creating a hybrid method between low resolution active depth cameras and high resolution multi camera depth methods.

Multiple methods have been proposed to reduce plane sweep complexity, reduce required computational power and increase quality. The method of Rogmans et al. [11] also uses a histogram to select applicable depth ranges, but without redistributing or changing the plane density. Gallup et al. [13] propose a histogram-based method to determine the optimal orientation of the planes to increase quality and reduce computational complexity, but also without optimizing for sparse scene regions.

The view interpolation system is achieved using commodity GPU hardware to acquire real-time processing. By redistributing computational power to signif-

icant parts of the scene, less power is wasted and more is available to other image processing stages, such as demosaicing [14], segmentation or depth filtering [2].

## 2 View Interpolation using Plane Sweeping

View interpolation allows the generation of novel views of a scene. We accomplish this by using the well-known plane sweep approach [15], implemented using traditional GPU paradigms to acquire real-time processing by leveraging the projective texturing capabilities of Cg shaders. We place a number of cameras directed at the scene and calibrate them to acquire the projection matrices  $P_i$  using the Multicamera Calibration Toolbox of Svoboda et al. [16]. Using those cameras, we can construct a novel viewpoint and generate the image thereof using a plane sweeping approach.

First, we divide the space in front of the virtual camera  $C_v$  into  $M$  planes on different depths  $D_i$  with  $D_{min} \leq D_i \leq D_{max}$ , parallel to the image plane of the virtual camera. Then, for every plane, we project the input camera images  $C_i$  to the plane, reproject them to the image plane of the virtual camera  $I_v$  and calculate the photoconsistency of every pixel on the virtual plane. This process is demonstrated in Fig. 1.

To acquire a metric for the photoconsistency, we use a cost function, aggregated over a window to improve quality. The cost function is defined as the sum of squared differences (SSD):

$$\sigma(x, y) = \sum_{i=1}^N \frac{\|\gamma - C_i(x, y)\|^2}{3N} \text{ with } \gamma = \sum_{i=1}^N \frac{C_i(x, y)}{N} \quad (1)$$

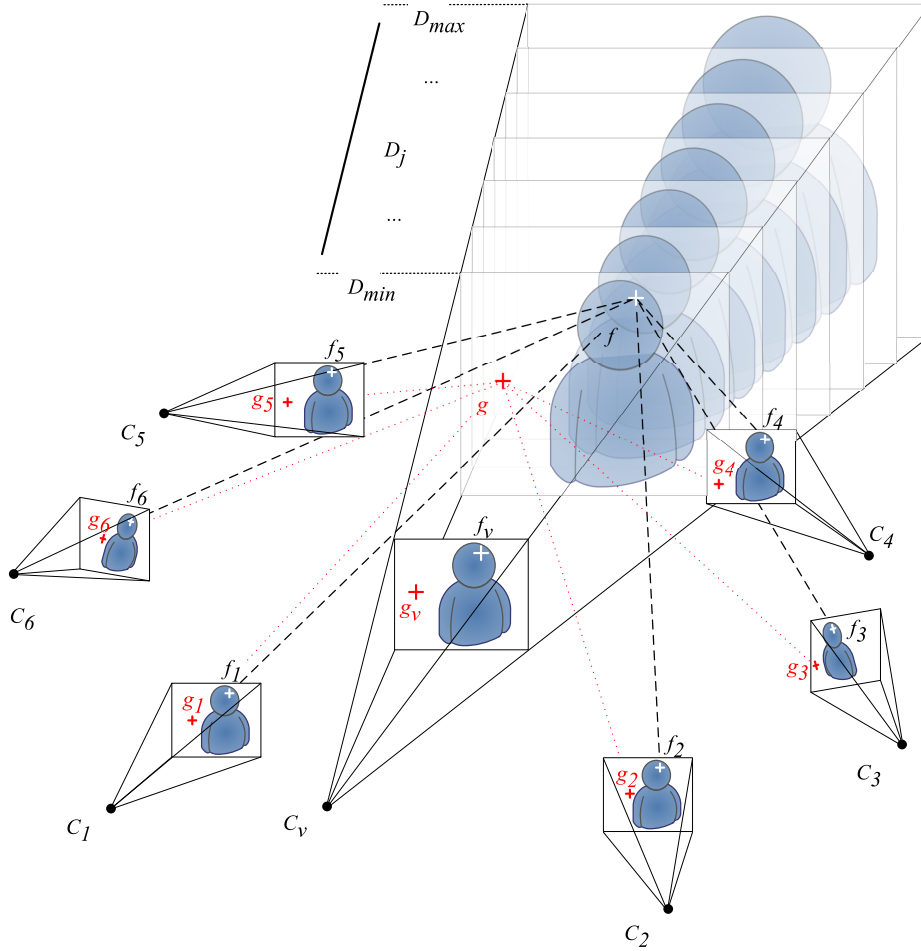
where  $\gamma$  is the average of the reprojected pixels and  $C_i$  is the  $i^{\text{th}}$  input image of total  $N$ . The final photoconsistency error is acquired by aggregating the SSD for the pixels in a fixed-size window, weighted by a (separable) Gauss filter:

$$\epsilon(x, y) = \sum_{u,v} w(u, v) \sigma(x + u, y + v) \quad (2)$$

for a window with coordinates  $u$  and  $v$ , centered at  $(x, y)$ , and Gauss weights  $w(u, v)$ .

