Generating fish eye panoramas from any arbitrary direction of view

Mayank Bhagya
MDN – Intern
mayankbhagya@gmail.com

Vinay Kumar
MDN
vinay.kumar@manipal.net

Abstract- **This paper discusses a method for stitching multiple fisheye images in order to generate a 360 degree spherical image and panoramic views corresponding to any arbitrary viewing direction.**

Index Terms**—Computer graphics, image registration, smart cameras, surveillance**

I. INTRODUCTION

Fisheye cameras are finding increasing use in surveillance and automotive imaging applications due to their ultra-wide-angle properties and cost-effectiveness. Their large wide-angle view makes it possible to use fewer cameras to generate panoramic views. However, fisheye images suffer from severe distortion due to the frontal hemispheric scene being mapped onto a flat surface which makes the stitching of multiple fisheye images a non-trivial task (Fig. 1).

Fig. 1. A fisheye image.

In our previous work ([1], [2]), we have discussed the correction of fisheye images and generating panoramic views by projecting them onto a cylindrical compositing surface. However, such panoramas are not completely free from distortion, which becomes apparent if we want the panoramic view to include areas near the zenith or the nadir of the camera.

In this paper, we present an algorithm for stitching multiple fisheye images in order to first generate a spherical image i.e. complete 360 degree global equidirectional view of the entire scene. The global equidirectional view can then be mapped onto perspective and panoramic views for any arbitrary viewing direction. In Section II, we discuss the mapping that allows us to map a fisheye image onto its equidirectional view, and the stitching of equidirectional views of multiple fisheye images to form a global equidirectional view of the scene. In Section III we describe the mapping of the global equidirectional view to a perspective view for any arbitrary viewing direction. In Section IV, we present the results of our technique using three closely placed fisheye cameras.

II. GENERATING A GLOBAL EQUIDIRECTIONAL VIEW

A fisheye camera can be modeled as a pinhole camera, where the incident light rays are projected onto a hemisphere whose radius is same as the focal length of the fisheye lens. (Fig. 2). Thus a fisheye image can be treated as an image formed on this sphere.

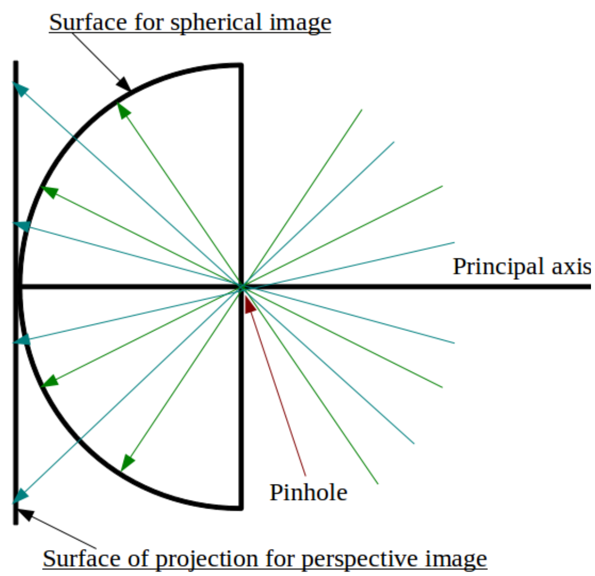


Fig. 2. Pinhole model of a fisheye camera.

Theoretically, if two fisheye cameras have coinciding pinholes, both the images will be formed on a single common sphere. In practice however, fisheye lenses kept relatively close to each other, can be assumed to have the same pinhole, and hence the images captured from such closely placed lenses can be treated as images being formed on a single sphere. Fig. 3 shows how the image is formed on two fish eye cameras with coincident pinholes.

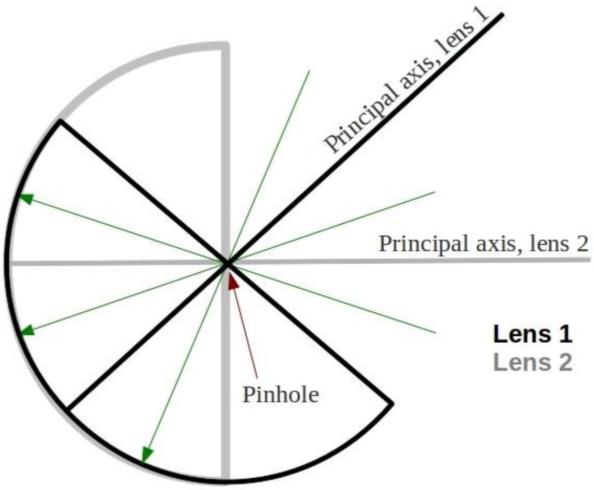


Fig. 3. Pinhole model of two fisheye cameras with coinciding pinholes.

