

# Fisheye Lens Correction by Estimating 3D Location

Ji-Won Lee

Department of Electronics Engineering,  
Ewha Womans University  
52 Ewhayeodae-gil, Seodaemun-gu,  
Seoul 03760, Korea  
leejw9991@ewhain.net

Byung-Uk Lee

Department of Electronics Engineering,  
Ewha Womans University  
52 Ewhayeodae-gil, Seodaemun-gu,  
Seoul 03760, Korea  
bulee@ewha.ac.kr

## ABSTRACT

Since a fisheye lens can capture a wide angle scenery, it is broadly used for surveillance or outdoor sports. However, acquired images suffer from severe geometric distortions. Most of the existing distortion correction algorithms depend on linear features: images of linear features are first identified and then 2 dimensional warping is applied to make the curved images look straight. We propose a novel fisheye distortion correction method that estimates 3 dimensional (3D) locations of a foreground first, and then projects them to an image plane by perspective projection. When we know approximate distance of the foreground object, as in cases of head mounted camera, we can assume the 3D object plane of the foreground, and then estimate the 3D location from image points after internal camera calibration. For head mounted camera, foreground is a face and body of a human, and distortion of human figure is quite unnatural and awkward. Moreover, human figures lack linear features which excludes the use of conventional 2D warping techniques. We present techniques to estimate the 3D position from a corresponding 2D image point, which enables calculation of 3D object location. And then apply perspective projection to the 3D object position to obtain a distortion-free image. We demonstrate the efficacy of the proposed method using fisheye camera images and the applicability of the proposed concept to real applications.

## Keywords

Fisheye Lens, Lens Distortion Correction, Radial Distortion

## 1. INTRODUCTION

Fisheye lens can capture extremely wide view, therefore, it is useful for surveillance or outdoor sports camera. However, fisheye images have geometric distortion which is more severe toward the boundaries of the image. Moreover, depending on the position, foreground objects or a human is distorted severely, and then the fisheye image looks unrealistic. Therefore it is necessary to correct for the geometric distortion to restore natural and realistic figures.

There are several methods on correcting fisheye lens distortion by processing on viewing spheres [Sha10a], [Cha13a], [Car09a], [Wei12a], [Kan06a]. Sharpless [Sha10a] and Chang [Cha13a] introduced two-step projection methods from viewing sphere which can map from wide angle images into image planes. Sharpless firstly maps the sphere image into equirectangular space and then rectilinear-projects by adjusting a scale controlling a distance between the center of the projection and the view plane. He applied the scale into azimuth angle and altitude angle and

then obtained corrected image coordinates. It makes the regions to be horizontally compressed that are between the radial lines from the central vanishing point. This concept was based on the paintings of Gian Paolo Pannini and many painters of Italian Baroque period: they created the wide angle paintings using standard perspective projection, but the distortion of the projection was not shown in the paintings. Sharpless [Sha10a] adopted the method to correct for the distorted wide angle scenery. Chang [Cha13a] further developed Sharpless' two step projection method from the viewing sphere to an image plane in [Cha13a]. For an initial projection, Chang [Cha13a] introduced a swung surface which was created by finding parameters linearizing line segments, which were found in 6 faces obtained by box projection. After the initial projection to the surface, the surface was projected to image plane using perspective projection. Then, this method conserved the linearity of horizontal lines better than those of vertical direction. While two methods established projection models, Carroll [Car09a] and Wei [Wei12a] adaptively corrected for the fisheye lens distortion by using user inputs, such as line constraints that should be conserved as straight lines. Carroll [Car09a] used line constraints users entered and dealt with the straightness of them, neighbor pixels' conformality and smoothness to warp each pixel in order to obtain natural images having little fisheye distortion. In other

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

words, for each pixel, they find a new image coordinates optimizing sum of several energies such as straightness, conformality and smoothness. In [Car09a], Carroll used static images, while Wei [Wei12a] utilizes video images of several frames. Wei [Wei12a] carried out the above process tracing input line constraints over consecutive frames. However, these methods could not work properly when the number of inputs were inadequate or the fisheye images were severely distorted.