The depth plane with the lowest error value  $\epsilon$  is chosen, thus selecting the depth value with the highest photoconsistency using a winner-takes-all approach. This will result in a simultaneous generation of the depth map and the final color,  $\gamma$ .

Quality is increased by using a foreground/background segmentation applied to the input images.  $\epsilon(x, y)$  is set to infinity when the projected pixels contains a background pixel. This will reduce mismatches caused by noise in the background and will reduce disappearance of foreground objects on a uniform background. This is due to the similarity of every pixel on a uniform background, thus obtaining a low error value  $\epsilon$  for incorrectly matched background pixels. This will



**Fig. 1.** Plane sweeping. The space in front of the virtual camera is divided in planes on different depths. The photoconsistency of every pixel for every plane is considered in the selection of the optimal depth plane, thus selecting the optimal depth and color for the virtual image.

result in the destruction of the foreground objects. These artifacts are greatly reduced by processing the foreground and the background independently.

In the traditional approach [15], the planes are distributed uniformly in the sweeping space. However, this will allocate computational resources to depth planes where no scene information is available. Therefore, we propose a method to reduce wasted computational power and increase quality in important regions of the scene.



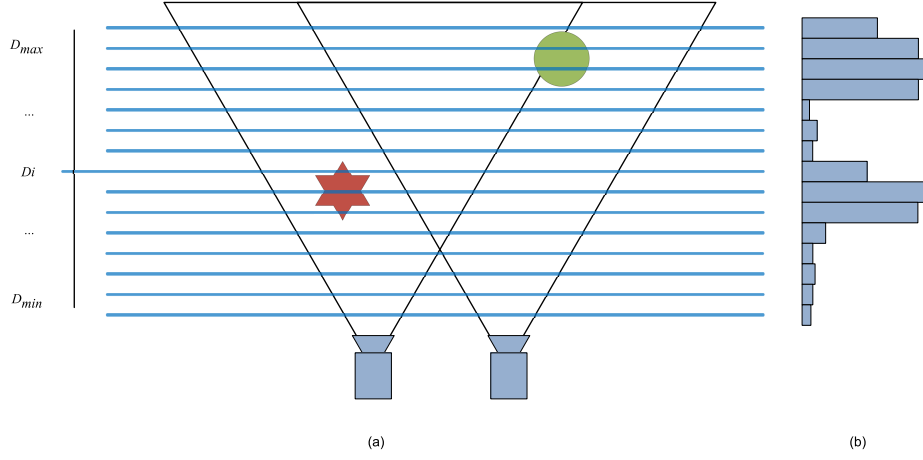
### 3 Adaptive Non-Uniform Plane Distribution

When the scene consists of a limited range of depths between  $D_{min}$  and  $D_{max}$ , some processing resources are allocated to depth planes where no scene is available. This is demonstrated in Fig. 2(a). Here, a lot of planes are placed in the scene where no objects are positioned. This will waste resources and introduce more noise due to mismatches between the cameras. Therefore, we rearrange the distribution of the depth planes to provide fewer planes in depth ranges with less object, and more, dense planes in scene regions with more objects. We determine the interest of a depth by analyzing the previous frame in a temporal sequence, or use the depth map of a depth camera.

