Acentric Panorama View

Ping-Hsien Lin, Tong-Yee Lee*, Po-Hua Huang

Visual System Lab. (*VSL*), Department of Computer Science and Information Engineering,
National Cheng Kung University, Tainan, Taiwan, R.O.C.
Email: tonylee@mail.ncku.edu.tw

ABSTRACT

In this paper, we propose a novel 2D plenoptic function called “acentric panorama view (APV)”. This novel 2D plenoptic function has the feature that it samples the panorama scenes without oversampling and undersampling like the traditional panorama view. A single APV can be accelerated by view culling and list-priority rendering algorithm. Constructing multiple APVs concentrically, we can have a novel 3D plenoptic function (concentric APVs), which are similar to concentric mosaics [12] and can highly enhance the visibility around the center. The concentric mosaics is indeed a subset of our concentric APVs.

Keywords: image-based rendering, plenoptic function, acentric panorama view (APV), concentric mosaics, concentric APVs, view culling, list-priority rendering algorithm.

1.INTRODUCTION

In traditional rendering approaches, the rendering speed is inversely proportional to the complexity of geometric models within the current field of view (FOV). As the complexity of the geometric models grows, the rendering speed drops. In order to keep a good rendering quality, the speed of traditional rendering approach usually cannot satisfy the demand for interaction in a virtual environment. Additionally, this kind of rendering scheme has the nature: it cannot guarantee a constant frame rate so as to preserve the visual smoothness.

To conquer the drawbacks of traditional rendering approaches, image-based rendering (IBR) arose few years ago. The basic idea of IBR is to generate the novel-view image by several nearby reference-view images. Some researchers present the details of their IBR techniques [2,3,4,6,11]; others [5,7,8,9,10,12,13] further provide the frameworks about how to construct the plenoptic function [1], which describes all the radiant energy perceived from the observer’s viewpoint. As the taxonomy in [12], we classify the plenoptic functions into several types by the dimension of the plenoptic function as listed in Table 1.

The original plenoptic function is a seven dimensions function, three for the viewing space (*x*, *y*, *z*), two for the viewing direction (azimuth angle ¯ and elevation angle ¤. This 2D space is called panorama view or environmental map, which has three basic shapes: sphere, cylinder, and cube), one for wavelengths (ª), and the residual dimension is time (*t*). Eliminating the two dimensions of wavelengths and time, McMillan and Bishop [5] introduced the 5D plenoptic function to computer graphics society for IBR. They gave the definition for IBR as: “given a set of discrete samples (complete or incomplete) from the plenoptic function, the goal of IBR is to generate a continuous representation of that function”. For the general terrain or city walkthrough (not flythrough, this means that the viewing space is a curved surface (2D space)) applications, the plenoptic function reduces to 4D. On the other hand the Lightfield [7] and Lumigraph [8] construct the plenoptic function by restraining the viewing space on a cube’s surface (2D). Therefore, they are also 4D plenoptic functions. Additionally, since their purpose is to explore a centered object, they merely use a general planar square image to represent the 2D space of viewing direction instead of a panorama view. But this approach doesn’t reduce the dimension of the problem. In IBR, there is a fundamental limitation: a reference view should not be used to generate a novel view far away from it for the sake of sampling density. In [12], they propose a 3D plenoptic function called “concentric mosaics”. Concentric mosaic is constructed by taking a series of slit images along a circle. Combining multiple concentric mosaics with different radii and sharing a common center, we can have the “concentric mosaics”. Though a single concentric mosaic is not a perfect plenoptic function (since all slit images is taken along a circle instead of a fixed point), it plays the same role as a plenoptic function at a specific location (i.e., the center). Thus the radius is the 1D viewing space of that plenoptic function. Due to the sampling limitation, the radius should be small and therefore the observer is restricted in a small circle. Finally, if we further restrain the viewing space in a fixed point (0D), the plenoptic function reduce to 2D panorama view [3,9].