A. *The θ - φ and α - β spaces*

The Cartesian coordinates of point P corresponding to Spherical coordinates (θ - φ) can be expressed as (Fig. 4):

$$\begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix} = f \cdot \begin{bmatrix} \sin(\theta)\cdot\cos(\phi) \\ \sin(\theta)\cdot\sin(\phi) \\ \cos(\theta) \end{bmatrix} \quad \text{- (1)}$$

where *f* is the focal length.

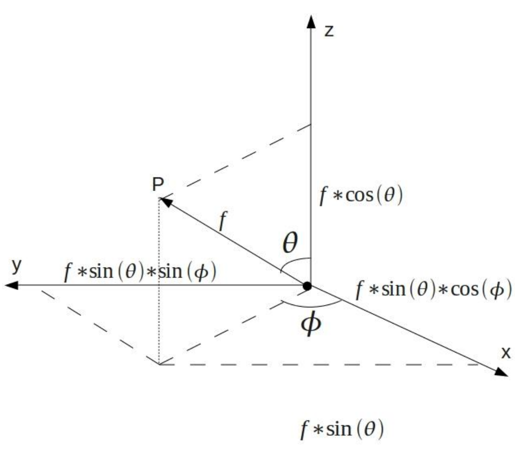
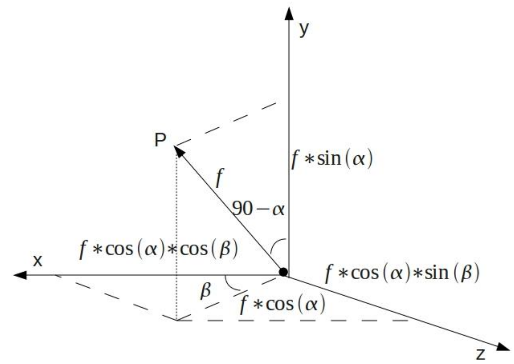


Fig. 4. A point P in θ - φ space.

It is also useful to express each point of the sphere in terms of latitudes and meridians. The Cartesian coordinates of point P in the latitude(α) - meridian (β) space can be written as (Fig. 5):

$$\begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix} = f \cdot \begin{bmatrix} \cos(\alpha)\cdot\cos(\beta) \\ \sin(\alpha) \\ \cos(\alpha)\cdot\sin(\beta) \end{bmatrix} \quad \text{- (2)}$$

where *f* is the focal length.



Inter-conversion between the above two coordinate systems is easy to derive and comes out to be:

$$\alpha = \arcsin(\sin(\theta)\cdot\sin(\phi)) \qquad \beta = \arcsin\left(\frac{\cos(\theta)}{\sqrt{(1-\sin^2(\theta)\cdot\sin^2(\phi))}}\right) \quad \text{- (3)}$$

$$\theta = \arccos(\cos(\alpha)\cdot\sin(\beta)) \qquad \phi = \arcsin\left(\frac{\sin(\alpha)}{\sqrt{(1-\cos^2(\alpha)\cdot\sin^2(\beta))}}\right) \quad \text{- (4)}$$

B. *Equidirectional view of a fisheye image*

An equidirectional projection maps meridians to equally spaced vertical straight lines, and circles of latitude to evenly