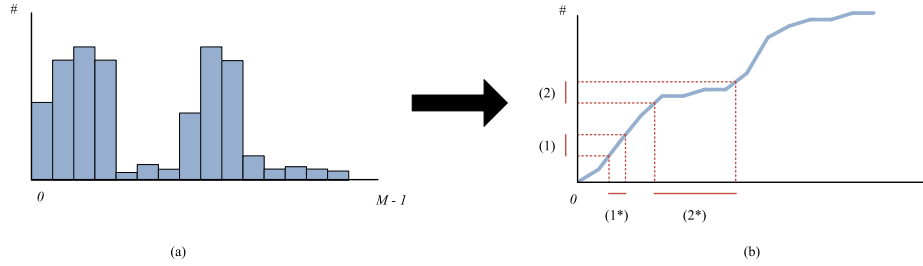
When using the depth map of the previous frame, we generate the histogram of the depth map using the well-known occlusion querying method [17] on GPU, allowing fast processing. The histogram can be seen in Fig. 2(b). The occurrence of every depth value, as determined by the depth of the depth planes, in the depth map is counted. The histogram will have discrete depth values between  $D_{min}$  and  $D_{max}$ , represented by the depth plane numbers, because there is a limited number of planes. Scene depths of high interest will contain more depth values than depths of low interest. If there are depths in the scene where no objects are present, few of this depth values will be available in the depth map and this will be reflected in the histogram. In the next frame, we want to provide more planes in depth ranges where a lot of depth values can be found, thus where there are large values in the depth histogram. The depth planes are not necessary uniformly distributed, thus the histogram uses the depth plane number as the bin value, instead of the depth directly. When using the depth camera, the depth map is used directly. All depth values are converted to the corresponding depth values in the coordinate space of the color cameras before using them in the histogram creation. These depth values correspond to plane depths, thus creating an analogous histogram as the previously discussed method.

To use the depth distribution information, we convert the histogram to its cumulative version, as shown in Fig. 3. Here, we do not count the number of occurrences per depth value, but we also include the number of occurrences lower than this depth. Furthermore, we rescale the depth values from  $[D_{min}, D_{max}]$ , as represented by the depth plane numbers, to  $[0, 1]$ . This will transform the non-uniform distribution of the depth planes to actual normalized depth values between 0 and 1. This transformation will generate an increasing function  $H(x) = y$ , where  $x \in [0, 1]$  is a normalized depth value and  $y$  is the number of values in the rescaled depth map smaller or equal to  $x$ . For values of  $x$  where there are a lot of corresponding values in the depth map,  $H(x)$  will be steep. For values of  $x$  with a low number of occurrences,  $H(x)$  will be flat. Because of the non-uniform depth plane distribution as input,  $H(x)$  will be constant at some points where there were no depth planes for the corresponding normalized depth value.

We will use the cumulative histogram to determine a mapping of a plane number  $m$  with  $0 \leq m < M$  to a depth value  $D_m$  with  $D_{min} \leq D_m \leq D_{max}$ . For a uniform distribution, this would be:



**Fig. 2.** (a) Uniform plane distribution (b) Histogram of the depth values.



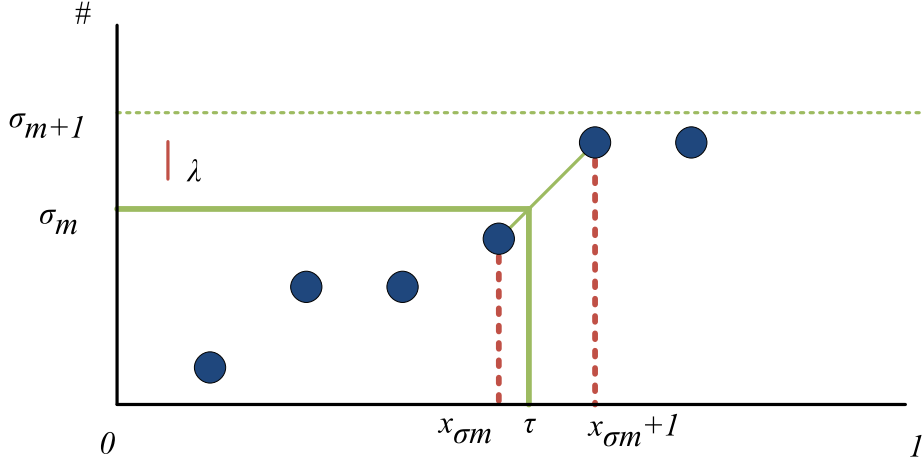
**Fig. 3.** (a) Resulting histogram (b) Corresponding cumulative histogram  $H(x)$ .

$$D_m = D_{min} + \frac{m}{M}(D_{max} - D_{min}) \quad (3)$$

We will adapt this uniform distribution method. When using the cumulative histogram to determine the distribution, we calculate a fraction  $\tau_m \in [0, 1]$  based on the plane number  $m$ , applied as follows:

$$D_m = D_{min} + \tau_m(D_{max} - D_{min}) \quad (4)$$

The fraction  $\tau_m$  is determined by the cumulative histogram. The  $Y$  axis is divided in  $M$  cross sections, with a distance  $\lambda$  from each other, where  $\lambda = \max(H)/M$ . Each cross section represents a depth plane  $m$ . The actual depth fraction  $\tau_m$  for each cross section  $\sigma_m$ , i.e. a depth plane, is calculated by first determining the depth value  $x_{\sigma_m}$  where  $H(x_{\sigma_m}) \leq \sigma_m$  and  $H(x_{\sigma_m} + 1) > \sigma_m$ . This is demonstrated in Fig. 4. Because the depth values  $x$  in the cumulative histogram are discrete, finding a value  $x_{\sigma_m}$  where  $H(x_{\sigma_m}) = \sigma_m$  is unlikely, and not desirable when generating planes that are dense, i.e. closer together, than the depth values provided in the cumulative histogram.