Dimension	Viewing space	Name	Year
7	free	plenoptic function	1991
5	free	plenoptic modeling	1995
4	on a cube’s surface	Lightfield, Lumigraph	1996
3	inside a small 2D circle	concentric mosaics	1999
3	inside a small 2D circle	concentric APVs	2000
2	at a fixed point	panorama	1994

Table 1. The taxonomy of plenoptic functions.

In this paper, we present a novel 2D plenoptic function called “acentric panorama view (APV)”. In section 2.1, we

[*] Corresponding author.

depict how the idea of APV emerges. Like traditional panorama view, the APV doesn't suffer the problems of over-sampling and under-sampling, either. Moreover, a single APV can be accelerated by view culling and list-priority rendering algorithm. Thus, we can fast decide the portion of APV needed to be rendered and then render them without *z*-buffer testing. When we construct multiple clockwise and counterclockwise APVs with different radii concentrically, we can have a novel 3D plenoptic function (concentric APVs), which is similar to concentric mosaics [12]. Indeed, the concentric mosaics is a subset of our concentric APVs. This 3D plenoptic function can highly enhance the visibility around the center.

The remainder of this paper is organized as follows. In section 2, we introduce the idea, representation, the sampling of APV, and the concentric APVs. The view culling and list-priority rendering algorithm in a single APV is shown in section 3. Finally, the conclusion and future work are given in section 4.

2.ACENTRIC PANORAMA VIEW (APV)

Sampling analysis of the pin hole camera model is indeed a three-dimensional problem. But for the reason of simplicity and without loss of generality, we only consider the two-dimension case in a plane (2D) walkthrough environment. In this section, we introduce how the acentric panorama view (APV) arises, the representation of APV, and the sampling of APV. In the following, we use the pictorial symbol in Figure 1 to represent a camera (or a depth image). Additionally, when we mention the word: "camera", it also means the depth image associated with that camera.

2.1 Idea of APV

If we arrange reference cameras arbitrarily, the sampling of these reference cameras would tend to over-sampling (oblique zone in Figure 2) or under-sampling (zigzag zone in Figure 2). Over-sampling leads to waste of storage, whereas under-sampling leads to lack of scene information. To avoid over-sampling, we should arrange the reference cameras clockwise or counterclockwise such that the abutting boundaries of FOV of two adjacent cameras are collinear, as shown in Figure 3. But after arranging a circular series of cameras, the over-sampling and under-sampling still arise.

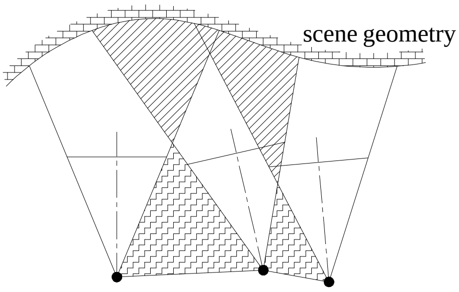


Figure 2. Oversampling (oblique zone) and undersampling (zigzag zone).

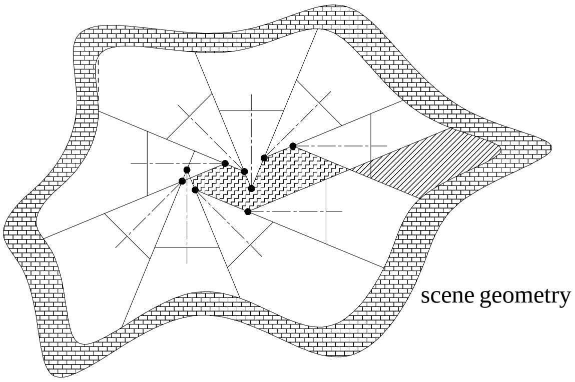


Figure 3. After arranging a circular series of cameras, the over-sampling and under-sampling still arise.

From another point of view, there are two important applications in image-based rendering: *panorama view* and *object view*. We use Figure 4 to demonstrate the relation and difference between these two configurations.

