

A GENERALIZED NON-LINEAR METHOD FOR DISTORTION CORRECTION AND TOP-DOWN VIEW CONVERSION OF FISH EYE IMAGES

VIVEK SINGH BAWA^{✉,1}, KRISHAN KUMAR², VINAY KUMAR¹

¹Electronics and Comm. Engg. Department, Thapar University, Patiala, India; ²Computer Vision Department, Add Innovations, Greater NOIDA, India

e-mail: bawa.vivek.01@gmail.com; krishanv911@gmail.com; vinay.kumar@thapar.edu

(Received November 5, 2016; revised April 14, 2017; accepted May 20, 2017)

ABSTRACT

Advanced driver assistance systems (ADAS) have been developed to automate and modify vehicles for safety and better driving experience. Among all computer vision modules in ADAS, 360-degree surround view generation of immediate surroundings of the vehicle is very important, due to application in on-road traffic assistance, parking assistance etc. This paper presents a novel algorithm for fast and computationally efficient transformation of input fisheye images into required top down view. This paper also presents a generalized framework for generating top down view of images captured by cameras with fish-eye lenses mounted on vehicles, irrespective of pitch or tilt angle. The proposed approach comprises of two major steps, viz. correcting the fish-eye lens images to rectilinear images, and generating top-view perspective of the corrected images. The images captured by the fish-eye lens possess barrel distortion, for which a non-linear and non-iterative method is used. Thereafter, homography is used to obtain top-down view of corrected images. This paper also targets to develop surroundings of the vehicle for wider distortion less field of view and camera perspective independent top down view, with minimum computation cost which is essential due to limited computation power on vehicles.

Keywords: center of distortion, homography, lens distortion correction, surround view, top down view

INTRODUCTION

With increasing number of road accidents, driver assistance system is becoming progressively popular. The system not only tries to ensure the safety of people behind the wheels but also prevents accidents and provides support for better and easy driving. The 360-degree surround view system provides a comprehensive image of the vehicles' immediate surroundings, helping drivers to keep a track of obstacles around, highlights lane markings, eliminates blind spots and provides precise maneuvering as a part of driving assistance. To achieve goals of surround view, image sensors should continuously capture very wide angle images. These images are processed and delivered to monitoring module in real time. In surround view system (SVS), hardware setup containing cameras, processing unit and display unit is installed on vehicle. The images captured by cameras are sent to image-processing unit; finally processed images are displayed on the LCD or Central Information Display. In our

design, four cameras are installed around the vehicle in this system: one on both front and rear bumpers, and rest on both side view mirrors of the car. The detailed setup is shown in (Fig. 1).

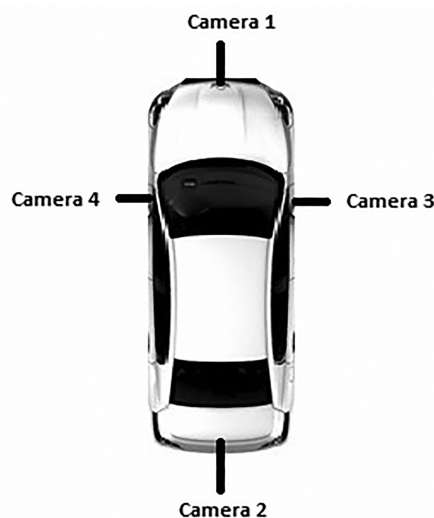


Fig. 1. Camera setup around the vehicle.

In some previous designs, five or six cameras are installed; four on each corner of the car and one on rear bumper (five cameras) and one on front bumper (six cameras). The choice of four cameras was made in light of the prime aim of this research; *i.e.*, to produce best output results of surround view construction while keeping computation and hardware cost to minimal. Since cameras need to cover a large area of vehicle's surroundings without loss of much data, best option is to use fish eye lenses. The ultra-fish eye lenses with 180° horizontal field-of-view (FOV) are used in this research.

In proposed work, cameras are installed with some degree of tilt towards the direction of FOV from the horizontal axis, so that wider region of interest (ROI) around the vehicle is captured. This helps in providing better visibility of surroundings to the driver, in turn increasing accuracy of the system. The cameras used in the system has frame rate of 30 fps to provide surrounding conditions in real time. This means, SVS should provide output at the same frame rates as camera; so that the system is able to deliver real time performance in motion.

Before embarking on this project, extensive literature review was done, which portrays that many techniques have been proposed and implemented till date. (Kedzierski *et al.*, 2008) used arc curvature and differential geometry of the input image for fisheye correction. The fish eye correction algorithm based on circle fitting was proposed by (Burchardt *et al.*, 2001). Li and Hartley (2005) proposed a non-iterative method by taking a number of correspondences from two views to calibrate radial distortion. This approach is independent of camera parameters and to convert the images into bird's eye view, calibration matrix is extensively used. Yu and Ma (2014), Li *et al.* (2014), Santhanam *et al.* (2014) and Dhane *et al.* (2012) have used reference points to convert fish eye images into top-down view images. Lin *et al.* (2012) has used top view transformation model for perspective transformation and involves an iterative method to determine the best results, if camera parameters are not known. Melo *et al.* (2013) and Brito *et al.* (2013) have developed techniques for calculation of center of distortion (COD) before calculating radial distortion. Both the papers used slope of lines in natural images or pre-designed pattern to find the COD for all the curves. Although this paper used slope of the lines, but this work tries to find the lines in the pattern with lowest slope to estimate COD. All of the above methods except (Dhane *et al.*, 2012) utilizes forward mapping that further requires interpolation to fill in the vacant

pixels, thus increasing computation efficiency and time. Proposed method uses inverse mapping technique similar to technique discussed by (Dhane *et al.*, 2012) for lens distortion correction. After practical implementations of conventional mapping and inverse mapping, later was found to reduce computational complexity as it eliminates need of interpolation.