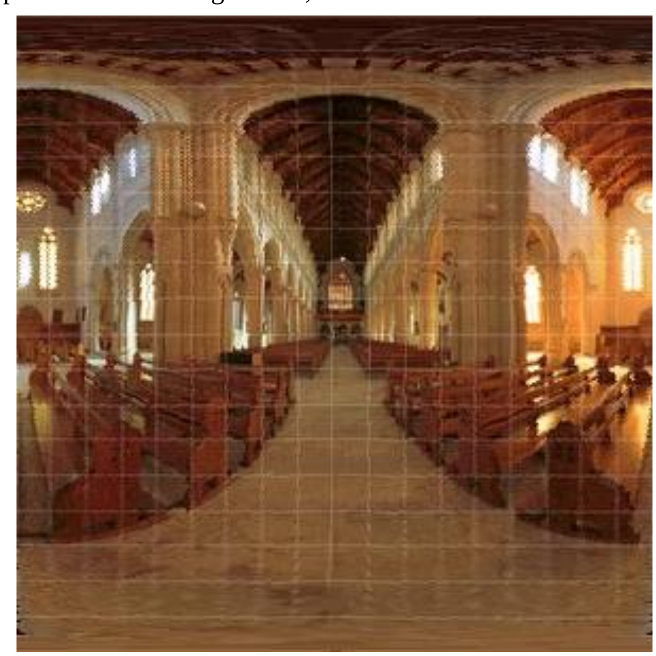


Fig. 6. Equidirectional projection of the fisheye image shown in Fig. 1.

spread horizontal straight lines (Fig. 6). As a representation of the entire spherical view on a flat plane, it has the characteristic of confining the distortion exclusively to the vertical direction only, while keeping the horizontal direction distortion-free. This allows us to easily and reliably stitch multiple equidirectional views from different cameras as long as their views can be mapped onto the same equidirectional view (i.e. the same α - β space).

In order to generate an equidirectional view from an input fisheye image we first note that the latitudes and meridians of an equidirectional view lie in the range [$-\pi/2$, $\pi/2$] and [0, π] respectively (Fig. 7). These latitudes and meridians span across a square plane of chosen dimensions (say l_o x l_o) and are related as follows:

$$\alpha_o = \left(\frac{1}{2} - \frac{y_o}{l_o}\right)\cdot\pi$$

$$\beta_o = \left(\frac{x_o}{l_o}\right)\cdot\pi$$

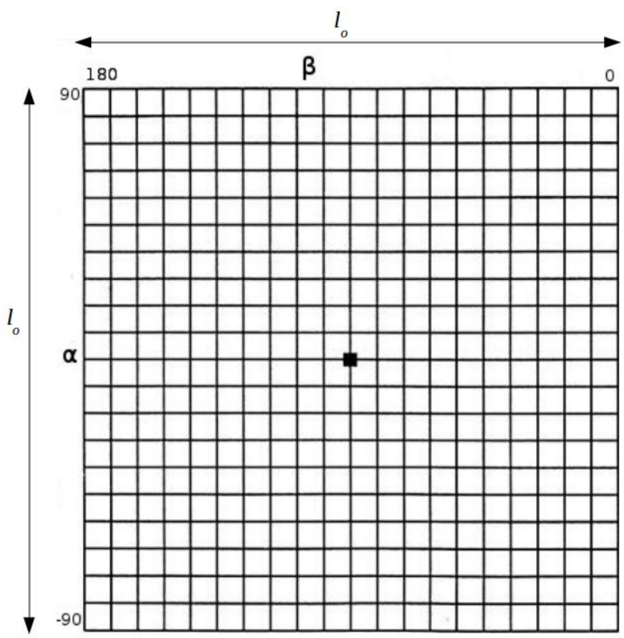


Fig. 7. Even distribution of latitudes and meridians on a plane, as in an equidirectional view.

Thus the pixel (x_o, y_o) of an equidirectional view can be mapped to its latitude and meridian values (α_o, β_o) which in turn can be mapped to spherical coordinates using equation (4). Using this and the mapping function of the fisheye camera one can map any pixel of an equidirectional view to a corresponding pixel in the input fisheye image. For our assumed mapping function:

$$R = f\,.\,\theta$$

the following relationship can be obtained between the pixel (x_o, y_o) of an equidirectional view and the pixel (x_i, y_i) of a fisheye image:

$$x_i = f\cdot\theta_o\cdot\cos(\phi_o)$$

$$y_i = f\cdot\theta_o\cdot\sin(\phi_o)$$

where ($\theta_o - \varphi_o$) is given by:

$$\theta_o = \arccos\left(\cos\left(\left(\frac{1}{2} - \frac{y_o}{l_o}\right)\cdot\pi\right)\cdot\sin\left(\frac{x_o\cdot\pi}{l_o}\right)\right)$$

$$\phi_o = \arcsin\left(\frac{\sin\left(\left(\frac{1}{2} - \frac{y_o}{l_o}\right)\cdot\pi\right)}{\sqrt{1-\cos^2\left(\left(\frac{1}{2} - \frac{y_o}{l_o}\right)\cdot\pi\right)\cdot\sin^2\left(\frac{x_o\cdot\pi}{l_o}\right)}}\right)$$

C. Stitching multiple equidirectional views

The images captured by closely placed cameras can be mapped onto the same equidirectional view using the transformations from section 2.2. These multiple equidirectional views can be stitched using any of the standard image registration techniques to obtain a global equidirectional view.