We propose a new fisheye lens distortion correction method using 3D position information with a prior on the distance, whereas current methods focus on conserving the linearity of straight lines, and mapping in 2D. In a situation of capturing an outdoor activity by a head mounted camera, the distance between the human and the camera is almost fixed, thus we can use the distance between them for distortion correction. If the foreground is assumed to be on a plane of known distance, we can calculate the 3D position of the foreground from images coordinates after inverse projection. Then, we can project these 3D coordinates using perspective projection, and reconstruct the foreground image without nonlinear geometric distortion. Usually, the foreground and camera is close and the solid angle covered by the object is considerable, which results in severe image distortion. Substantial distortion of human face or body is quite unpleasing, however, because there is no straight line features in human images, it is not possible to apply the existing distortion correction methods. The proposed algorithm does not apply 2D image warping; the novel method estimate the 3D position of the foreground object from a prior and the image position, and then apply perspective projection to the estimated 3D location for generating distortion corrected images.

In order to correct for the distortion of foreground in fisheye lens images, we firstly segment the foreground in captured images, estimate 3D coordinates of them assuming the distance to the foreground, and then project 3D coordinates into the image coordinates by perspective projection where the mapped coordinate is inversely proportional to the distance to the lens.

The paper is organized as follows. Section 2 describes the proposed method which is estimating 3D location of foreground and then perspective projection of the foreground to an image plane. Section 3 shows the experimental results. Lastly, Section 4 presents conclusions and future work.

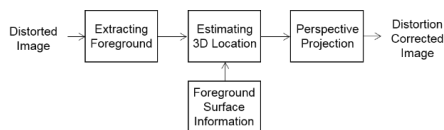


Figure 1. Block diagram of the proposed method

## 2. PROPOSED METHOD

We describe an innovative concept of distortion correction: virtual 3D coordinates of foreground objects are estimated first and then re-project them to obtain distortion-free images.

Figure 1 shows a block diagram of the proposed process. As shown in Figure 1, we initially segment the foreground object from an image, and then estimate 3D locations of the foreground using image coordinates assuming planar object plane. The distance and the orientation of the plane is assumed to be known. For head mounted cameras, the distance from the camera to the foreground is stable and the image distortion is not sensitive to perturbation of the distance, therefore, distortion correction can be accomplished.

Once 3D coordinates are estimated, geometric distortion can be completely eliminated without linear features. We can apply perspective projection to generate an image without nonlinear distortion.

### 2.1 Internal Calibration

Fisheye camera lens model can be characterized as a function of image distance  $r$  and an angle  $\theta$ , where  $r$  is the distance between the principal point and the image position  $p$  (in Figure 2(b)) on the image plane, and  $\theta$  represents the angle between the incoming ray from the 3D point  $P$  (in Figure 2(b)) and the optical axis of the camera. Figure 2(a) shows curves of several fisheye lens models and Figure 2(b) represents an image plane  $(x, y)$  and the camera lens coordinates in 3D space  $(X_c, Y_c, Z_c)$ , which represents the lens model.

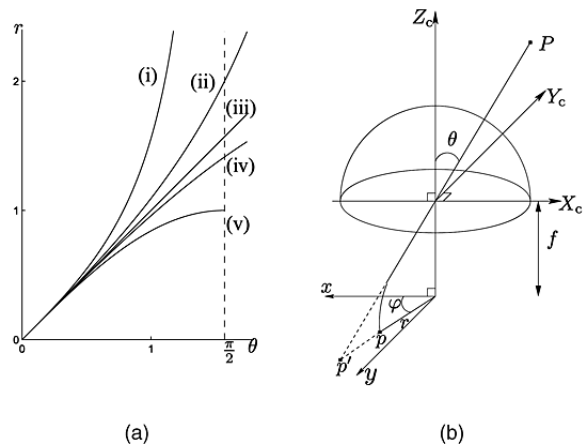
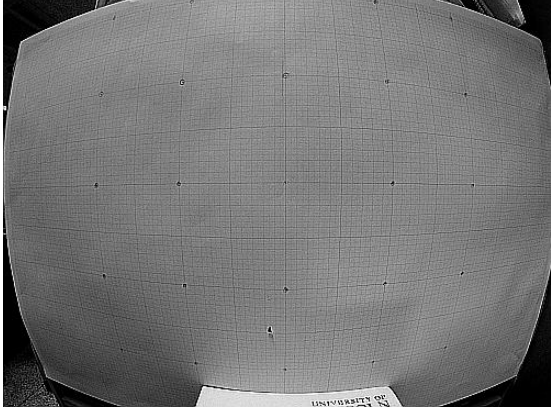