Contributions of this work are three folds:

1. This is the first work for vehicle surround view to consider COD. A fast and efficient pattern based approach has been developed for accurate COD detection.
2. A mapping based algorithm is developed, where look up tables (LUT) are used to map pixels.
3. A novel technique has been developed to estimate homography matrix parameters when input camera projection plain changes.

Rest of the paper is divided into following sections: lens distortion correction discussed in Section II, homography, top-down view transformation and a novel technique of computation of homography matrix based on camera tilt using generalized point selection are also given in Section II, results are presented in Section III followed by discussion in Section IV.

MATERIALS AND METHODS

LENS DISTORTION CORRECTION

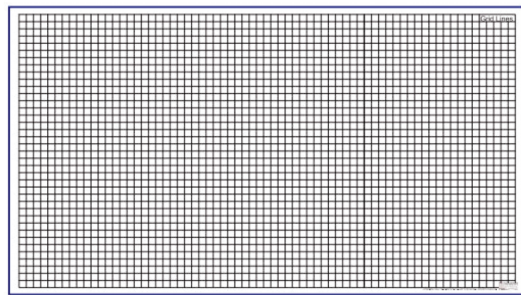
Most of the fish-eye lenses provide a wide field-of-view (170° or more sometimes). These lenses have a short focal length causing the lens to have wider FOV. With this advantage, the images captured by the fish-eye lenses have a disadvantage of being distorted; *i.e.*, the straight lines do not appear straight anymore and there is a loss of rectilinear projection. The captured images have both radial as well as tangential distortion, since the tangential distortion is negligible; therefore, only radial distortion is taken into account. Fish-eye lenses produce a zooming effect that makes objects appear bigger near optical center compared to objects at distance. The resulting images appear as if being watched through a long barrel, thus called as 'barrel distortion'. This effect causes objects towards the image boundary to get cramped in relatively lesser number of pixels. In the proposed method, barrel distortion is corrected in two steps: radial correction and stretching, which result in corresponding LUTs. These LUTs are further used for inverse mapping.

Generally optical axis of the lens and axis of camera sensor are not aligned. Distortion at any given coordinates in the image is dependent upon the distance of that pixel from the center of lens. Till now, effects of misalignment of sensor center and COD has not been considered in 360°surround view construction (SVC) technique.

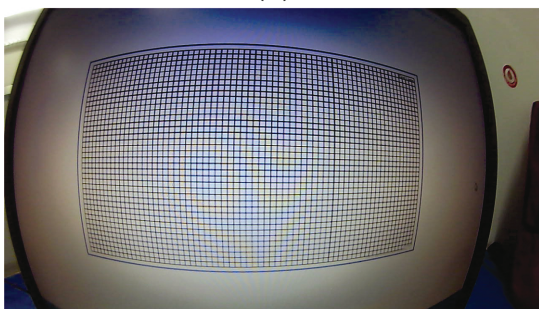
Center of distortion estimation (COD)

Ideally, center of distortion and center of image should overlap, but in real camera modules these centers are slightly mismatched. This happens due to misalignment of lens and sensor in camera module. A simple and accurate method for COD estimation is developed in this paper, which follows the steps given below:

- a) ‘n’ images of a grid shown in (Fig. 2a) are captured with ultra-fish-eye lens resulting in images shown in (Fig. 2b).
- b) In each image, two orthogonal lines with minimum curvature are selected (found near the center of image).
- c) The intersection of these orthogonal lines is considered as COD of corresponding image.
- d) This process is repeated for both horizontal and vertical lines in selected n images to get n COD co-ordinates.
- e) The final COD is estimated by taking average of these n CODs, (Fig. 2b).



(a)



(b)

Fig. 2. (a) Grid of lines (b) Fish Eye Image of grid of lines.

In above algorithm, RANSAC technique (Schnabel *et al.*, 2007) is used to estimate the best fit line segment. This algorithm was implemented using n = 20, multiple samples help avoiding COD calculation error.

Position of any pixel in image is calculated using COD as origin of coordinate system. All the distance and angles, in further equations, are calculated with respect to COD.

Radial distortion correction

Barrel distortion is present all over the image and is symmetric around COD. All the pixel coordinates are referred in polar coordinate system with COD of image as origin. Let image generated by fish eye lens be FP (fish-eye projection) and image for real world plain be represented by WP (world plain image without fisheye distortion). Let distance of pixel from COD in FP image be ‘r’ and distance of pixel in WP be ‘R’ and angle made by a pixel with the camera axis in FP and WP are represented by ‘α’ and ‘Θ’, respectively. Focal length of camera lens is ‘f’. The intuitive representation of WP and FP are shown in (Fig. 3).