**Fig. 4.** Detail of the cumulative histogram with discrete values.  $\tau$  is calculated by determining  $x_{\sigma m}$  and  $x_{\sigma m} + 1$ , such that  $H(x_{\sigma m}) \leq \sigma_m$  and  $H(x_{\sigma m} + 1) > \sigma_m$ , where  $\sigma_m$  represents a depth plane number.

Once  $x_{\sigma m}$  is determined,  $\tau_m$  is calculated as follows:

$$\phi = \frac{m\lambda - H(x_{\sigma m})}{H(x_{\sigma m} + 1) - H(x_{\sigma m})} \quad (5)$$

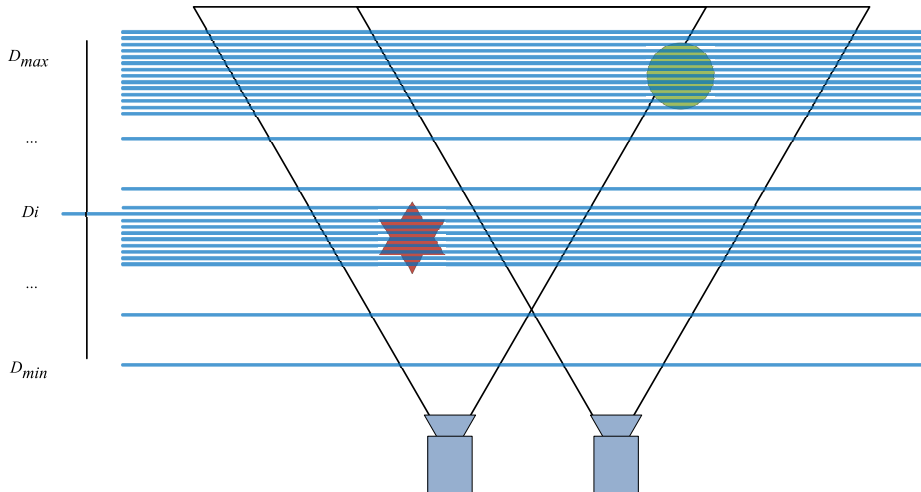
$$\tau_m = \phi(x_{\sigma m} + 1) + (1 - \phi)x_{\sigma m} \quad (6)$$

Figure 3(b) shows the transformation from a uniform depth plane distribution to a non-uniform distribution based on the cumulative histogram. In point (1), where the cumulative histogram is steep, there will be a dense plane distribution, as can be seen at (1\*). When the cumulative histogram is flat, a sparse plane distribution is acquired, as can be seen at (2\*).

Using  $\tau_m$ , an actual depth for every plane  $m$  ( $0 \leq m < M$ ) is determined and used in the plane sweeping step:

$$D_m = D_{min} + \tau_m(D_{max} - D_{min}) \quad (7)$$

This can be seen in Fig. 5. Here, the planes are redistributed using the cumulative histogram of Fig. 3(b). As can be seen, more planes are available for determining the depth of the objects, and fewer planes are available in empty space. It is desirable to include some planes in the empty spaces between objects to allow the appearance of objects in dynamic scenes. To allow this, all the values in the histogram are increased with a fixed number, based on the number of pixels. This way, the cumulative histogram will be less flat in less interesting regions, allowing some planes here. In our tests, 0.1% of the total amount of pixels demonstrated to be a correct value.



**Fig. 5.** Redistributed depth planes.

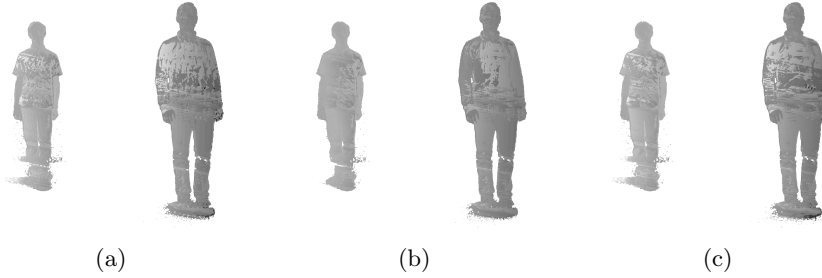
## 4 Results

We tested the proposed method on different scenes and compared image quality and planes required.