Figure 2. (a) Various models of fisheye lens. Lens models can be represented as  $r - \theta$  curve with focal length  $f = 1$  where  $r$  is the distance from the principal point to a point  $p$  in the image plane and  $\theta$  is an azimuth angle of object position  $P$ . (b) Fisheye camera model setup. The image point of the point  $P$  is  $p$  whereas it would be  $p'$  by a pinhole camera [Kan06a].



**Figure 3. Fisheye image of a grid pattern for internal calibration captured by GoPro HERO4 Silver. Optimized focal length and principal point are found by using the coordinates  $(u, v)$  of this image.**

We used GoPro HERO4 Silver camera model and we found that the orthographic model ((v) in Figure 2(a)) fits the distortion of the camera. In Figure 4,  $\theta - r$  curves from experiment (red dot) and by orthographic lens model (blue line) are plotted together, which shows good agreement of the observed data and the orthographic model. The proposed method can be applied to any lens distortion model.

By finding an optimized focal length and principal point for this model, we enhance the accuracy for estimating the 3D coordinates. For orthographic lens, projection can be described by the following equation,

$$r = f \sin \theta, \quad (1)$$

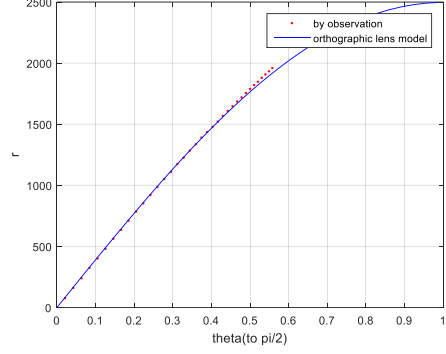
where  $r$  is the image distance from the principal point  $(u_0, v_0)$  to an image point  $(u, v)$ . We used a grid pattern perpendicular to the optical axis as shown in Figure 3. Then the tangent of angle  $\theta$  is  $\frac{\ell}{d}$  where  $d$  is the distance from the optical center to the center of the grid pattern  $(x_0, y_0)$  and  $\ell$  is the distance from  $(x_0, y_0)$  to the grid point  $(x_p, y_p)$ . The image position of  $(x_0, y_0)$  corresponds to the principal point. We applied affine transform to establish a mapping of the center of the grid pattern and the principal point [Lee16a]. To find the optimized internal parameters, focal length and principal point, we find the minimum mean square error solution:

$$\arg \min_{f, u_0, v_0} \sum_i \frac{(r_i - f \sin \theta_i)^2}{N}, \quad (2)$$

where  $N$  is the number of data points.

## 2.2 Estimating 3D Location of Foreground

Once we have an equation of the foreground plane in 3D space, then we can find the 3D location of a point from a calibrated image point. The intersection of the optical axis and the foreground plane is  $(0, 0, d)$  and the tilt angle is  $\alpha$ ; the rotation axis of tilt is the



**Figure 4. Observed  $\theta - r$  curve (red dot) is obtained by calculating  $\theta$  and  $r$  using 36 image points which are 1cm apart horizontally from the principal point of the image (max  $\theta = 56.2^\circ$ ). Theoretical  $\theta - r$  curve (blue solid line) is drawn from the orthographic lens model.**

$y$ -axis (in Figure 5) for simplicity. We derive the 3D locations of the foreground by finding the intersection of the image ray and the object plane.