Both panorama view and object view arise based on the concept of a "circle". In object view, all the COPs of cameras are distributed on a circle, and the viewing direction of each camera points to the center of the circle (Figure 4(f)). In panorama view, all the COPs are superposed at the same point, and all the viewing directions point outward from that point (Figure 4(a)). But, if we move each camera along its viewing direction with a fixed distance, the COPs of the cameras are also distributed on a circle (Figure 4(b)). Comparing Figure 4(b) with Figure 4(f), we find that the only difference between these two configurations is the viewing direction of each camera. The former points outward and the latter points inward. Between these two configurations, we can rotate each camera with the same angle from panorama view or object view, as shown in Figure

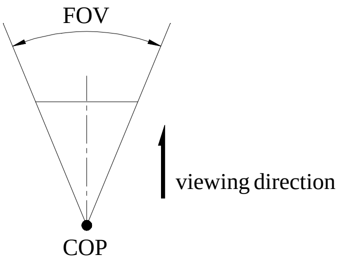


Figure 1. Pictorial symbol of a camera (or a depth image)

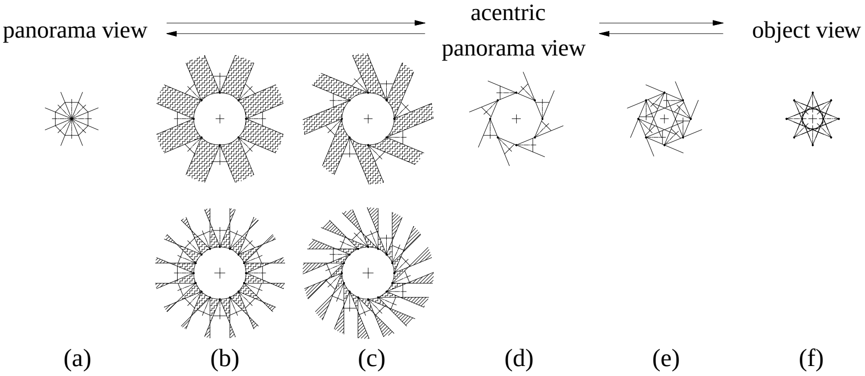
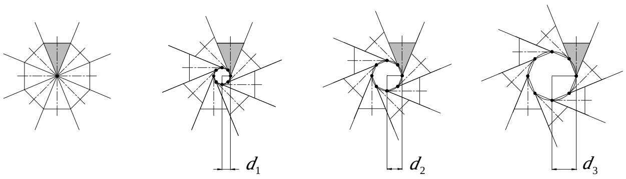


Figure 4. Configurations between panorama view and object view.

4(c), (d), (e).

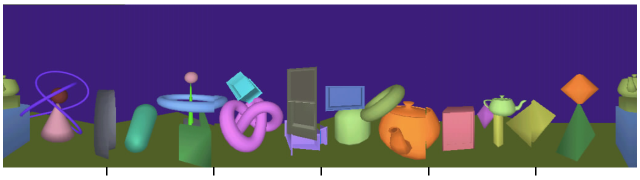
In traditional panorama view, there is no problem of over-sampling and under-sampling. But in above figure of Figure 4(b), the under-sampling arises. If we add additional cameras uniformly to reduce the under-sampling area (as shown in the below of Figure 4(b)), the over-sampling will arise. Here we don't consider the sampling condition inside the circle. This area can be thought as an empty space (i.e., no scene geometry inside this circle). From Figure 4, we can find that only the traditional panorama view (Figure 4(a)) and Figure 4(d) have no over-sampling and under-sampling conditions. The sampling configuration in Figure 4(d) satisfies the statement we describe above (the reference cameras are arranged clockwise or counterclockwise such that the abutting boundaries of FOV of two adjacent cameras are collinear, as shown in Figure 3) and without problem of oversampling and undersampling. We call this sampling configuration (Figure 4(d)) the "Acentric Panorama View (APV)".