The first experiment shows the quality increase when a low number of planes is available. To increase overall quality in both methods, foreground and background segmentation is used. Figure 6(a) shows the result for a uniform depth plane distribution. Artifacts caused by the sparse plane distribution can be clearly seen; the depth map shows clear outliers. The depth map when using a non-uniform plane distribution, based on the histogram of the first depth map, can be seen in Fig. 6(b). Less noise and outliers in the depth values can be perceived. Furthermore, the silhouette is more distinct and the features of the persons are clearer. Using the non-uniform plane distribution increased the quality of the depth map using a low number of planes, thus increasing overall performance.

Figure 6(c) shows the result for a high number of planes. Here, some noise and unclear edges can be perceived. These artifacts are effectively filtered out using the non-uniform plane distribution. The depth planes generating vague edges and noise are not used and can not contribute to the depth map, and thus to the noise and artifacts.

To demonstrate the effect of the cumulative histograms, Fig. 7 and 8 show an input image of a video sequence (a), the corresponding cumulative histogram of the depth map of the preceding frame (b) and the corresponding fraction  $\tau$  from equation 6 (c). When only one dominant depth can be perceived, such as in Fig. 7, one steep section in the cumulative histogram is visible. This part will be transformed to a flat value of  $\tau$ , thus increasing the density of the planes in the corresponding region in the sweeping space. Flat sections of the cumulative

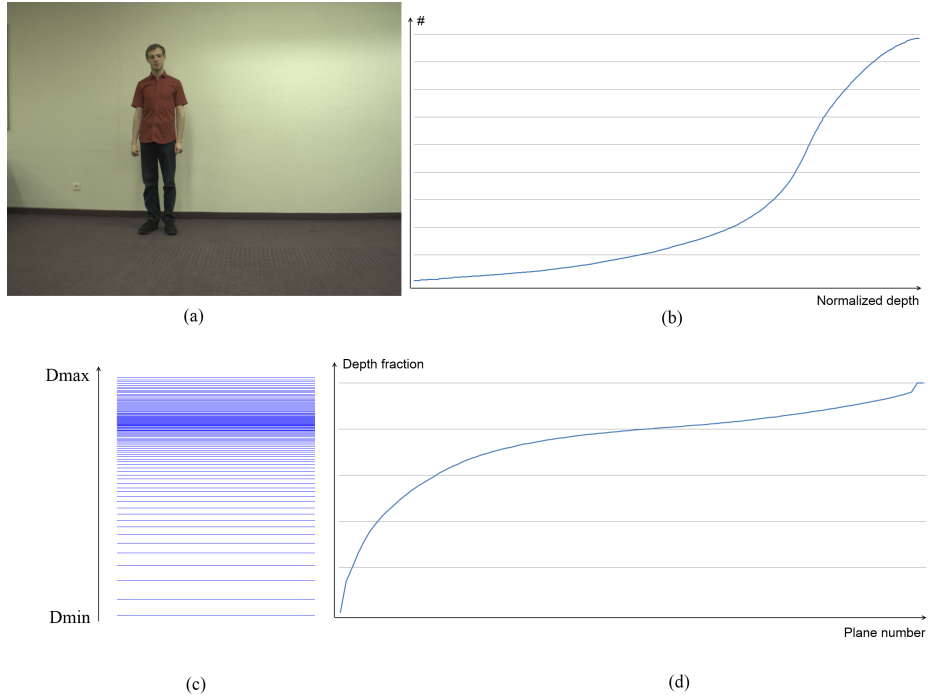


**Fig. 6.** Comparison of the depth map with: (a) a uniform depth plane distribution (only 50 planes used), (b) a non-uniform depth plane distribution (only 50 planes used), and (c) a uniform depth plane distribution (with 256 planes used).

histogram will correspond to steep values in the graph of  $\tau$ , resulting in a sparse plane distribution. When multiple dominant depths are available in the scene, the cumulative histogram will show multiple steep sections (see Fig. 8). This will result in multiple dense regions in the plane distribution, as reflected by the values of  $\tau$  in Fig. 8(c).

The second experiment shows the results for the interpolation for soccer games. The video streams used are from a live game. We placed 8 cameras at one side of the scene and performed real-time interpolation between the cameras. To increase overall quality in both methods, foreground and background segmentation is used. High-quality interpolation for soccer scenes consists of different steps, apart from the plane sweeping, such as debayering and depth filtering. Therefore, the interpolation step should be as fast as possible to reduce execution time and allow real-time processing. By reducing the number of depth planes, the performance can be greatly increased. By incorporating an adaptive plane distribution system, such as described in section 3, fewer depth planes are required while preserving high quality.