Figure 5 shows a 3D space with a surface and  $x, y, z$ -axis. Also the center of the lens is shown as the origin  $\mathbf{O}$ . As shown in this figure, we can derive the coordinates on the object surface  $(x_p, y_p)$  when we know image coordinates  $(u, v)$ .

Figure 5 also illustrates the relationship of the object surface coordinates and the image coordinates. The principal point  $\mathbf{O}'$  is  $(u_0, v_0)$ , and the angle between a ray from the object center to a surface point  $P$  and a horizontal axis of the surface is  $\varphi$  which is obtained from the following equation,

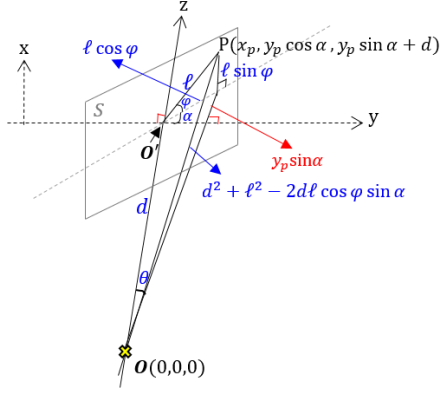
$$\varphi = \tan^{-1} \left( \frac{u - u_0}{(v - v_0) / \cos \alpha} \right), \quad (3)$$

$u$  and  $v$  are the image coordinates along the  $x$ -axis and  $y$ -axis of Figure 5, respectively. Since the foreground surface  $S$  is tilted by the angle  $\alpha$  along the axis of  $y$ , we divide  $v - v_0$  by  $\cos \alpha$  to compensate for the foreshortening. The direction angle  $\varphi$  on the foreground surface can be obtained from the image coordinates because we know the tilt angle of the foreground surface. The vertical axis of the surface is the same as that of the image plane, whereas the horizontal axis is rotated by  $\alpha$ . And then, we can calculate the image distance  $r$ ,

$$r = \sqrt{(u - u_0)^2 + (v - v_0)^2}, \quad (4)$$

which is the Euclidean distance from the principal point  $(u_0, v_0)$  to an image point. Moreover, we can obtain the angle  $\theta$ , in the 3D space, using the calibrated lens model of the equation 1.

With the known angles  $\varphi$ , and  $\theta$ , we are ready to calculate the distance  $\ell$  which is the distance from the center of the object surface to a point on the



**Figure 5.**  $(x_p, y_p)$  of the foreground surface  $S$  can be derived using the 3D relations in terms of the known priors, such as the distance  $d$  between the center of the lens  $O$  and the center of the surface  $O'$  and tilt angle of the surface  $\alpha$ , and parameters  $\theta$  and  $\varphi$  derived from image coordinates. Also the perspective projection is the mapping of each 3D coordinate to the image position depending on the value  $z$ . The depth  $z$  is calculated from  $\ell$  as in the figure because we know  $\varphi$  and  $\alpha$  already from the image.

surface. The equation for  $\ell$  can be derived as,

$$\ell = \frac{-d \sin \theta (w \sin \theta - \sqrt{1 - w^2} \cos \theta)}{w^2 - \cos^2 \theta}, w = \cos \varphi \sin \alpha. \quad (5)$$

Since the image direction angle  $\varphi$  is obtained by equation 3, we could calculate the object image location  $(x_p, y_p)$  on the surface plane by the following formula,

$$(x_p, y_p) = (\ell \cos \varphi, \ell \sin \varphi) \quad (6)$$

In other words, the surface coordinates are reconstructed from the image coordinates and object surface information through equations 3 to 6.

### 2.3 Rendering an Image without Distortion

In order to render an image without distortion, we apply perspective projection to the surface coordinates  $(x_p, y_p)$ . And then, perspective projected foreground is overlaid on the input image so that we can obtain a distortion corrected foreground over a wide angle background image.