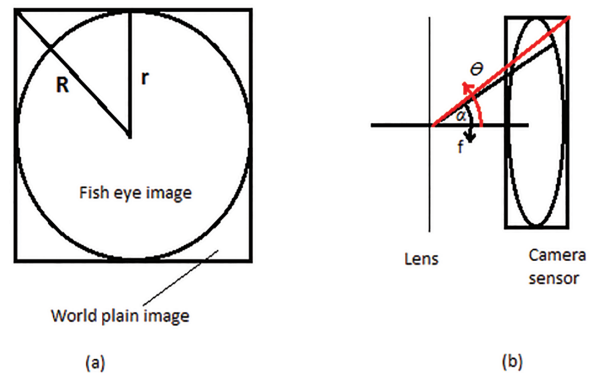


Fig. 3. Image representation in image plain with and without fish-eye distortion. (a) Pixel distance from center: R is the distance without distortion and r is pixel distance and space compression due to fish eye lens. (b) The angle made by pixel: α and Θ are angle with camera axis with and without distortion respectively.

Distance from origin for every output pixel (R) is calculated using the r of input pixel. Prior to that, the dimensions of the output (radially corrected) image are calculated using the following equations:

$$H = 2 * f * \tan \left(\sin^{-1} \left(\sin \left(\tan^{-1} \left(\frac{\text{height}}{2 * f} \right) * \mu \right) \right) \right), \quad (1)$$

$$W = 2 * f * \tan \left(\sin^{-1} \left(\sin \left(\tan^{-1} \left(\frac{\text{width}}{2 * f} \right) * \mu \right) \right) \right), \quad (2)$$

$$r = f * \tan \left(\sin^{-1} \left(\frac{\sin(\tan^{-1} \frac{R}{f})}{\mu} \right) \right), \quad (3)$$

where, μ = refractive index of the lens material (ranging 1-1.15), f = focal length of the camera lens. H and W are height and width of the corrected image respectively.

If focal length is not available, it can be determined using image dimensions captured from that same camera.

$$f = \frac{\text{Framewidth}}{4 * \sin\left(\frac{\text{field of view}}{4}\right)}. \quad (4)$$

Using Eq. 3, a inverse mapping matrix is developed which will describe backward relationship from pixel coordinates of radially corrected image (R, a) to the input fish-eye image (r, θ).

Pixel stretching

Radially corrected image is stretched to generate a conventional image with horizontal and vertical stretching. Stretching procedure is mainly performed to construct the image boundaries, which are left vacant after radial correction. Radially corrected image still has distortion; moving away from the center in horizontal and vertical direction, quantity of vacant boundary pixel keeps on increasing. The proposed method uses inverse mapping for stretching. For vertical stretching, the corresponding input y-coordinate is calculated for every y-coordinate of the output image. The y-coordinate of every output pixel is represented as Y , whereas y-coordinate of every input pixel is 'y' in original image, and the following equation is used for the mapping of input pixel to output stretched image pixel:

$$y = cf * \tan \left(\sin^{-1} \left(\frac{\sin(\tan^{-1}(\frac{Y}{cf}))}{s} \right) \right), \quad (5)$$

where, s = stretching factor that determines the maximum stretching, oh = height of output image, ih = height of input image for this stage, cf = correction factor, referring to the overall stretching for the entire image pixels,

$$s = \frac{oh}{ih}. \quad (6)$$

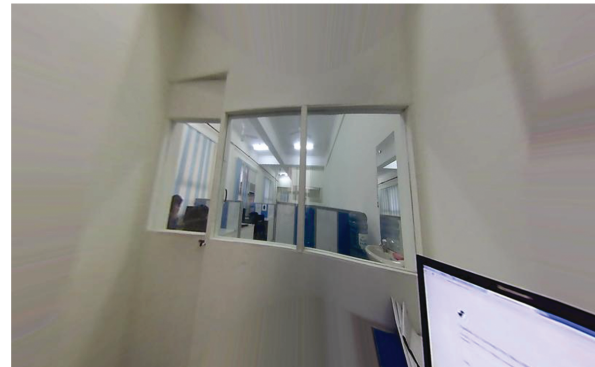
The correction factor cf is experimentally determined constant linear multiplier. The radial distortion increases non-linearly with radius; therefore theoretically a nonlinear cf (directly proportional to pixel radius) would result in better image. But when the

experiments were conducted for non-linear cf , they did not show any significant improvement in image quality. Therefore, final algorithm was set with constant cf . In other words, if a column is selected in radially corrected image; pixels at both the ends have zero pixel value. Vertical stretching extrapolates central values to generate the complete column. Same process is repeated for every row and column in image.

Barrel distortion (Fig. 4a) is evident in center of the image. The vertical lines appear curved and the image center is zoomed. Whereas the window at the sides, especially towards the ends, is less distorted but clustered. On application of the proposed distortion correction algorithm, curved lines become straight and window which lies in the center of the image acquire lesser image area than in the original image as shown in (Fig. 4b). Thus, the center zooming is reduced but the image area near the image boundary appears to have noise, because sensors do not capture enough information there.



(a)



(b)

Fig. 4. (a) Image captured by a fish eye camera with 180° field of view (b) The final lens distortion corrected image using proposed method.

TOP DOWN VIEW TRANSFORMATION

The cameras installed for driving assistance are tilted at some angle from the vertical axis to capture images of the surroundings. These images suffer from perspec-

tive distortion but are also free from the presence of unnecessary distant objects. These images need to be transformed to a bird's eye view. Lens distortion corrected (LDC) image (calculated in previous subsection) still has effect of camera tilt, to understand it better let us take an example for intuitive understanding on top down view. Let image of a rectangle is captured directly from above (top down view) shown as rectangle ABCD in (Fig. 5). While same rectangle captured with a camera having tilt will appear as a trapezium (A'B'CD).