The origin of APV can be depicted in another way. Figure 5(a) is the traditional panorama view. We can move each camera's COP perpendicular to its viewing direction with a fixed distance to form an acentric panorama view (Figure 5(b), (c), (d)). Figure 6 and 7 are two examples of counterclockwise APV and clockwise APV respectively. The discontinuity of an APV image is due to the different sampling position.



(a) (b) (c) (d)

Figure 5. Another concept of the origin of APV.



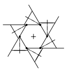
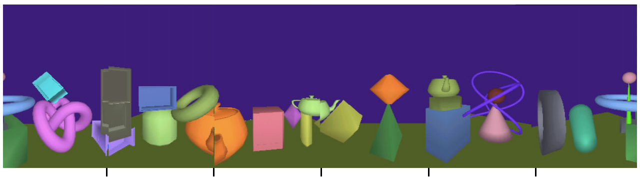


Figure 6. An example of counterclockwise APV.



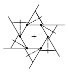


Figure 7. An example of clockwise APV.

2.2 Representation of APV

To express an APV completely, we need four parameters (Figure 8). The first parameter is the center position of the APV: o. The second parameter is the offset vector: d. The offset vector indicates not only the distance of camera's COP from o, but also the starting position of the first camera's COP. The third parameter is the FOV of camera: v (v must divide 360). In other words, the APV is composed of $\frac{360}{v}$ cameras. The last parameter: c indicates whether the configuration is clockwise or counterclockwise. Therefore, an APV is denoted by:

$$APV(o,d,v,c) \tag{1}$$

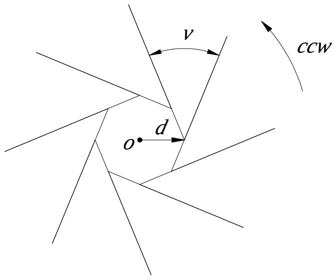


Figure 8. An example of counterclockwise APV.

2.3 Sampling of APV

The original planar walkthrough problem is to construct a complete 4D plenoptic function (i.e., to construct the 2D plenoptic functions (or panorama views) everywhere in the 2D walking space). Since the sampling density and visibility in close-by area are similar, the simplest way to construct this 4D plenoptic function is to resampling the 2D walking area uniformly for the placement of the reference 2D plenoptic functions (or panorama views, as shown in Figure 9), then use these reference 2D plenoptic functions to generate the novel views. This method would worsen the quality of generated images as the resampling scale increases. The image quality is affected by two factors: the sampling density and the visibility. The sparser sampling density blurs the generated images more seriously, and the worse visibility sampling causes more holes and gaps in the generated images.

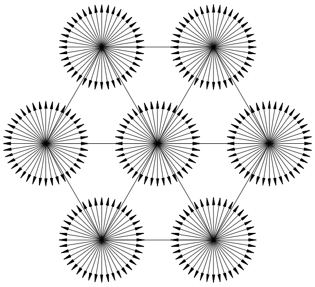


Figure 9. The simplest method to construct the 4D plenoptic function for planar walkthrough.

In [12], they presented a 3D plenoptic function "concentric mosaics" to let users move freely in a circular region. A single concentric mosaic is formed by taking one-column images along a circle in its tangent direction, as shown in Figure 10. Figure 11 shows four counterclockwise concentric mosaics with different radii. Therefore, the concentric mosaic is equivalent to APV with a very small v, such that each image of this APV has just one column. A single traditional panorama view can be used to generate a novel view close to the center. But as the radius of a concentric mosaic (or the offset vector length of an APV) increases, it is less proper to be used to generate novel views. Or we can say it is just more proper to generate the views along the circle in its tangent direction with the radius of concentric mosaic when the radius decreases. Since all image columns of a single concentric mosaic (or an APV) don't share a common COP. Therefore, a single concentric mosaic (or an APV) is not much meaningful. Additionally, the visibility is another issue to tell how the concentric mosaic (or an APV) differs from the traditional panorama view. For example, in the case of Figure 12, the traditional panorama view can see objects: A, B, C, E, and G; and the counterclockwise and clockwise concentric mosaics can see objects: A, B, C, D, F, and H. But the sampled surfaces are different in each case. Therefore, traditional panorama view and a single concentric mosaic (or an APV) can be thought as the sampling collection of panorama scenes with different visibility but without oversampling and undersampling. Intuitively, we can combine multiple clockwise and counterclockwise concentric mosaics with different radii and sharing a common center to form the "concentric mosaics" (or combine multiple clockwise and counterclockwise APVs with different length offset vectors and sharing a common center to form the "concentric APVs", which will be further discussed in section 2.4) for the improvement of visibility.