The results can be seen in Fig. 11. The quality is increased compared with the uniform plane distribution, as seen in Fig. 10. Details of the quality difference can be seen in Fig. 9. In the uniform plane distribution in 9(a), missing heads and limbs can be perceived, caused by the low number of planes used to determine the interpolated view of the players. By redistributing the depth planes to the position of the players, as can be seen in Fig. 11 and Fig. 9(b), artifacts are seriously reduced. The non-uniform plane distribution method is especially applicable to soccer scenes due to the sparse location of players on the field and the multiple open spaces in the scene. Redistributing the depth planes will thus increase performance by reducing the number of wasted planes. The quality is not reduced by the movement of the scene due to the inclusion of depth planes in empty space. By including a few number of depth planes in empty space, players moving in these spaces are detected and the plane distribution is adapted accordingly.

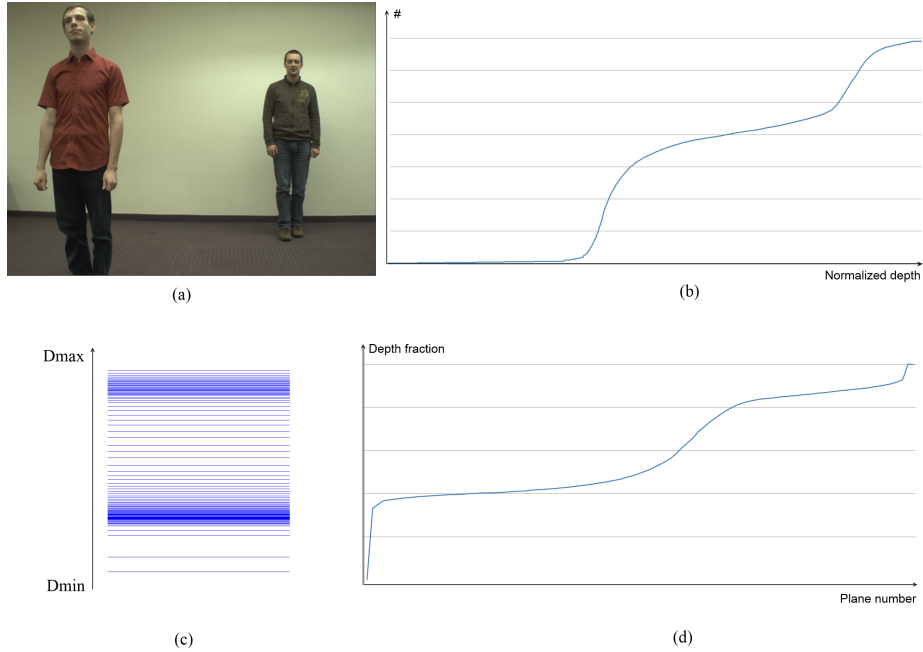


**Fig. 7.** (a) Input image with one person. (b) Cumulative histogram of the depth map. (c) New depth plane distribution. (d) Corresponding fraction  $\tau$  for a given plane number.

To demonstrate the high quality of our results, we increased the number of depth planes to 5000. The quality of the result is high, as shown in Fig. 12, but real-time processing is no longer possible due to the high computational requirements. Comparing Fig. 11 and Fig. 12, we see little difference, proving the effectiveness of our method.

The last experiment demonstrates the results when using a lower quality depth camera to obtain the histogram used for the determination of the plane distribution. We used the Kinect [12] as the depth camera. The Kinect image can be seen in Fig. 13(a), together with the corresponding color image. As can be seen in the depth map, the quality is not sufficient for view interpolation, but can be used for our method. The depth map is relatively complete, so we can use fewer planes than when using the depth from the previous frame. We need fewer planes for empty spaces to compensate for moving objects; the depth map will contain these objects nonetheless. The resulting depth plane distribution and corresponding cumulative histogram can be seen in Fig. 13.

Figure 14(a) shows the result for a uniform depth plane distribution using 10 planes. As can be seen, the depth map contains lots of errors, noise and artifacts; many parts are missing or are incorrect. Figure 14(b) shows the result for a non-



**Fig. 8.** (a) Input image with two persons on different depths. (b) Cumulative histogram of the depth map. (c) New depth plane distribution. (d) Corresponding fraction  $\tau$  for a given plane number.

uniform depth plane distribution using 10 planes, based on the depth map of the Kinect. The depth map is complete, and noise and artifacts are seriously reduced, while maintaining high performance. This demonstrates the usefulness of a lower quality depth camera for our method. However, due to the limited range of the Kinect, only small scale scenes can be used.

## 5 Conclusions

In this paper, we presented a method to reduce computational requirements for view interpolation. When the depth of the scene is not distributed evenly, the plane sweeping method can search in depth ranges where no objects are present, thus reducing computational power and increasing the opportunity for noise. Our method uses the cumulative histogram of the previous temporal frame or from the depth map of a low quality depth camera to determine a more suitable depth plane distribution where the planes are more dense in regions with objects and sparse in regions with no objects. Some planes are assigned to empty spaces to cope with dynamic scenes. All algorithms are implemented using commodity graphics hardware to achieve real-time processing.