This intuition is used to develop a generalized method for parameter calculation at various tilts. In other words, consider a simple view of a road or any plain having parallel lines after removal of barrel distortion. These parallel lane markings will appear as if they are converging at some point. The aim of top down view conversion requires these lane markings to appear parallel to each other. To achieve this aim, homography technique is used.

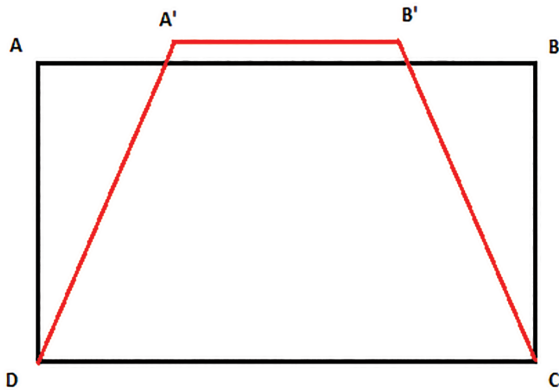


Fig. 5. Effect of camera tilt of object structure.

To apply homography matrix, four reference points are considered in the input image and four points are taken in the required perspective image corresponding to these input points. Thus, eight sets of co-ordinates are available and the homography relation can be applied as follows:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} * \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \quad (7)$$

where, $(x, y, 1)$ are input coordinates and (X, Y, Z) are output co-ordinates. The variables $h_{11}, h_{12}, h_{13}, h_{21}, h_{22}, h_{23}, h_{31}, h_{32}, h_{33}$ are parameters of homography matrix.

In order to convert output co-ordinates to image co-ordinates, X and Y are divided by Z , thus the obtained output points would become $(\frac{X}{Z}, \frac{Y}{Z}, 1)$ and h_{33} will be 1. For eight set of co-ordinates, eight equations

can be represented, giving the value of eight normalized elements of homography matrix, discussed in following subsection.

Calculation of elements of homography matrix

The obtained eight equations can be represented in the form of matrix shown in Eq. 8. (x_i, y_i) are input image co-ordinates and (X_i, Y_i) are output image co-ordinates. With the help of Eq. 8, the elements of homography matrix are calculated. The homography matrix obtained is unique for the chosen set of points. Once, the matrix is obtained, by inverse mapping, corresponding position of pixel in input image is calculated for every output image pixel.

$$\begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \end{bmatrix} = \begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -(X_1 * x_1) & -(X_1 * y_1) \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -(x_1 * Y_1) & -(y_1 * Y_1) \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -(X_2 * x_2) & -(X_2 * y_2) \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -(x_2 * Y_2) & -(y_2 * Y_2) \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -(X_3 * x_3) & -(X_3 * y_3) \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -(x_3 * Y_3) & -(y_3 * Y_3) \\ x_4 & y_4 & 1 & 0 & 0 & 0 & -(X_4 * x_4) & -(X_4 * y_4) \\ 0 & 0 & 0 & x_4 & y_4 & 1 & -(x_4 * Y_4) & -(y_4 * Y_4) \end{bmatrix} * \begin{bmatrix} X_1 \\ Y_1 \\ X_2 \\ Y_2 \\ X_3 \\ Y_3 \\ X_4 \\ Y_4 \end{bmatrix}. \quad (8)$$

The final result is the top-down view of the input image.

$$[O] = [H] * [I]. \quad (9)$$

For inverse mapping, inverse of homography matrix is calculated and that is applied for every output pixel.

$$[H^{-1}] * [O] = [I]. \quad (10)$$

Generalization of homography matrix

As discussed above, to convert the image into top-down view, we need four points in input image and their corresponding projections in output image; *i.e.*, the desired view image. The process is tedious, because for every image a proper set of points should be selected. The selection of output set of points depends entirely on the type of view required, image resolution and tilt angle of camera installed on the vehicle. In the proposed work, selection of points has been generalized, so that it works for every image within a range of camera tilt. The generalization has been done as a function of camera tilt angle t and the algorithm works well with a band of $\pm 8^\circ$ of the actual camera tilt. Once the cameras are installed at the standard angle of 57° , the user may require to adjust the field of view (FOV) for parking. For increasing or decreasing the FOV, user needs to adjust the camera tilt, but as soon as the tilt varies a new set of parameters are required. Storing a set of parameters for every variation in tilt requires lot of memory and computation.

To overcome this issue, a generalized method was developed to automatically calculate the shift in the coordinates of the output image point locations after the tilt is provided.

The central idea behind generalization of points is that any square or rectangle lying in the camera frame appears to be a trapezium, whose slant angle depends on the tilt angle of the camera installed. In addition, size of the square or rectangle, which will appear in the output image, can also be taken as function of camera tilt. Therefore, these four co-ordinates of the trapezium and the tilt angle can be used to predict the x-axis and y-axis location of the four point for required output image. The non-linear regression was used to estimate the shift in output image pixel coordinate with respect to tilt provided.

PREDICTION OF POINTS

A set of input and output coordinates were required initially to form a model to predict the shift in pixel coordinates with respect to change in tilt. Initially a set of images were captured at various angles and their corresponding output image were also captured to generate homography matrix. Now non-linear second order and third order regression technique are used to develop the relation between the shift of pixel's X and Y co-ordinate with respect to camera tilt. In (Fig. 5), the rectangle ABCD represents four points in output image, while trapezium A'B'CD represent the input image points for the homography image. When the tilt is provided to camera, there is obvious shift in the location of points A'B'CD. Individual regression model is developed for each X and Y coordinate of the trapezium corners.

