Translating the Viewing Position in Single Equirectangular Panoramic Images

Frode Eika Sandnes[1,2] and Yo-Ping Huang[3]
[1]Department of Computer Science, Faculty of Technology Art and Design, Oslo and Akershus University College of Applied Sciences, Oslo, Norway
[2]Faculty of Technology, Westerdals School of Art, Communication and Technology, Oslo, Norway
[3]Department of Electrical Engineering, National Taipei University of Technology, Taipei, Taiwan 10608
email: Frode-Eika.Sandnes@hioa.no, yphuang@ntut.edu.tw

Abstract—**Equirectangular panoramas are popular tools for achieving 360 degree immersed viewing experiences. A panorama captures a scene from one point and panoramic viewers allow the user to control the viewing direction, but the viewer is not allowed to move around. This study proposes a strategy for transforming equirectangular panoramic images with the effect of moving freely in three dimensions. The strategy assumes that the panorama is an enclosed space comprising a flat ground and flat vertical walls. A equirectuangular Hough transform is proposed for detecting the boundaries of the respective planes. The panoramic image is then decomposed into the respective planes, the viewing point is translated and a new panoramic image based on the new viewing position is composed. Preliminary proof of concept test shows that the strategy allows free translation within simple panoramic images.**

Keywords-equirectangular panorama, Hough transform, translation, edge detection, immersed interaction

I. INTRODUCTION

Equirectangular panoramas have become popular especially with services such as Google street view [1], but are also used in other domains such as museums [2]. Equirectangular panoramas are single images that capture a scene viewed from one point going in all directions. Full equirectangular panoramic images are twice as wide as they are high as they represent 360 degrees horizontally and 180 degrees vertically. Panoramic images are viewed using standard interactive panoramic viewing software [3, 4]. The user can usually control the viewing direction and field of view giving the user a convincing impression of being present in the actual location.

Equirectangular panoramas are captured either by special cameras lenses or multiple images with the camera pointing in the different direction. Special software is used to transform single images taken by a special lens, or merge multiple images, into the equirectangular panoramic image [5, 6].

Although panoramic viewers gives the observer a realistic immersed third-person experience, they do not allow the observer to move around. For example, Google street view allows the viewer to move around, but the viewer is moving from one panorama to another where the panoramas are captured at regularly spaced positions along the streets. The transitions between the different panoramas are discrete and sudden.

The objective of this work is to develop a method that allows a viewer to move around certain panoramas. This work assumes that the panorama contains man-made objects, which often contain flat faces. Example scenarios include moving around inside a room defined by one panoramic image, or moving from one panoramic image to another in other to create more smooth transitions between different panoramic images. In practice, the planes would usually represent walls, being it inside a building or outside between buildings.

The method uses a Hough transform to attempt identifying flat vertical faces, or walls, in the panorama. Using the face information the panoramic image is decomposed into the respective faces, the viewing position is moved and finally a new panoramic image is reconstructed using the flat faces according to the new viewing position.

II. BACKGROUND

Several approaches with similar goals have been attempted in the literature. Common to these methods is that they rely on two or more panoramic images. Shi et al. [11, 12] proposed a method for changing the viewer's position in a location by interpolating between different cubic panoramic views. Cubic panoramas comprise a set of six square images representing the panorama in all six direction like a cube or box. Shi et al.'s interpolation approach is based on ray-tracing techniques, which checks the color consistency of corresponding pixels in different panoramas.

Shanat and Laganiere [13] solved the same problem of interpolating between different panoramic views, but did it in real time with a graphical processing unit. Their approach is based on computing the optical flow field between adjacent cubic panoramas, and then morphing is used to achieve the motion between the panoramas. Both color features and geometric features were incorporated into the interpolation.

Zhang et al.'s [14, 15, 16] approach to interpolating between different panoramas is based on computing a complex mesh of triangles, when are subsequently used to resynthesize panoramic

978-1-5090-1897-0/16/$31.00 ©2016 IEEE

views from different positions. A similar triangularisation approach was proposed by Zhao et al. [17] who also prioritized achieving pleasant visual transitions over realistic representations.

The more general problem of detecting planes in three-dimensional space has received much research attention [18, 19, 20, 21], which of most are based on stereographic images. Some attempts have also been based on incomplete 3D-data [22] and hand-held camera swipes [23]. Attempts have also been based on single images where assumption is made about the scene, such as wall detection [24].