We tested the method on different kinds of input sequences, including a scene under controlled conditions and a scene of a live soccer game. Both results proved the effectiveness of the proposed method by providing high quality results with a low number of depth planes, thus reducing computational requirements.

## Acknowledgements

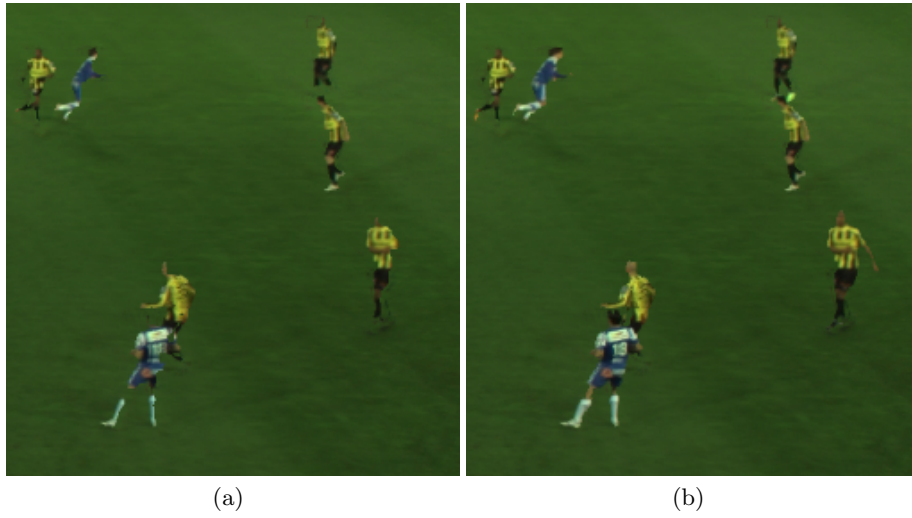
Patrik Goorts would like to thank the IWT for its PhD specialization bursary.

## References

1. Goorts, P., Dumont, M., Rogmans, S., Bekaert, P.: An end-to-end system for free viewpoint video for smooth camera transitions. In: Proceedings of the Second International Conference on 3D Imaging (IC3D 2012), 3D Stereo Media (2012)
2. Goorts, P., Ancuti, C., Dumont, M., Bekaert, P.: Real-time video-based view interpolation of soccer events using depth-selective plane sweeping. In: Proceedings of the Eight International Conference on Computer Vision Theory and Applications (VISAPP 2013), INSTICC (2013)
3. Dumont, M., Rogmans, S., Maesen, S., Bekaert, P.: Optimized two-party video chat with restored eye contact using graphics hardware. *e-Business and Telecommunications* (2009) 358–372
4. Matusik, W., Buehler, C., Raskar, R., Gortler, S., McMillan, L.: Image-based visual hulls. In: Proc. of the 27th annual conference on Computer graphics and interactive techniques, ACM Press/Addison-Wesley Publishing Co. (2000) 369–374
5. Miller, G., Hilton, A., Starck, J.: Interactive free-viewpoint video. In: IEE Eur. Conf. Vis. Media Prod, Citeseer (2005) 50–59
6. Kutulakos, K., Seitz, S.: A theory of shape by space carving. *Intl. Journal of Computer Vision* **38**(3) (2000) 199–218
7. Seitz, S., Dyer, C.: Photorealistic scene reconstruction by voxel coloring. *Intl. Journal of Computer Vision* **35**(2) (1999) 151–173
8. Seitz, S., Curless, B., Diebel, J., Scharstein, D., Szeliski, R.: A comparison and evaluation of multi-view stereo reconstruction algorithms. In: *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*. Volume 1., Ieee (2006) 519–528
9. Zitnick, C., Kang, S., Uyttendaele, M., Winder, S., Szeliski, R.: High-quality video view interpolation using a layered representation. In: *ACM Transactions on Graphics*. Volume 23., ACM (2004) 600–608
10. Yang, R., Pollefeys, M., Yang, H., Welch, G.: A unified approach to real-time, multi-resolution, multi-baseline 2d view synthesis and 3d depth estimation using commodity graphics hardware. *Intl. Journal of Image and Graphics* **4**(4) (2004) 627–651
11. Rogmans, S., Dumont, M., Cuypers, T., Lafruit, G., Bekaert, P.: Complexity reduction of real-time depth scanning on graphics hardware. In: *Proc. of Intl. Conference on Computer Vision Theory and Applications, Lisbon, Portugal (February 2009)*. (2009)
12. Zhang, Z.: Microsoft kinect sensor and its effect. *Multimedia, IEEE* **19**(2) (2012) 4–10