Fig. 8. Equidirectional view of an image of the left side of the scene.

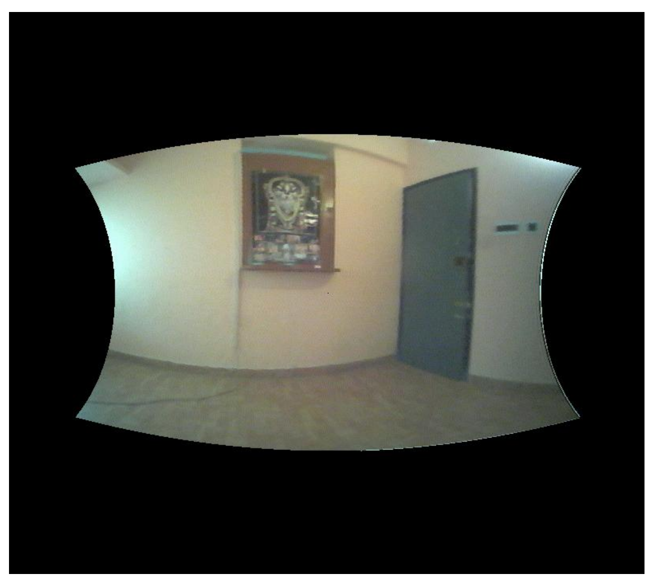


Fig. 9. Equidirectional view of an image of the right side of the scene.

To stitch the images shown in Fig. 8 and Fig. 9 we used a very basic image registration technique that minimizes pixel wise error (see [2]). The stitched global equidirectional view is shown in Fig. 10.

Fig. 10. Stitched global equidirectional view.

III. GENERATING PERSPECTIVE VIEWS

Perspective views can be generated if a mapping between output perspective plane and the global equidirectional view is known. For each point on the output plane, we must be able to locate the corresponding point on the global equidirectional view, which it should be mapped to.

Let us assume, the viewer (located at the origin) is viewing the scene at an angle (α_v, β_v) (where α_v represents the latitude and β_v represents the meridian).

For each point (x_p, y_p) of the output perspective image, we need to determine the corresponding point (x_o, y_o) on the global equidirectional view. We first determine the polar co-ordinates of the sphere (θ_s, φ_s) that should be mapped to the output point (x_o, y_o) when the viewing axis and the principal axis are coincident. If center of the output plane is treated as origin:

$$\theta_s = \arctan\left(\frac{\sqrt{x_o^2 + y_o^2}}{f}\right)$$

$$\phi_s = \arcsin\left(\frac{y_o}{f \cdot \tan(\theta)}\right)$$

Next we find coordinates that correspond to this (θ_s, φ_s) in the latitude meridian space, (α_s, β_s) using (3).

Now, since the viewing axis and principal axis have an angle of (α_v, β_v) between them, we calculate the actual location of the point on the sphere (α_{rot}, β_{rot}). Since (α_s, β_s) is known, we can calculate the Cartesian coordinates (x_s, y_s, z_s) of the point on the sphere using (2). To rotate these points by (α_v, β_v), these Cartesian coordinates are subjected to an affine transformation:

$$\begin{bmatrix} x_r \\ y_r \\ z_r \end{bmatrix} = R \cdot \begin{bmatrix} x_s \\ y_s \\ z_s \end{bmatrix}$$

where R is given by:

$$R = \begin{bmatrix} \sin(\beta_v) & -\sin(\alpha_v)\cdot\cos(\beta_v) & \cos(\alpha_v)\cdot\cos(\beta_v) \\ 0 & \cos(\alpha_v) & \sin(\alpha_v) \\ -\cos(\beta_v) & -\sin(\alpha_v)\cdot\sin(\beta_v) & \cos(\alpha_v)\cdot\sin(\beta_v) \end{bmatrix}$$

Once we obtain the rotated Cartesian coordinates (x_r, y_r, z_r), we can use (2) to obtain (α_{rot}, β_{rot}).