As base of trapezium is symmetric around the center, therefore finding coordinates of one side of the trapezium can be used to calculate coordinates of other side of base. Let initial coordinates of point C and D (Fig. 5) be (c_x, c_y) and (d_x, d_y) respectively and, the shifted pixel coordinates after tilt, C' and D' be (c'_x, c'_y) and (d'_x, d'_y) respectively. Then shifted base position will be given by following equation:

$$c'_x = \alpha_1^x t^2 + \alpha_2^x t + \alpha_3^x, \quad (11)$$

$$c'_y = \alpha_1^y t^2 + \alpha_2^y t + \alpha_3^y, \quad (12)$$

where t represents the tilt in camera from the standard base position for which initial coordinates were located and its coefficients (α_i^x, α_i^y) are the parameters which are estimated using regression technique. Since change in location of base coordinates is not significant, therefore second order regression was able to

predict the location of pixels very accurately. The second point can be located using the shift in first point as shift is symmetric around the x axis.

To calculate remaining two points for perspective transformation we need to calculate horizontal and vertical shift in location of points from their original position, which is calculated using non-linear regression estimation. But for these equations the third order non-linearity was used, because experiments did show that third order equation were able to predict better than second order equations.

$$Height = \beta_1 t^3 + \beta_2 t^2 + \beta_3 t + \beta_4, \quad (13)$$

$$Shift = \gamma_1 t^3 + \gamma_2 t^2 + \gamma_3 t + \gamma_4, \quad (14)$$

where, β_i and γ_i are parameters of regression. The *Height* in Eq. 13 represents the distance by which the line A'B' has moved from its original location, and *Shift* represents the shrink in length of this line segment. For remaining two coordinates of the trapezium, we need to know its slant angle or indirectly speaking, the shift in the x and y coordinate of the trapezium at the top ends with respect to the base points, which can be calculated using Eqs. 13, 14.

Tilt estimation error: We calculate angles of the final rectangle in top down view. Theoretically, a perfect top down view should have a rectangle, but point prediction method introduces an error in final top down view, this error is termed as 'tilt estimation error'.

Table 1 shows the predicted outputs for various tilts. For this experiment the actual angle was taken same as the angle of cameras mounted on vehicle, which was 57° . Then the points were predicted for various tilt angles and in final perspective transform the transformation error was within 5° of the required results and tilt was varied between 49° and 65° .

Table 1. *Perspective transform angles for predicted homography matrix for various angles.*

Camera Tilt (degrees)	Slope of the output line (degrees)	Perspective error (degree)
65	84.80	5.2
63	86.52	3.48
61	87.74	2.26
59	89.46	0.54
57 (actual angle)	89.13	0.87
55	87.57	2.43
53	86.50	3.5
51	85.70	4.3
49	85.26	4.74

Fig. 6 shows the results of top down view conversion using the developed technique. (Fig. 6c) shows the results of predictive generalization technique for image captured at 42° shown in (Fig. 6a) and (Fig. 6d) show the results for the image (Fig. 6b), which was used to find the point to calculate the parameters of homography matrix.

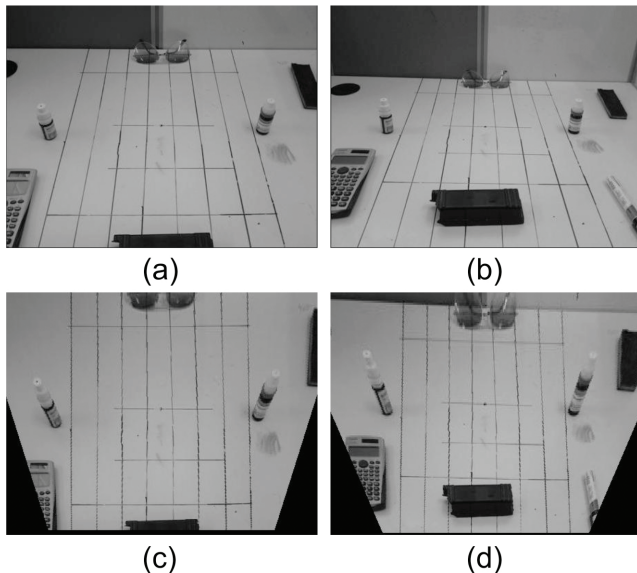


Fig. 6. Results of the predictive generalization of homography matrix for various tilt angles. (a) Image captured with camera tilt angle of 42° (b) Image captured with camera tilt angle of 57° for which actual homography parameters were calculated (c) Top down view using point prediction method for 42° (d) actual top down view conversion using homography matrix at 57° .

RESULTS

Implementation flow of the complete algorithm from fish eye lens distortion correction to top down view conversion is shown in (Fig. 7). Proposed technique involves computation only for generating inverse mapping matrix. Once the inverse mapping matrixes are collected for all the sub-processes (section-II), they can be used to map pixels from raw input to final output without any computation. To explain it better, let's take a detailed look at the flow chart in (Fig. 7).