13. Gallup, D., Frahm, J.M., Mordohai, P., Yang, Q., Pollefeys, M.: Real-time plane-sweeping stereo with multiple sweeping directions. In: Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on, IEEE (2007) 1–8
14. Goorts, P., Rogmans, S., Bekaert, P.: Raw camera image demosaicing using finite impulse response filtering on commodity gpu hardware using cuda. In: Proceedings of the Tenth International Conference on Signal Processing and Multimedia Applications (SIGMAP 2012), INSTICC (2012)
15. Yang, R., Welch, G., Bishop, G.: Real-time consensus-based scene reconstruction using commodity graphics hardware. In: Computer Graphics Forum. Volume 22., Wiley Online Library (2003) 207–216
16. Svoboda, T., Martinec, D., Pajdla, T.: A convenient multicamera self-calibration for virtual environments. Presence: Teleoperators & Virtual Environments **14**(4) (2005) 407–422
17. Green, S.: Image processing tricks in opengl. Presentation at GDC **2005** (2005)



**Fig. 9.** Details of the quality differences between (a) figure 10 and (b) figure 11 (our method).



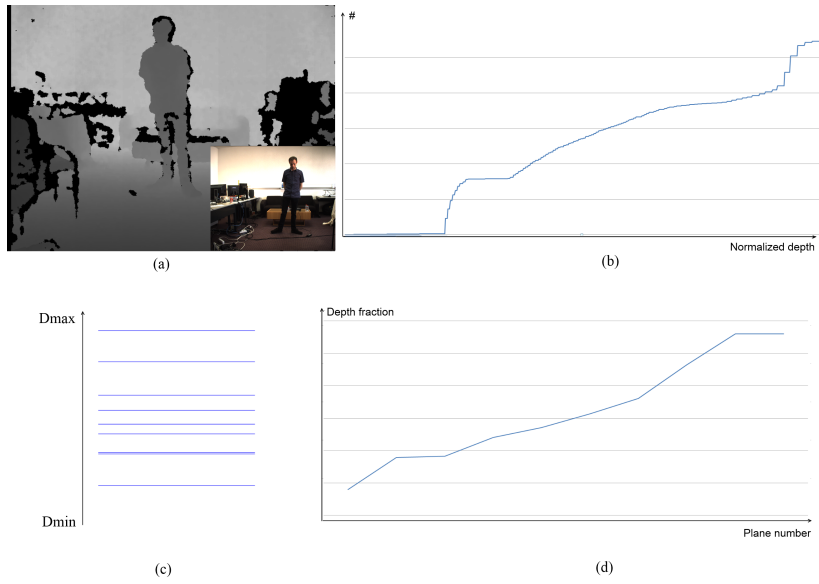
**Fig. 10.** Plane sweeping of a soccer scene with a low number of depth planes (40) and a uniform plane distribution. Many artifacts and missing people can be perceived.



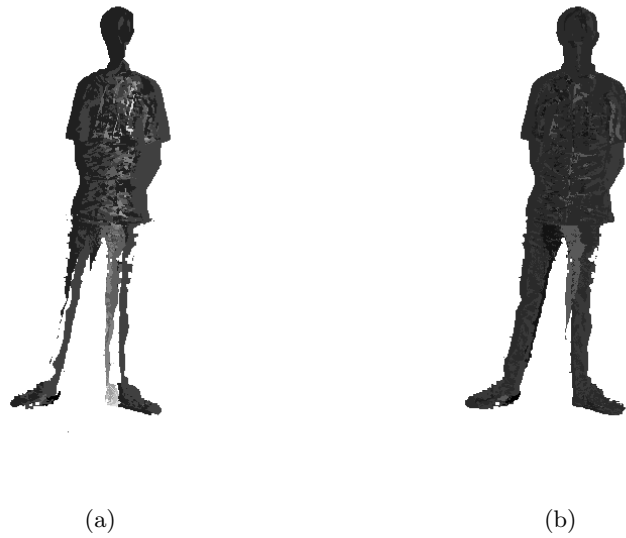
**Fig. 11.** Plane sweeping of a soccer scene with a low number of depth planes (40) and an adaptive plane distribution. The quality is greatly increased in comparison with figure 10.



**Fig. 12.** Plane sweeping of a soccer scene with a high number of depth planes (5000) and a uniform plane distribution. The quality is comparable with Fig. 11, which proves the effectiveness of the method.



**Fig. 13.** (a) Kinect input image with one person, with the color image in the bottomright corner. (b) Cumulative histogram of the depth map. (c) New depth plane distribution for 10 planes. (d) Corresponding fraction  $\tau$  for a given plane number.



**Fig. 14.** Comparison of the depth map with: (a) a uniform depth plane distribution (only 10 planes used), and (b) a non-uniform depth plane distribution using kinect (only 10 planes used).