The approach presented herein distinguish itself from previous attempts on panoramic images in that it does not require multiple panoramic images. However, unlike previous methods it does make assumptions about the scene geometry being rich in planes.

III. METHOD

The proposed method involves transforming an equirectangular panorama as viewed from a translated position. The method involves three steps. First, the enclosing planes, or walls, are detected. Next, these planes are extracted from the panorama, the viewing position is translated and finally the new panorama is constructed from the new viewing position.

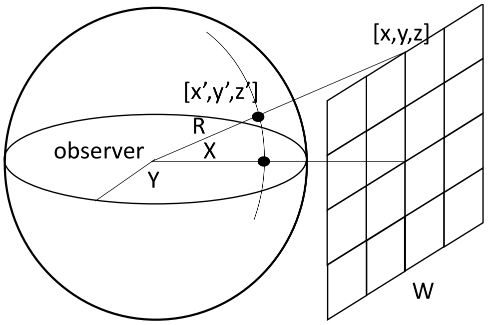

Figure 1. Projecting image planes onto the viewer's sphere..

A. Plane to sphere projection

The method proposed herein is making extensive use of plane to sphere projections. The observer is defined as being in the origin of a three dimensional Cartesian coordinate system, surrounded by a sphere *S* with radius *R*, here arbitrarily set to 1, located at [0, 0, 0]. The observer observes a plane perpendicular to the viewer, that is, a plane with its normal pointing towards the center of the sphere (see Fig. 1). A rectangular image is defined on the plane. The task is to project this image onto the surface of the sphere defined using a geographical coordinate system of latitude and longitude [7, 8].

The face image is $pixels_x$ pixels wide and $pixels_y$ pixel high. The center of the image is defined as the closest point to the center of the sphere, that is, the normal. The size of the image (*width*, *height*) in the Cartesian space is defined by the horizontal and angles a, b between the normal vector and the vector from the sides of the image (see Fig. 1). Mathematically, the pixel points for an image lying on the plane z=1 are defined as

$$x_{i,j} = i\frac{width}{pixels_x} - \frac{width}{2}, i \in [0..pixels_x] \tag{1}$$

$$y_{i,j} = j\frac{height}{pixels_y} - \frac{height}{2}, j \in [0..pixels_y] \tag{2}$$

$$z_{i.j} = 1 \tag{3}$$

where the image size is given by

$$width = 2 \times R \tan a \tag{4}$$
$$height = 2 \times R \tan b \tag{5}$$

If the image is quadratic if *width* = *height*.

Next, the point of intersection [x', y', z'] between the viewing sphere *S* and the line going from the center of the sphere to an image pixel point [x, y, z] is simply:

$$[x', y', z'] = \frac{R}{\sqrt{x^2+y^2+z^2}}[x, y, z] \tag{6}$$

The point of intersection is thus defined by the vector with length *R* along the line going from the sphere origin to the grid point.

Finally, the intersection point [x', y', z'] is transformed into geographical spherical coordinates [φ, ϕ] analogous to latitude and longitude, using the following expression

$$\varphi = \tan^{-1}(y', x') \tag{7}$$

$$\phi = \sin^{-1}\left(\frac{z'}{R}\right) \tag{8}$$

Note that it is the arctan2 function that is used which yields an angle in the range -180 to 180 degrees.

B. Plane boundary detection

The proposed method assumes that a panoramic view representing surrounding planes. For example, the viewer may be inside a box, in a city with buildings, etc. For simplicity, it is assumed that the viewer is fully immersed inside six enclosing planes perpendicular to the viewer and to each other respectively, making up the walls, floor and ceiling.