The input image to the system is a fish eye image captured from four fish eye cameras, mounted on four sides of the vehicle. The input images taken from fish eye cameras with field of view of 180° are shown in (Fig. 8). These are four images from each side of the vehicle. The Figs. 8a and 8b are the images captured from cameras mounted on front and back side, while

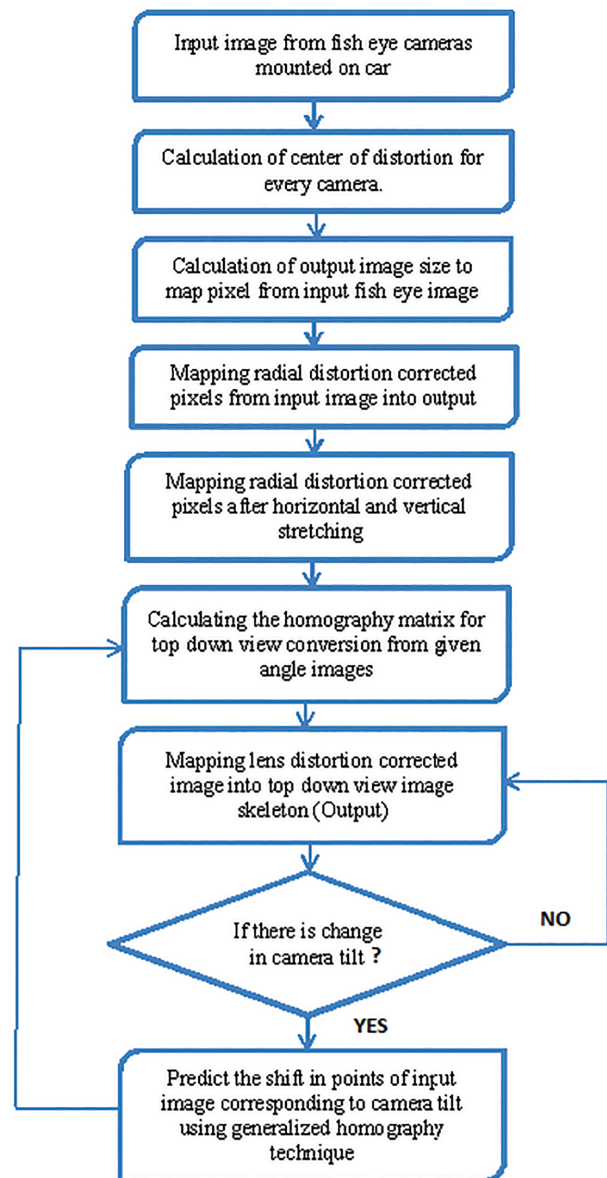


Fig. 7. Algorithm flow chart from input image to top down view conversion.

(Fig. 8c) and (Fig. 8d) are the images taken by cameras mounted below right and left rear view mirrors. It is clear from the images that the body of the vehicle (on which cameras are mounted) looks curved. Each image passes independently into the system. The COD for each camera has been calculated prior to mounting using given pattern shown in (Fig. 2a). Distortion radius is calculated with respect to this COD. Reverse mapping matrix is calculated using Eqs. 1–4. Pixels from input image are mapped directly into the radial corrected image skeleton once LUT is known. It is important to know that LUT is specific to the given setup. Same LUT can be used for all the vehicles of same model but radial distor-

tion correction matrix has to be calculated for every new vehicle model, which will have different dimension. The final results are shown in (Fig. 9) for the input shown in (Fig. 8). There are four outputs, (Fig. 9), with respect to four inputs, (Fig. 8).

5th step of the flow chart converts radial distortion corrected image into stretched image, for which a LUT is developed and it is used to map pixels from input to output. In 6th and 7th step of flow chart, the homography matrix is calculated and corresponding LUT is stored to map input image to required perspective (top down view). The last step is automatic generation of homography matrix for various camera tilts, but it will not be applicable once the camera is set by driver in a particular angle.

Three LUT are required to generate final output image from input image. This map can be used to make a signal flow path for each input pixel on a FPGA and output image will be delivered in real time on the vehicle central display, which makes it feasible to use camera frame rate of 30 fps.

The final results from the camera mounted on the front of the vehicle are shown in (Fig. 10). It is clear from (Fig. 10a), there is very high fish eye distortion in the image (all the lines in image look curved). (Fig. 10b) shows the lens distortion corrected image, which looks like an image taken by a normal flat lens camera. The final image in (Fig. 10c) is the top down view.