Firstly, we need to find the 3D coordinates of the object surface points so that we can apply perspective mapping. The 3D coordinates  $(x, y, z)$  are derived by the following equation:

$$x = x_p \quad (7)$$

$$y = y_p \cos \alpha \quad (8)$$

$$z = y_p \sin \alpha + d, \quad (9)$$

the relation between an object image position  $(x_p, y_p)$  which is the surface coordinates and the 3D camera coordinates  $(x, y, z)$  are shown in Figure 5.

Once we obtain the 3D position, we can calculate image coordinates which are finally mapped to the corrected image. At this stage, we adjust the range of  $z$  values because it controls the ratio of the sizes of the near and far foregrounds.

We refer the parameter to adjust the range of  $z$  as  $zRangeScale$ , and the scaled  $z$  values are represented with  $z_s$  as in the following equation,

$$z_s = (z - m_z)zRangeScale + m_z, \quad (10)$$

where  $m_z$  is the minimum  $z$  value of the foreground and  $zRangeScale$  is the  $z$ -length of the foreground object.

After finding  $z_s$  for each  $(x_p, y_p)$ , we derive a corresponding image coordinates  $(u, v)$  which is the result of perspective projection by the following formula,

$$u = g_s (m_z / z_s) x + u_0 \quad (11)$$

$$v = g_s (m_z / z_s) y + v_0, \quad (12)$$

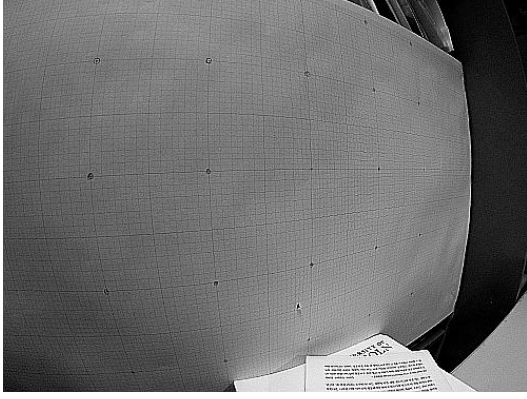
where  $g_s$  controls the global size of the foreground object on the resulting image plane. The principal point  $u_0$  and  $v_0$  are added to shift the  $(0, 0)$  principal point of the camera model to the actual principal point on the image coordinates.

## 3. EXPERIMENTS

After finishing internal camera calibration which estimates internal parameters of the camera, such as the focal length and the principal point, we estimate the 3D coordinates of an object from image coordinates. In this section, we verify the accuracy of each step and analyze experimental results.

### 3.1 Estimation of 3D Coordinates

We applied the calibration process which is described in Section 2.1 by using an image of a tilted grid pattern. This tilted grid pattern image has distortions, therefore the same interval is shown differently depending on the position in the image as in Figure 6. However because the grid pattern shows the ground truth location, we can find the error of the estimated position easily. We verified that the derived equations for calculating the 3D object plane coordinates  $(x_p, y_p)$  are accurate with the test results. In this experiment, we optimize the distance  $d$  and the angle  $\theta$  by using 30 image coordinates of horizontal points on the tilted grid image of Figure 6.



**Figure 6. A fisheye image of a tilted plane captured by GoPro HERO4 Silver. The plane is set up away from the lens by 30cm, and rotated horizontally by  $30^\circ$  ( $d \approx 30\text{cm}$ ,  $\alpha \approx 30^\circ$ ).**

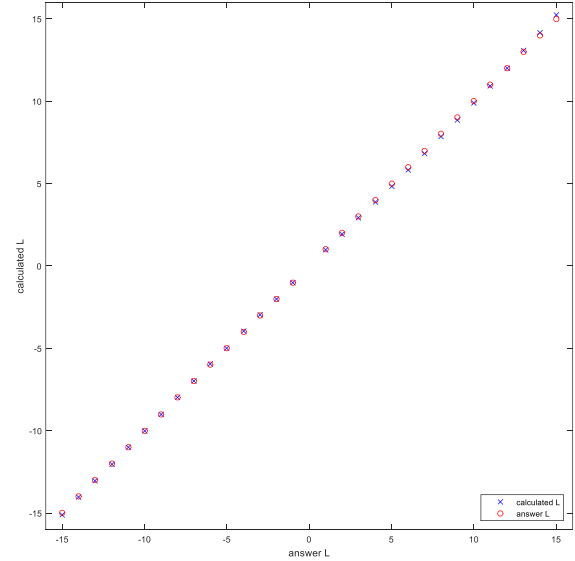
We can verify the accuracy of the calculated coordinates  $(x_p, y_p)$  because we know the true coordinates by the reading the graph and then compare with the calculated position. We represent a result showing both the true and obtained  $\ell$  which is the distance from the center of the foreground surface, and calculate an error between them as in Figure 7. We use Figure 6 for this task. It is an image of a grid pattern 30 cm away and with  $30^\circ$  tilt.