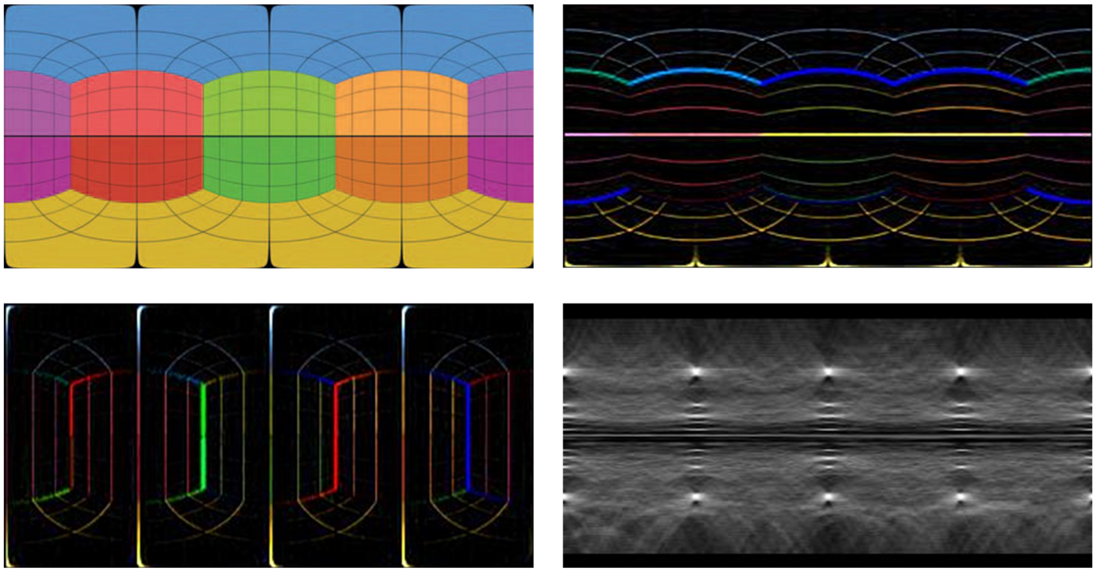

Figure 2. A equirectangular panorama of a cube, panorama subject to horizontal edge detection and vertical edge detection and the accumulator array for the Hough transform of the horizontal edges.

It is assumed that the floor is defined by the plane below defined by the normal vector [0, 1, 0] and a ceiling above the viewer defined by [0, -1, 0].

The first task is thus to detect the boundaries of the six planes. This is achieved in three steps. First, vertical and horizontal edge detection is applied to the image using a vertical and horizontal Sobel convolution filter V and H, respectively, with the following kernels:

$$V = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}, \quad H \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \tag{9}$$

One property of the equirectangular panoramic space is that vertical lines in the original scene also remain vertical in the equirectangular panoramic space. Therefore, the vertical edges separating the walls are found by taking the vertical projection [9] of the vertical edge detected image, or more specifically

$$p(x) = \sum_{y} B(x, y) \tag{10}$$

where $p(x)$ is the projection at x and $B(x,y)$ is the binary value of at x, y, defined by

$$B(x, y) = \begin{cases} 1, \ I(x, y) = foreground \\ 0, \ I(x, y) = backgroun \end{cases} \tag{11}$$

Here, $I(x,y)$ is the pixel value at x, y.

The vertical projection reveals the horizontal position of the vertical edges as peaks in $p(x)$.

Unlike vertical lines, horizontal lines do not remain straight nor horizontal in the equirectangular plane. Instead, straight lines become curved in the panoramic domain. Therefore, an equirectangular Hough transform [10] is proposed for detecting horizontal lines, and hence detecting the horizontal boundaries of the wall-planes, that is, the top and bottom of these planes.

The new Hough transform builds an accumulator table $H(\varphi,\phi)$ comprising a count of the number of pixel values that are set on the projected horizontal line that intersects the viewing vector $[\varphi,\phi]$. Here φ is the latitude of the viewing orientation using geographical coordinate system in the range -180 to 180, and ϕ is the longitude in the range -90 to 90. The accumulator array is constructed by generating all the straight lines in the Cartesian space that are horizontal with z=0 and that is normal to the viewing vector. The projection of each of these straight lines onto the equirectangular panoramic image are traced and all non background pixels are counted. Or more concisely

$$H(\theta, \varphi) = \sum_{[x,y] \in T(\theta, \varphi)} B(x, y) \tag{12}$$

where $T(\varphi, \phi)$ is the line perpendicular to the viewing vector projected onto the equirectangular space. The maximum values of the accumulator table reveals the $[\varphi, \phi]$ position of vertical lines in the equirectangular space, that is, φ reveals the plane orientation and ϕ reveals where the plane begins or ends, where a low ϕ is the bottom of a plane and ϕ is the top of a plane.

C. Plane extraction

Once the plane boundaries are extracted the rectilinear image is projected back onto the respective plane within the defined boundaries making up the face. The proof-of-concept implementation used in this study achieves this by traversing pixel points on the plane and filling these with the corresponding value of the corresponding point projected onto the equirectangular space.

More concisely, a point $[x, y]$ in the image space $Q(x, y)$ is converted to Cartesian space $[x_p, y_p, z_p]$. The Cartesian coordinate is projected onto a point [φ, ϕ] on the viewer sphere, and the pixel value $I(\varphi, \phi)$ at this point is used to the original pixel in the image face $Q(x, y)$.