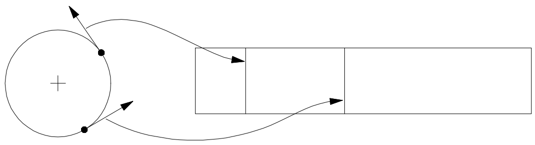


Figure 10. Construction of a counterclockwise concentric mosaic. [12]

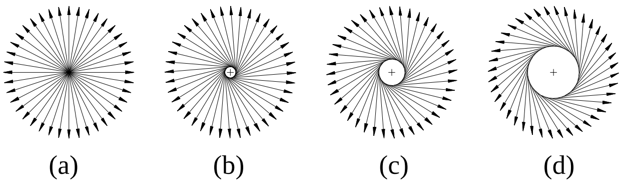


Figure 11. The counterclockwise concentric mosaic with different radii (length of offset vectors). The radius of sub-figure (a) is zero, hence it is equivalent to the traditional panorama view.

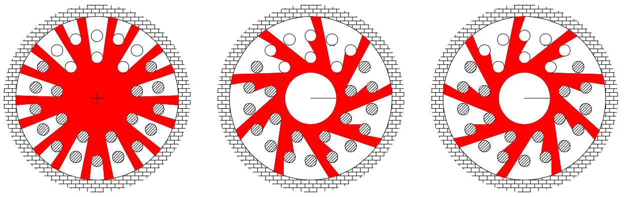


Figure 12. Illustration of visibility in different configurations. The red region is visible to each configuration.

From above discussions, we know the concentric mosaic is a subset of APV. And since the image's FOV of an APV can be 45°, 60°, or some other degrees that can divide 360, we can capture and store the images of an APV more naturally. This characteristic is beneficial especially to the usage of a real camera.

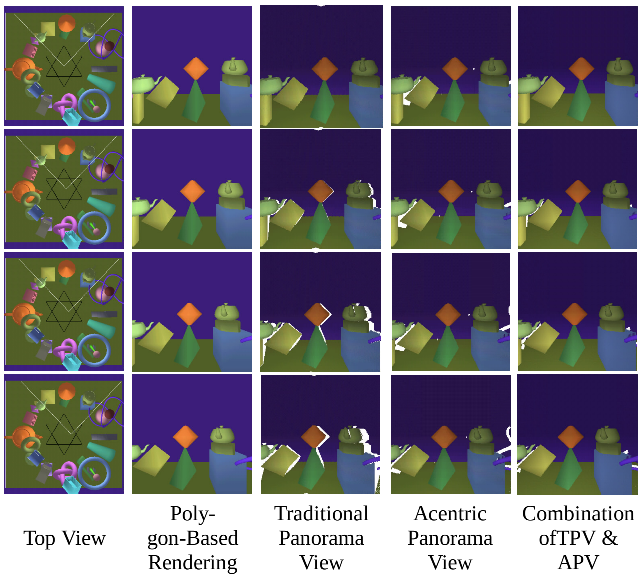


Figure 13. Demonstration of visibility in different framworks. The image sequence in each column is captured from the center to the left gradually as shown in "Top View" column. The star symbol at the center on the ground is the combined configuration of a clockwise APV and a counterclockwise APV.