In Figure 6, horizontal 30 points of 1 cm interval are used for calculating foreground image position  $(x_p, y_p)$ . The distances  $\ell$  between the center of the grid paper (0,0) and the coordinates  $(x_p, y_p)$  are shown with the corresponding ground truth coordinates in Figure 7. Also we calculate the maximum absolute error (MAE) and mean square error (MSE) for the 30 data points after optimizing the  $d$  and  $\alpha$ . The errors show quite accurate results: the root mean square error of 30 points is 0.1 cm and the maximum error is 0.24 cm. Therefore the defined equations are reliable for calculating the 3D coordinates. Based on the fact that the derived equations can estimate the 3D positions with high reliability, we can reconstruct the 3D foreground surfaces from image coordinates of real scenes in the following subsection with high accuracy.

### 3.2 Perspective Projection Method

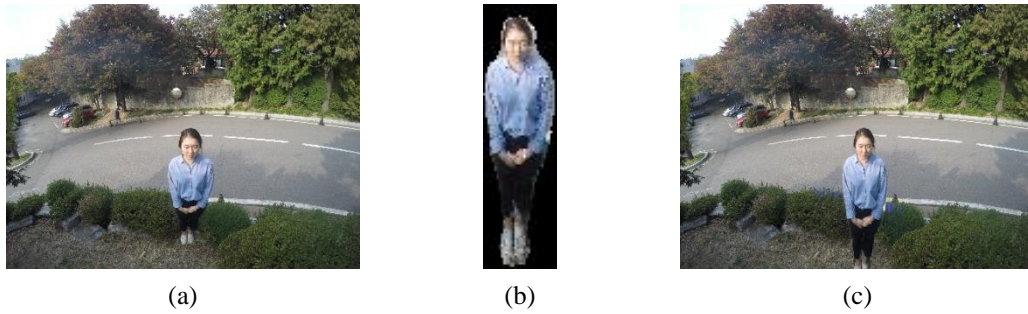
We apply a perspective projection for re-projecting the obtained 3D coordinates to the image plane to correct for the fisheye lens distortion of the foreground image. We use indoor/outdoor real images for finding 3D coordinates of foregrounds, and then reconstruct a distortion-free result image by the perspective-projection method of section 2.3.

We correct for the geometric distortion of a human figure in Figure 8(a). Firstly, a foreground object is segmented using MATLAB toolbox Image Segmenter.

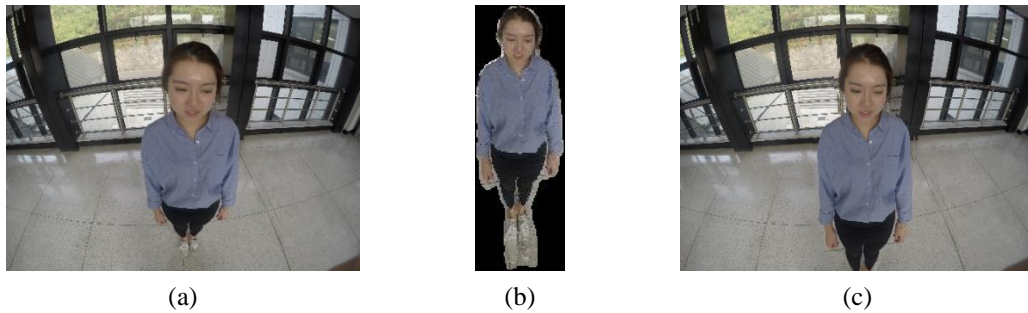


**Figure 7. Calculated (blue  $\times$ ) and ground truth (red  $\circ$ ) distance  $\ell$ .  $\ell$  is calculated by the proposed equations using the foreground information of the distance, the tilt angle, and the image coordinates.**