D. Viewpoint translation

Translation is specified using a vector [o_x, o_y, o_z], which represents the relative move of the viewer. To simplify computation the origin of the Cartesian center is kept as the viewing position. Therefore, the planes are instead projected by [-o_x, -o_y, -o_z] giving

$$x = n_x + p_x - o_x \tag{13}$$

$$y = n_y + p_y - o_z \tag{14}$$

$$z = n_z + p_z - o_z \tag{15}$$

Where [n_x, n_y, n_z] is the plane normal vector and [p_x, p_y, p_z] is a point on the plane.

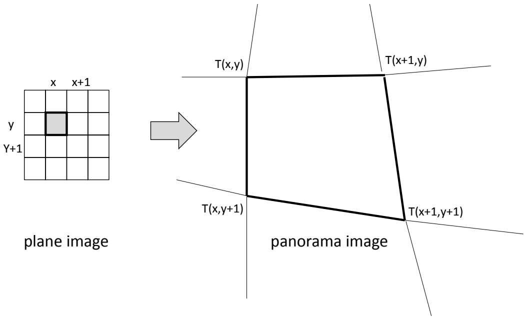

Figure 3. Image on the plane to panorama mapping.

E. Panorama construction

Having obtained the respective faces of the panorama and given a desired translation the new equirectangular image is constructed as follows. The points of each phase image I is traversed, and for each point [x, y] the corresponding projected point [φ, ϕ] given by the tranformation $T(x, y)$ on the viewing sphere is computed. Since the relationship between pixels in the plane and the equirectangular space is non-linear the blank space between some points are filled in using polygons (see Fig. 3). More precisely pixel value $I(x, y)$ of the face is used to color a polygon defined by the four points $T(x, y)$ -> $T(x+1, y)$ -> $T(x+1, y+1)$ -> $T(x, y+1)$.

IV. EXPERIMENTS

A simple proof-of-concept implementation was realised in java. The resulting transformed panoramas produced with the implementation are shown in Fig. 4 left and an example corresponding rendered view on the right. The renderings were created using the FSPViewer panoramic viewer with a 138 degrees horizontal field of view, a yaw of 180 degrees and pitch of 0 degrees. The viewing parameters are kept constant such that the effects of only the translation parameters are visualized. The top panorama is the original without transformation, the second is lowered with an offset of [0, 0, -0.4], the third panorama is viewed with an offset of [0, 0, -0.8] which is close to the floor. The last two panoramas alters the x and y direction with offsets of [-0.4, -0.4, -0.8] and [-0.8, -0.8, -0.8], respectively. The cube itself has dimensions of 2×2×2 units.

The illustrations show that the projections after the translation transformation is nearly visually perfect. However, a low-resolution image was used and the aliasing effects are quite noticeable. No particular effort is made to perform anti-aliasing apart from filling in uncovered pixels using polygons. Moreover, this simple example is a very easy case as it has regular faces that are easy to identify.

V. CONCLUSIONS

A technique for translating the viewer's position in single equirectangular panoramas was presented. The translation is achieved by transforming the panoramic image in three steps. First, the main flat vertical faces of the scene is identified, then the panorama is decomposed into its respective faces, and finally the faces are used to reconstruct the panorama from the new viewing position. The approach is limited to panoramas with flat faces or approximately flat faces and does not work for general panoramas. The method is therefore applicable within buildings and between buildings and other structures with flat faces. Future work includes improving the face detection algorithm and incorporate techniques reported in augmented reality research [24] to make the method more robust and capable of detecting general surfaces. Another direction is to explore the proposed technique for moving around hand drawn panoramic sketches [25].

REFERENCES

[1] D. Anguelov, C. Dulong, D. Filip, C. Frueh, S. Lafon, R. Lyon, A. Ogale, L. Vincent and J. Weaver. "Google street view: Capturing the world at street level." Computer, Vol. 6, 32-38, 2010.

[2] K. Kwiatek, and M. Woolner. "Transporting the viewer into a 360 heritage story: Panoramic interactive narrative presented on a wrap-around screen." In The 16th International Conference on Virtual Systems and Multimedia (VSMM), IEEE, pp. 234-241, 2010.