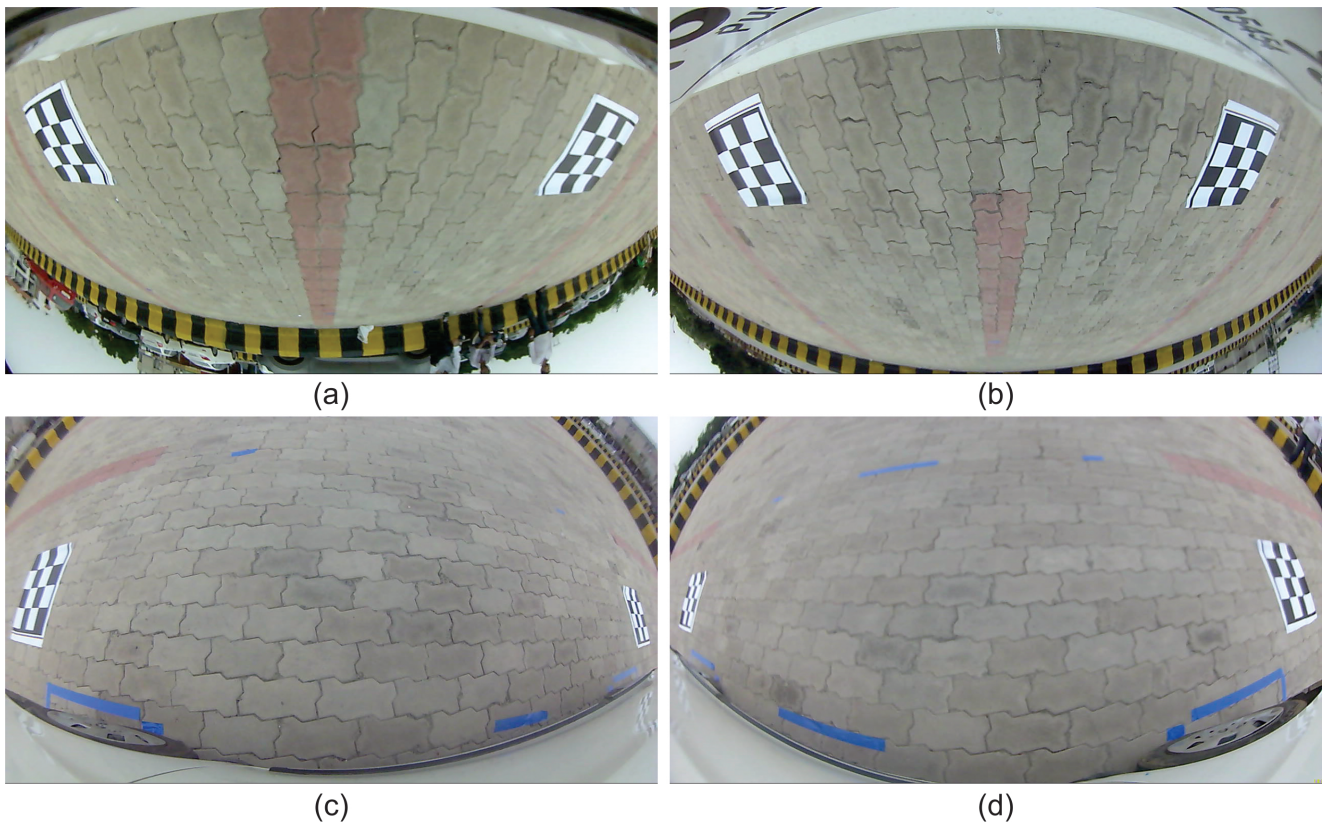


Fig. 8. Images taken by fish eye cameras mounted on car (a) image taken by front camera (b) Image taken by back camera (c) Image taken by right side camera (d) image taken by left side camera.

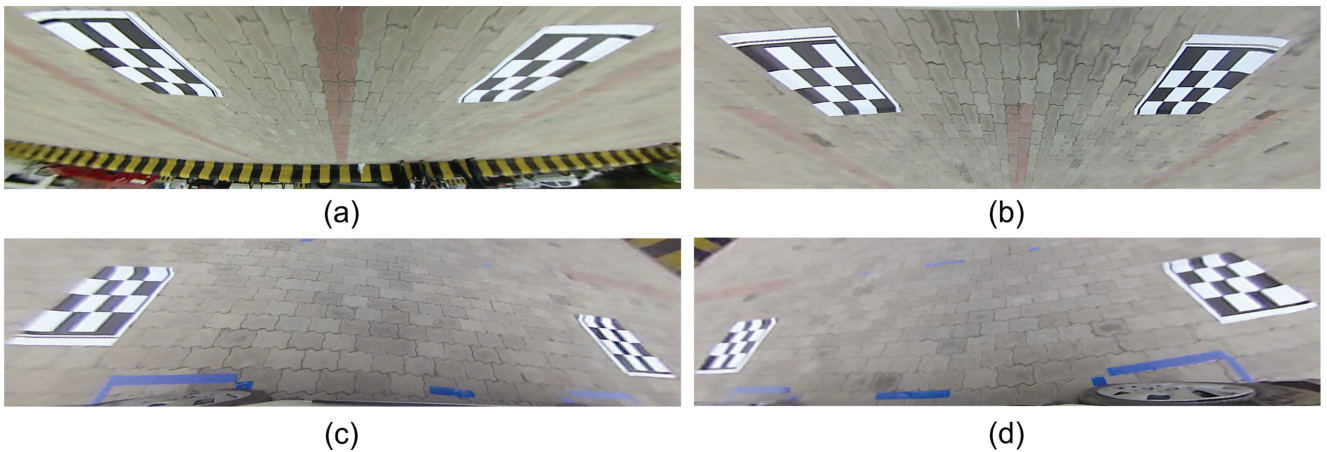


Fig. 9. Output images produced by proposed algorithm for the set of input fish eye images (a) Output image for front camera input (b) Output image for back camera input (c) Output image for right camera input (d) Output image for left camera input.