Figure 13 shows the views generated by polygon-based rendering, traditional panorama view, acentric panorama view, and the combination of traditional and acentric panorama view. The acentric panorama view in this figure is just formed by one clockwise APV and one counterclockwise APV (the star symbol in the "Top View" column in Figure 13). To better understand the sampling condition, we don't use any holes filling technique in Figure 13. Inspecting column 3 (traditional panorama view) and 4 (acentric panorama view), we can know each configuration has its sampling superiority over the other. It is obvious that the quality of the views generated by the combination of traditional and acentric panorama view is better (i.e., with the least number of holes and gaps) than that generated by each alone. But in Figure 13, there are still some holes and gaps in the 5th column. This is because we just use one clockwise APV and one counterclockwise APV. To solve this sampling deficiency problem, we can use the configuration of concentric APVs to improve the sampling.

2.4 Concentric APVs

As mentioned above, constructing multiple clockwise and counterclockwise APVs with different length offset vectors concentrically (Figure 14 is an example) can highly enhance the visibility within the small area around the center. We call this configuration the "concentric APVs". When the FOV of each camera is very small, the concentric APVs reduces to the concentric mosaics [12]. Therefore the concentric mosaics is a subset of our concentric APVs, and the concentric APVs is also a 3D plenoptic function.

When the offset vector increment of a concentric APVs was selected properly, the concentric APVs can sample the scenes very densely and the problems of holes and gaps are insignificant. The redundancy data can be eliminated by constructing all the samples to form the LDIs [11] located at the center of concentric APVs. And the list-priority rendering algorithm [4] can be applied successfully in LDI structure to accelerate the rendering speed.

Figure 14 shows one case out of the infinite configurations of concentric APVs. The configuration in Figure 14 is to align all offset vectors collinear. The above figure of Figure 14 is one panorama view and three counterclockwise APVs with different radii. The below figure of Figure 14 is formed by superposing one panorama view, three counterclockwise APVs, and three clockwise APVs. This figure can give us the visual sense about the sampling distribution of a configuration. We will further discuss the sampling distributions for several different concentric APVs in the future.

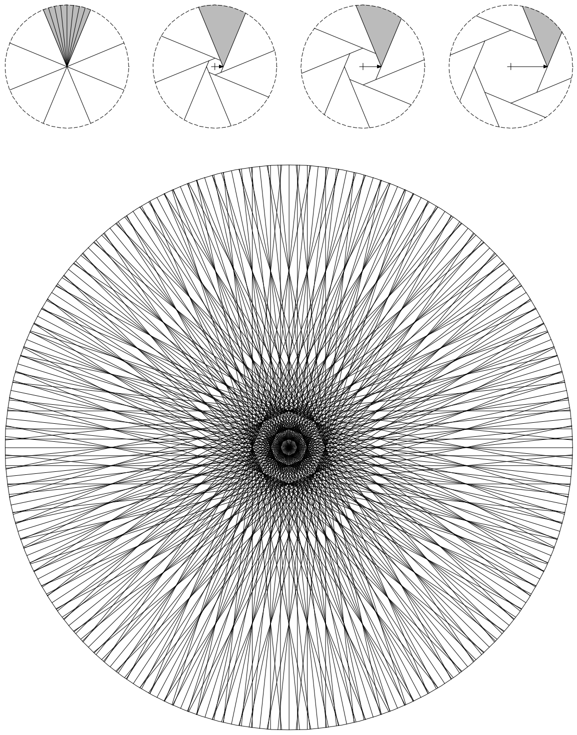


Figure 14. An example of concentric APVs. The offset vectors are aligned collinear)

3.ACCELERATION OF APV RENDERING

Despite of rendering techniques (traditional rendering or image-based rendering), the view culling schemes can accelerate rendering speed dramatically, since it doesn't render the scene primitives we can not see. For example, in Figure 15, we just need to render the depth image A, B, C, D, and E. On the other hand, list-priority algorithms can render the scene primitives with correct visibility but without *z*-buffer testing, therefore can accelerate the rendering speed, too. The list-priority algorithm for a single depth image has been introduced by McMillan [4]. These two rendering acceleration techniques are independent and both can be utilized simultaneously under our APV configuration as described in section 3.1 and 3.2 respectively.