The point on the global equidirectional plane (x_o, y_o) that correspond to (α_{rot}, β_{rot}) can then be computed since we know the dimension of each equidirectional image (l_o x l_o):

$$x_o = \frac{l_o}{\pi} \cdot \beta_{rot}$$

$$y_o = \frac{l_o}{\pi} \cdot \alpha_{rot}$$

IV. RESULTS

We used a setup that included three fisheye cameras with a horizontal field of view 120 degrees, placed close to each other such that their principal axes lie in the same horizontal plane. The processing on the captured images was done using C routines running on a Linux – i686 system. Sample input images and their equidirectional views are shown in Fig. 11. The global equidirectional view generated by stitching them in shown in Fig. 12. The perspective views for various angles (α_v, β_v) that were generated using this global equidirectional view are shown in Fig. 13.

V. CONCLUSION

We have detailed the stages in an algorithm that allows us to take several different fisheye images and produce panoramas or perspective scenes from any arbitrary viewing direction. This involves the conversion of fisheye images from closely spaced fisheye cameras to their equidirectional projections. These equidirectional projections are stitched using simple image registration techniques to obtain a global equidirectional view. The global equidirectional view is mapped to perspective views for any arbitrary viewing direction.

REFERENCES

[1] Altera and MDN, *A flexible architecture for fisheye correction in automotive rear-view cameras.* http://www.altera.com/literature/wp/lit-wp.jsp.

[2] Altera and MDN, *Generating panoramic views by stitching multiple fisheye images.* http://www.altera.com/literature/wp/lit-wp.jsp.

[3] D. Salomon, *Transformations and Projections in Computer Graphics*. Springer-Verlag, London, 2006.

Fig. 11. Three input images (top row) and their corresponding equidirectional projections.