[3] F. Senore. FSPViewer, http://www.fsoft.it/FSPViewer/, downloaded 20. November, 2015

[4] T. P. Keane, N. D. Cahill, H. Rhody, B. Hu, J. Tarduno, R. Jacobs and J. Pelz. "Sphere 2: Jerry's rig, an OpenGL application for non-linear panorama viewing and interaction." In Image Processing Workshop (WNYIPW), IEEE, pp. 13-16, 2012.

[5] Y. Xiong and K. Turkowski. "Registration, calibration and blending in creating high quality panoramas." In Proceedings of The Fourth IEEE Workshop on Applications of Computer Vision, WACV'98, IEEE, pp. 69-74. IEEE, 1998.

[6] G. Kweon, and Y. Choi. "Image-processing based panoramic camera employing single fisheye lens." Journal of the Optical Society of Korea, Vol. 14, No. 3: 245-259, 2010.

[7] F. E. Sandnes, "Where was that photo taken? Deriving geographical information from image collections based on temporal exposure attributes." Multimedia Systems Vol. 16, No. 4-5: 309-318, 2010.

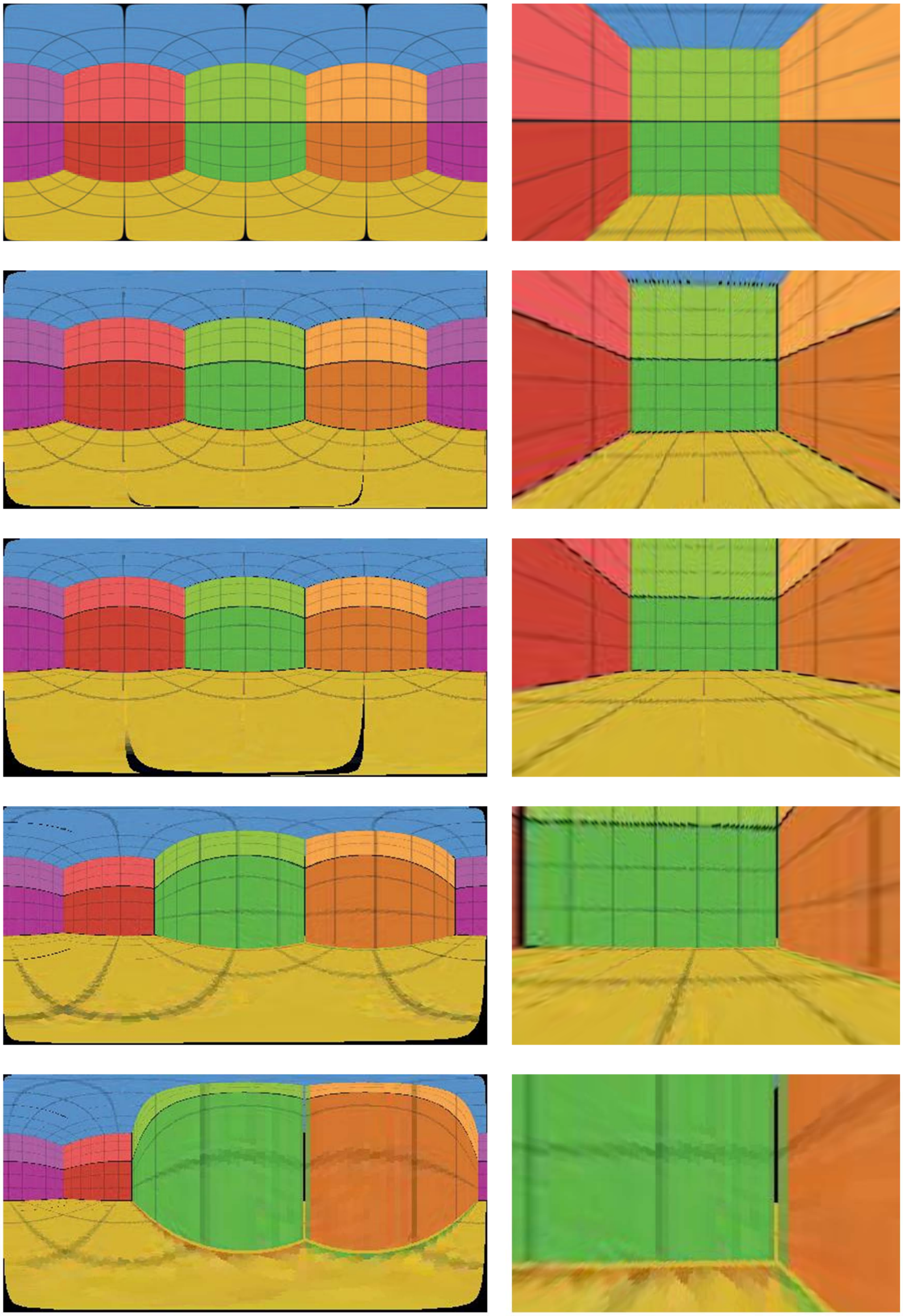

Figure 4. Example translations using a single panoramic image (top) left. The images on the right shows example renderings.