3.1 View Culling in APV

In this section, we show how to perform the view culling inside a counterclockwise APV (i.e., the eye position must be inside the polygon boundaries formed by the camera's COPs of an APV). The view culling in a clockwise APV can be deduced similarly. First, we find the first depth image that can be seen in a counterclockwise order (image *A* in Figure 15). For this purpose, we calculate $r_{eye} \times r_*$ and $r_{eye} \times l_*$ (* denotes one of *A* through *I*) for each camera. There is only one camera (camera *A*) satisfying the rule that $r_{eye} \times r_*$ is negative (inward the paper) and $r_{eye} \times l_*$ is positive (outward the paper). Then, keeping on the counterclockwise order, we find the first pair of two adjacent cameras that locate on each side of the left border of eye (camera *E* and *F*). The former (camera *E*) is in the seen portion and the latter (camera *F*) is in the unseen portion. Thus, in Figure 15, only the counterclockwise series from camera *A* through *E* are within view frustum.

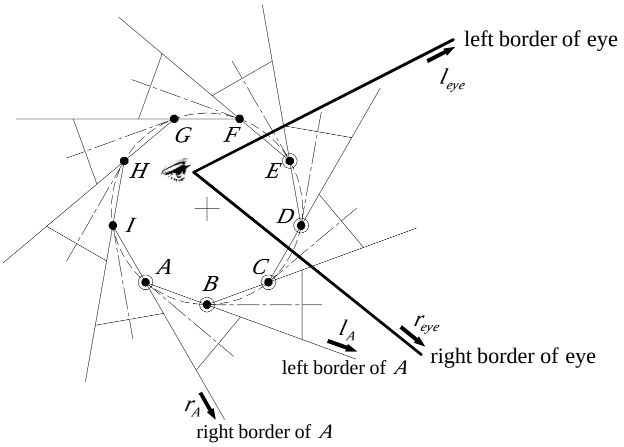


Figure 15. View culling of a counterclockwise APV.

3.2 List-Priority Rendering Algorithm in APV

McMillan's list-priority rendering algorithm [4] comes from the epipolar geometry of two pinhole cameras in computer vision. Figure 16 illustrates the basic idea of this algorithm. If the epipole of reference depth image is negative (for example, the epipole e_{21} in Figure 16 is negative), the rendering order should be from epipole toward image borders for saving *z*-buffer testing; otherwise, the rendering order is from image borders toward epipole.

McMillan's list-priority rendering algorithm cannot be applied successfully to multiple depth images acquired from different positions. Though the depth images are taken from different positions in a single APV, we can apply the list-priority rendering algorithm [4] to each depth image within current FOV in a clockwise (for clockwise APV) or counterclockwise (for counterclockwise APV) order, and still keep the visibility correct. We take the counterclockwise APV for example (Figure 15). It should be emphasized that the viewer must be constrained inside the polygon boundaries formed by the camera's COPs of an APV. Since the APV has no over-sampling, it partitions the space into several disjoint regions. Additionally, in a counterclockwise APV, the right of two adjacent depth images within viewer's FOV is always farther than the left. Therefore, in the case of Figure 15, rendering the depth images in the order: *A-B-C-D-E* (i.e., a counterclockwise order), and applying the list-priority rendering algorithm to each depth image, we can still have correct visibility. Similarly, in a clockwise APV, the visible depth images should be rendered in a clockwise order.

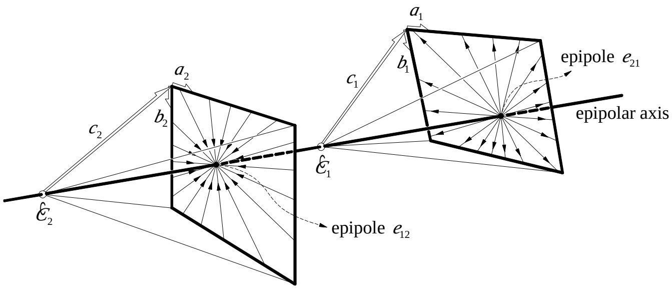


Figure 16. Vectors a_i, b_i, and c_i define an idea pinhole camera i with COP located on $\dot{\mathcal{E}}_i$. The two cameras are reference view and novel view reciprocally. If the camera is the reference one, its image should be warped in the order indicated by the arrows.