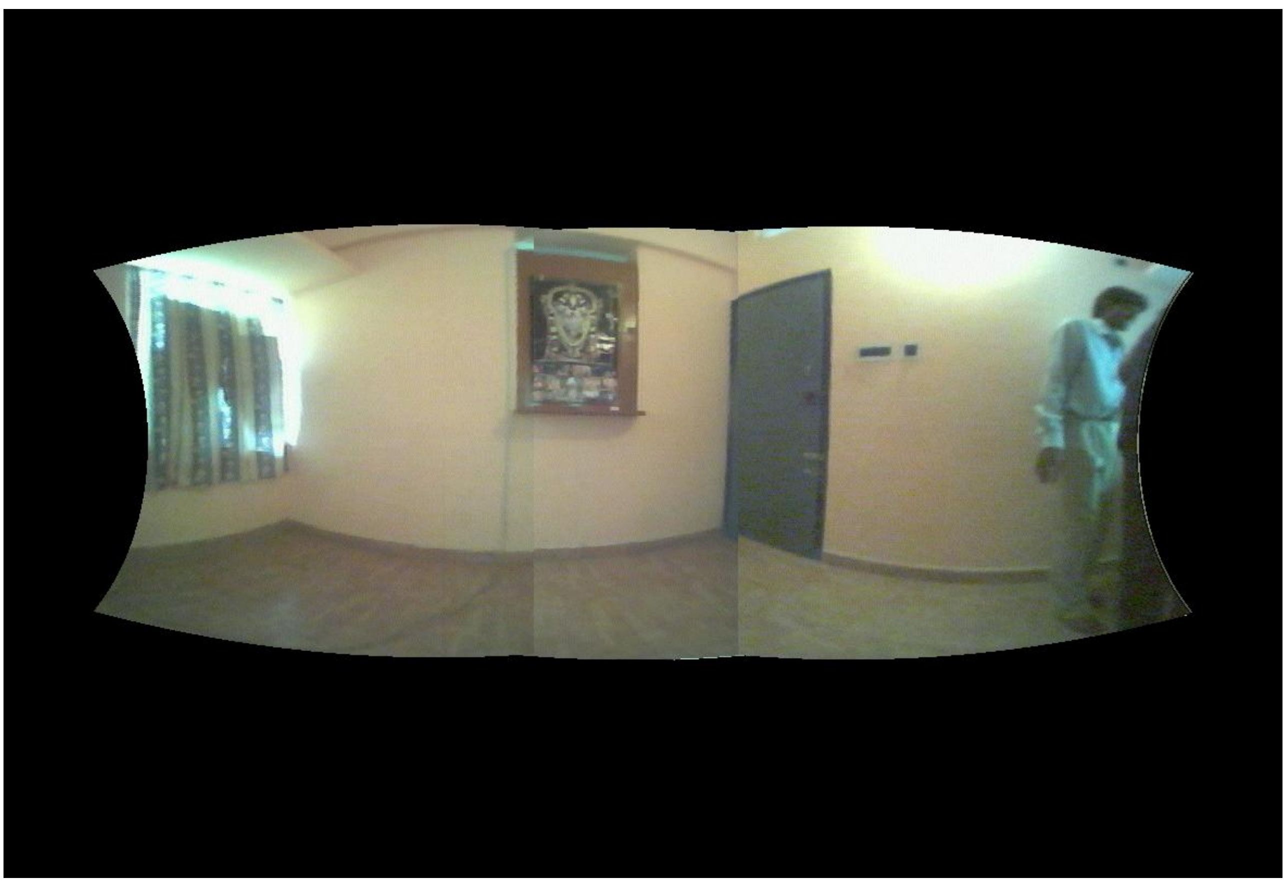


Fig. 12. Global equidirectional view.

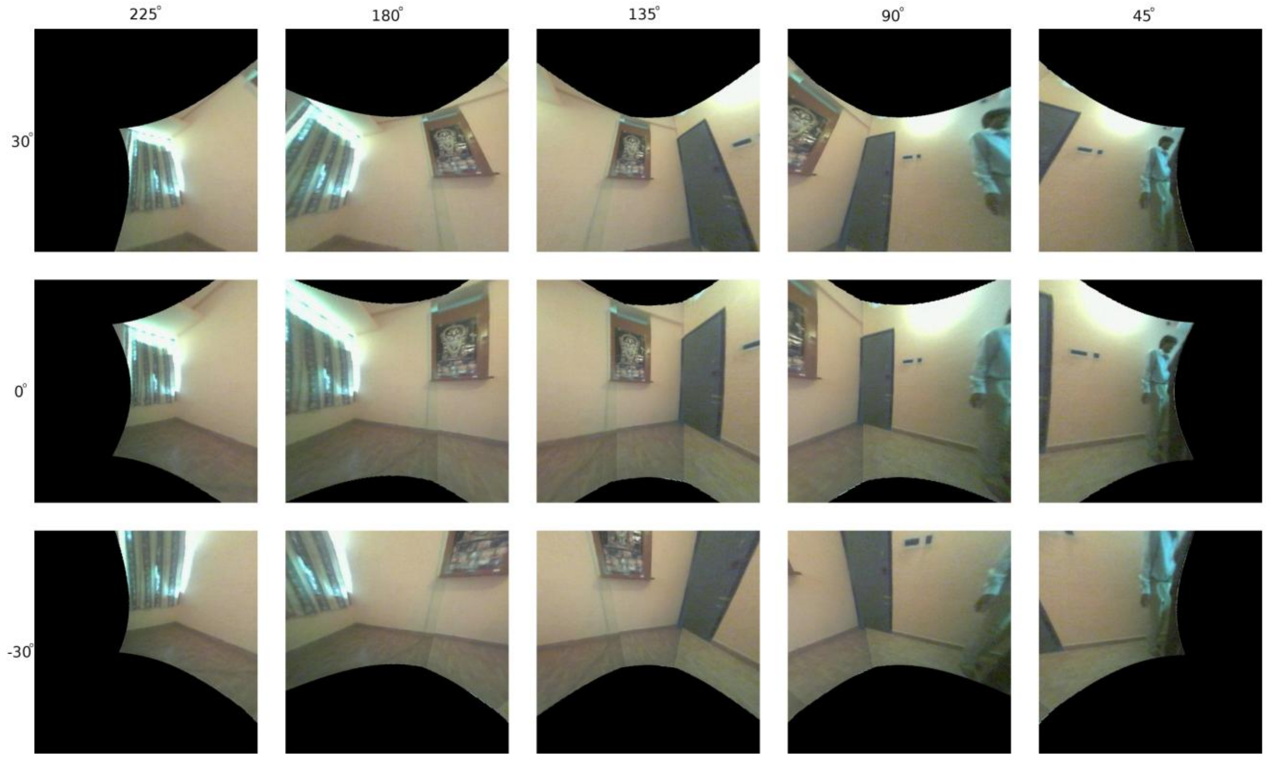


Fig. 12. Perspective views generated at various angles. Horizontal axis represents meridians and vertical axis represents latitudes.