[8] F. E. Sandnes, "Determining the geographical location of image scenes based on object shadow lengths." Journal of Signal Processing Systems Vol. 65, No. 1, 35-47, 2011.

[9] Y.-P. Huang, T.-W. Chang, Y.-R. Chen and F. E. Sandnes. "A back propagation based real-time license plate recognition system." International Journal of Pattern Recognition and Artificial Intelligence Vol. 22, No. 2, 233-251, 2008.

[10] D. H. Ballard, "Generalizing the Hough transform to detect arbitrary shapes." Pattern recognition Vol. 13, No. 2, 111-122, 1981.

[11] F. Shi, R. Laganiere, E. Dubois and F. Labrosse. "On the use of ray-tracing for viewpoint interpolation in panoramic imagery." In Canadian Conference on Computer and Robot Vision, 2009. CRV'09, IEEE, pp. 200-207, 2009.

[12] S. Kolhatkar and R. Laganiere. "Interactive scene navigation using the gpu." Computer graphics international, Singapore, 2010.

[13] S. Kolhatkar and R. Laganiere. "Real-time virtual viewpoint generation on the GPU for scene navigation." In 2010 Canadian Conference on Computer and Robot Vision (CRV), IEEE, pp. 55-62, 2010.

[14] C. Zhang, E. Dubois and Y. Zhao. "Intermediate cubic-panorama synthesis based on triangular re-projection." In 17th IEEE International Conference on Image Processing (ICIP), 2010, IEEE, pp. 3985-3988, 2010.

[15] C. Zhang, Yan Zhao, and F. Wu. "Triangulation of cubic panorama for view synthesis." Applied Optics, Vol. 50, No. 22, 4286-4294, 2011.

[16] C. Zhang, E. Dubois, and Y. Zhao. "Virtual cubic panorama synthesis based on triangular reprojection." Computer Animation and Virtual Worlds, Vol. 25, No. 2, 143-154, 2014.

[17] Q. Zhao, L. Wan, W. Feng, J. Zhang, and T.-T. Wong. "Cube2Video: Navigate between cubic panoramas in real-time", IEEE Transactions on Multimedia, Vol. 15, No. 8, 1745-1754, 2013.

[18] M. I. A. Lourakis, A. A. Argyros and S. C. Orphanoudakis. "Detecting Planes In An Uncalibrated Image Pair." In BMVC, pp. 1-10. 2002.

[19] M. I. A. Lourakis, S. V. Tzurbakis, A. Argyros, and S. C. Orphanoudakis. "Feature transfer and matching in disparate stereo views through the use of plane homographies", IEEE Transactions on Pattern Analysis and Machine Intelligence Vol. 25, No. 2, 271-276, 2003.

[20] J. Piazzi, and D. Prattichizzo. "Plane detection with stereo images." In Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006, IEEE, pp. 922-927, 2006.

[21] J. Prankl, M. Zillich, B. Leibe and M. Vincze. "Incremental Model Selection for Detection and Tracking of Planar Surfaces." In BMVC, pp. 1-12. 2010.

[22] M. Heracles, B. Bolder and C. Goerick. "Fast detection of arbitrary planar surfaces from unreliable 3D data." In IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009. IROS 2009. IEEE, pp. 5717-5724, 2009.

[23] D. Chekhlov, A. P. Gee, A. Calway and W. Mayol-Cuevas. "Ninja on a plane: Automatic discovery of physical planes for augmented reality using visual slam." In Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, IEEE, pp. 1-4, 2007.

[24] G. Simon, "Automatic online walls detection for immediate use in AR tasks." In IEEE/ACM International Symposium on Mixed and Augmented Reality ISMAR 2006, IEEE, pp. 39-42, 2006.

[25] F. E. Sandnes, "PanoramaGrid: A Graph Paper Tracing Framework for Sketching 360-degree Immersed Experiences." In Proceedings of the International Working Conference on Advanced Visual Interfaces, ACM, pp. 342-343, 2016.