And then if we know the distance from the lens to the 3D foreground surface and the tilt angle of the foreground, then we can derive 3D coordinates of every foreground pixel from the extracted image coordinates and the prior. The reconstructed object surface from the original image is represented in Figure 8(b). The coordinates of the object plane is  $(x_p, y_p)$  as explained in Section 2.2. The recovered foreground is perspective projected and overlaid to the original image as shown in Figure 8(c). We can notice that the human figure looks more natural after the distortion correction using the proposed method. It is hard to find linear features in this human figure image, therefore, previous distortion correction cannot be applied to this class of images. We apply the correction method to Figure 9(a) and the resulting image, Figure 9(c), shows more familiar looking human figure with better proportions. The background of this image is window frames and floor, which can be approximated as two separate planes in 3D space. We applied the same correction algorithm to the background, and the result image is Figure 10. Input image Figure 9(a) is divided into the figure foreground, the floor background and the window frame background, and then 3D position of each object plane is recovered and then perspective projected separately. This result shows excellent correction of distortion of the human figure and the background. Merging of the foreground and background is not perfect yet, therefore, we notice mismatch between them. The performance of the proposed distortion correction algorithm heavily depends on segmentation and merging of the foreground.



**Figure 8. (a) Original Image, (b) recovered foreground image, (c) overlay of the perspective projection of the foreground**



**Figure 9. (a) Original Image, (b) recovered foreground image, (c) overlay of the perspective projection of the foreground**

#### 4. CONCLUSION

Most of previous researches corrected fisheye lens distortion by finding features of straight lines and then applied 2D warping to make the images of linear features straight. We propose a novel correction method for fisheye lens distortion. For cases where the distance between the lens and the foreground object/human is stable, for example head mounted cameras, we propose an algorithm to correct for the fisheye distortion by estimating 3D locations of the foreground. In order to fulfill this objective, we derived techniques for estimating 3D positions from image coordinates of the foreground and apply a perspective mapping to the estimated 3D coordinates to render a distortion-free image of the foreground. Future work includes application of the proposed method when the 3D foreground is composed of many planes or curved surfaces. Estimation of the depth or distance to the foreground is also an interesting topic.



**Figure 10. Perspective projection of the background (the window frames and the floor) with the overlay of the perspective projection of the foreground human figure.**

#### 5. REFERENCES

- [Car09a] Carroll, R., Agrawala, M., & Agarwala, A. Optimizing content-preserving projections for wide-angle images. *ACM Transactions on Graphics*, **28**(3), Article No. 43, 2009.
- [Cha13a] Chang, C. H., Hu, M. C., Cheng, W. H., & Chuang, Y. Y. Rectangling Stereographic Projection for Wide-Angle Image Visualization. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2824-2831, 2013.
- [Kan06a] Kannala, J., & Brandt, S. S. A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **28**(8): 1335-1340, 2006.
- [Lee16a] Lee J., Lee B. Optimized modeling of orthographic fisheye lens by estimating principal point. In *Proceedings of Image Processing and Image Understanding Workshop*, Vol. 28, pp. 727-729, 2016.
- [Sha10a] Sharpless, T. K., Postle, B., & German, D. M. Pannini: a new projection for rendering wide angle perspective images. In *Proceedings of the Sixth international conference on Computational Aesthetics in Graphics, Visualization and Imaging*. Eurographics Association, pp. 9-16, 2010.
- [Wei12a] Wei, J., Li, C. F., Hu, S. M., Martin, R. R., & Tai, C. L. Fisheye video correction. *IEEE Transactions on Visualization and Computer Graphics*, **18**(10): 1771-1783, 2012.