4.CONCLUSION AND FUTURE WORKS

In this paper, we propose a novel 2D plenoptic function: acentric panorama view (APV). A single APV has no problem with over-sampling and under-sampling. Its rendering speed can be accelerated by view culling and list-priority rendering algorithm. Additionally, it has the potential to be constructed into a 3D plenoptic function (concentric APVs).

In the future, we will further discuss the sampling distribution of different concentric APVs. Additionally, an especial potential of image-based rendering is that it is easier to achieve the constant frame rate than the polygon-based rendering. However, due to the problems of holes and gaps in image-based rendering, the scenes that is close to the viewer should be rendered using their original geometric models. Therefore, the combination of image-based and polygon-based rendering techniques becomes a tendency in computer graphics. And how to use the structure of APV to partition scene geometries for achieving the constant frame rate is worthy to be further discussed.

To generate a virtual environment from real scenes is another hot topic. Using this technique, we needn't spend the time for construction of scene geometries and the views would look more real. The issue of this technique is how to capture the real scene geometies, including their shape and color. We will discuss whether the APV configuration would benefits this issue in the future.

5.ACKNOWLEDGEMENTS

This project is supported by National Science Council, Republic of China, Taiwan, under contract No. NSC-89-2218-E-006-028.

6.REFERENCES

[1] Adelson, E. H., and J. R. Bergen, "The Plenoptic Function and the Elements of Early Vision," Computational Models of Visual Processing, Chapter 1, Edited by Michael Landy and J. Anthony Movshon. The MIT Press, Cambridge, Mass. 1991.

[2] Chen, S. E., L. Williams, "View Interpolation for Image Synthesis," Proceedings of SIGGRAPH '93, pp. 270-288, 1993.

[3] Chen, S. E., "QuickTime VR - An Image-Based Approach to Virtual Environment Navigation," Proceedings of SIGGRAPH '95, pp. 29-39, 1995.

[4] McMillan, L., " A List-Priority Rendering Algorithm for Redisplaying Projected Surfaces," Technical Report 95-005, University of North Carolina, 1995.

[5] McMillan, L., G. Bishop, "Plenoptic Modeling: An Image-Based rendering System," Proceedings of SIGGRAPH '95, pp. 39-46, 1995.

[6] Steven M. S., Charles R. D., "View Morphing," Proceedings of SIGGRAPH '96, pp. 21-30, 1996.

[7] Marc L., Pat H., "Light Field Rendering," Proceedings of SIGGRAPH '96, pp. 31-42, 1996.

[8] Steven J. G., Radek G., Richard S., Michael F. C., "The Lumigraph," Proceedings of SIGGRAPH '96, pp. 43-54, 1996.

[9] Richard Szeliski and Heung-Yeung Shum, "Creating Full view Panoramic Image Mosaics and Environment Maps", SIGGRAPH '97, p.251-258, August 1997.

[10] P. Rademacher and Bishop G. "Multiple-center-of-projection images", SIGGRAPH '98, p.199-206, July 1998.

[11] Jonathan S., Steven G., Li-Wei He, and Richard S., "Layered Depth Images," Proceedings of SIGGRAPH '98, 231-242, 1998.

[12] Heung-Yeung Shum, Li-Wei He, "Rendering with Concentric Mosaics," Proceedings of SIGGRAPH'99, pp. 299-306, 1999.

[13] Oliveira, Manuel M. and Gary Bishop. Image-Based Objects. Proceedings of 1999 ACM Symposium on Interactive 3D Graphics, (Atlanta, Georgia), April 26-28, 1999, pp. 191-198.