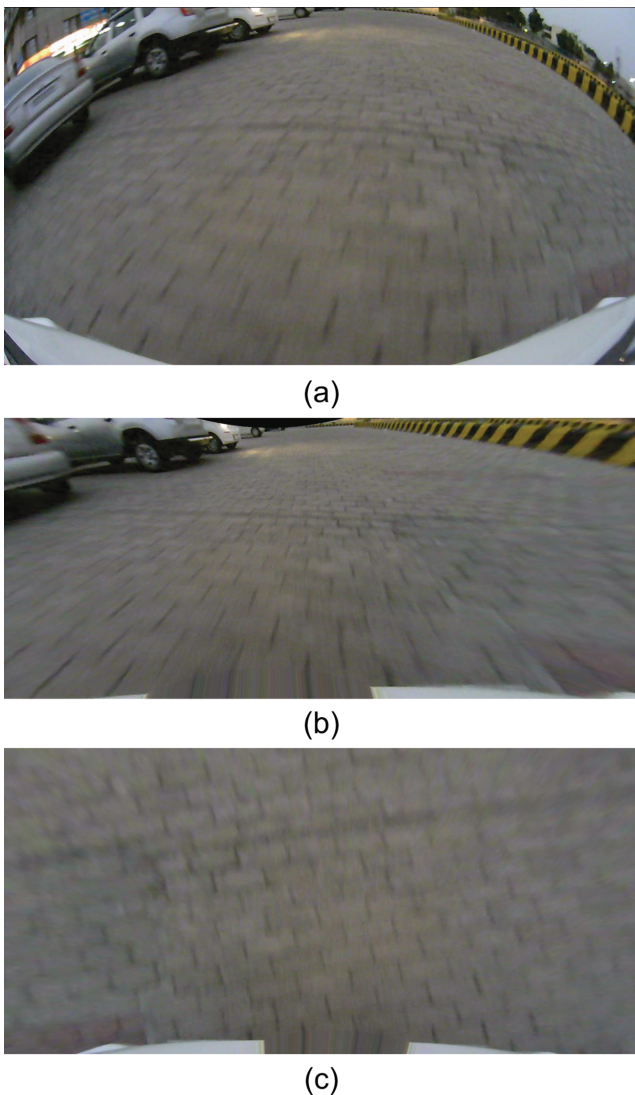


Fig. 10. (a) Image from the fish eye camera mounted on front of the vehicle (b) Image after lens distortion correction (c) The final output image as top down view of the input image.

DISCUSSION

The present manuscript presents a novel method for generating top-down view of the frames captured by cameras with ultra-fish eye lenses, which are mounted on all four sides of the vehicle. The COD of the fish eye captured images or frames is estimated and then passed through the lens distortion correction algorithm to obtain distortion free image. The images captured at an angle from horizontal are converted into top-down views using the concept of homography. Later, homography matrix is generalized for a finite range of resolution of the input image. All above-mentioned steps were successfully applied and tested on experimental data. The proposed method produces satisfactory results on random images independent of camera tilt angle. The generalization technique developed produces best results in tolerance band of 8° (of actual camera tilt) as shown in Table 1. Table 1 presents the relation between camera tilt and slope of one of the vertical lines observed in (Fig. 6b). The highest perspective error in the given angle variation is 5.2° as shown in table 1. The future scope of the research remains in development of the next stage of surround view system, where stitching (Peer *et al.*, 2002) of these images to produce complete 360° surroundings of vehicle in a single image. There is also scope in development of lens distortion correction technique, where loss of information still can be reduced and perspective transformation method can be improved to make it generalized for any image resolution and tilt angles.

REFERENCES

- Bangadkar S, Dhane P, Nair S, Kutty K (2011). Mapping matrix for perspective correction from fish eye distorted images. IEEE-International Conference on Recent Trends in Information Technology, ICRTIT 2011 MIT, Anna University, Chennai. June 3-5.
- Brito JH, Angst R, Koser K, Pollefeys M (2013). Radial distortion self-calibration. IEEE Conference on Computer Vision and Pattern Recognition, pp. 1368–75.
- Burchardt CB, Voss K (2001). A new algorithm to correct fish-eye- and strong wide-angle-lens-distortion from single images. Image Processing, 2001. Proceedings. International Conference on In Image Processing, 2001. Vol. 1 (2001), pp. 225–8.
- Dhane P, Kutty K and Bangadkar S (2012). A generic non-linear method for fisheye correction. Int J of Comp Appl (0975-8887) 51:10, August.
- Hartley R and Kang SB (2007). Parameter-Free Radial Distortion Correction with Center of Distortion Estimation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 29;8.
- Kedzierski M, Fryskowska A (2008). Precise Method Of Fisheye Lens Calibration. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. XXXVII. Part B5. Beijing.
- Lin CC and Wang MS (2012). A vision based top-view transformation model for a vehicle parking assistant. Sensors 12:4431–46.
- Li H and Hartley R (2005). A Non-iterative Method for Correcting Lens Distortion from Nine Point Correspondences. In Proc. OmniVision'05, ICCV-workshop.
- Li W, Qu F, Wang Y, Liu P (2014). A Fast Vehicle Top-View System. 2nd International Conference on Systems and Informatics.
- Melo R, Antunes M, Barreto JP, Falcao G, Goncalves N (2013). Unsupervised Intrinsic Calibration from a Single Frame Using a. IEEE International Conference on Computer Vision, pp. 537–44.
- Peer P, Solina F (2002). Panoramic depth imaging: Single standard camera approach. Int J of Comp Vision, 47: 149–60.
- Santhanam SM, Balisavira V, Roh SH, Pandey VK (2014). Lens Distortion Correction and Geometrical Alignment for Around View Monitoring System. In Consumer Electronics (ISCE 2014), The 18th IEEE International Symposium on 2014 Jun 22 (pp. 1–2).
- Schnabel R, Wahl R, Klein R (2007). Efficient RANSAC for point - cloud shape detection. In Computer graphics forum, Blackwell Publishing Ltd 26;2:214–26).
- Yu M and Ma G (2014). 360° Surround View System with Parking Guidance. SAE International, 2014-01-0157.
- Zhang Q and Kamata SI (2015). Fisheye Image Correction Based on Straight-line Detection and Preservation. IEEE International Conference on Systems, Man, and Cybernetics.
- Zhang B, Appia V, Pekkucuksen I, Batur AU, Shastry P, Liu S, Sivasankaran S, Chitnis K and Liu Y (2014). A surround view camera solution for embedded systems. IEEE Conference on Computer Vision and Pattern Recognition.