2014-01-0157
Published 04/01/2014

doi:10.4271/2014-01-0157
saecomveh.saejournals.org

360° Surround View System with Parking Guidance

Mengmeng Yu and Guanglin Ma
Delphi Automotive

ABSTRACT

In this paper, we present a real-time 360 degree surround system with parking aid feature, which is a very convenient parking and blind spot aid system. In the proposed system, there are four fisheye cameras mounted around a vehicle to cover the whole surrounding area. After correcting the distortion of four fisheye images and registering all images on a planar surface, a flexible stitching method was developed to smooth the seam of adjacent images away to generate a high-quality result. In the post-process step, a unique brightness balance algorithm was proposed to compensate the exposure difference as the images are not captured with the same exposure condition. In addition, a unique parking guidance feature is applied on the surround view scene by utilizing steering wheel angle information as well as vehicle speed information.

CITATION: Yu, M. and Ma, G., "360° Surround View System with Parking Guidance," *SAE Int. J. Commer. Veh.* 7(1):2014, doi:10.4271/2014-01-0157.

INTRODUCTION

Advanced Driver Assist Systems become more and more hot spot in the intelligent transportation system (ITS) research area. Major ITS research topics were focused on the road safety as well as traffic management. In recent years, with government regulation of backup aid systems and also with the market growing of emerging countries where major drivers are inexperienced new drivers, parking aid applications have also become a hot research area.

The 360 degree vehicle surround view monitoring system is a major research area in parking aid applications and it is already commercialised in many major car makers. For example, vehicle surrounding monitoring systems using multi-view cameras have been widely used in intelligent transport systems, such as the track driving assistant system [1] and the bird's eye view vision system [2]. In the track driving assistant system [1], four catadioptric cameras mounted at four corners of a truck are used to capture the surrounding roadway images. The bird's-eye view vision system uses six fisheye cameras to avoid the occlusion caused by the vehicle itself [2].

In this paper, three key issues are solved to improve the performance of the surround view monitoring system. First, we combine two look-up tables into one to reduce computational cost into half.

Second, after correcting the distortion of four fisheye images and registering all images on a planar surface, the four single images need to be stitched. There are several approaches, such as graph cut [3, 4] and dynamic programming [5, 6, 7]. However, most of the method required highly computational and memory costs. In our work, a flexible stitching method was developed to smooth the seam of adjacent images away to generate a high-quality result.

Third, in the post-process step, since the four images were taken at different environment with different cameras, a color and brightness compensation approach for constructing panoramic images is needed. Pham and Pringle [8] utilized polynomial mapping function to perform color correction. Ha et al. [9] perform linear correction and operate in the YCbCr color space. Matthew Brown and David G. Lowe[10] use a multiband blending scheme ensures smooth transitions between images and preserve high frequency details. In our work, given the real-time running and low cost requirement, we are developing a fast and efficient color and brightness compensation approach for surround view image stitching.

Although the surround view monitoring system is a very useful system for back up aid and parking assist, inexperienced drivers still have difficulty doing parallel or vertical parking even with the help of the surround view system. Based on this fact, in addition to the surround view monitoring system, a parking guidance algorithm was developed. The algorithm can help the driver to do safe and convenience convenient vertical and parallel parking in any parking spot visible within the surround view scene. Furthermore, the algorithm can also help the driver to identify whether the available parking area is sufficient for the host vehicle to park in or not.

SURROUND VIEW SYSTEM

Figure 1 illustrates the general algorithm flowchart of our surround view system. Since we use four fish-eye cameras to implement this system, the first step should be correcting the distortion of the wide-angle images. Then those images need to be projected into one planar, which is a top-down view planar. After projection, all four images need to be arranged into their corresponding places. The four images have not been combined into one. Therefore, the next step is stitching them into one top-down view image. Finally, some post-processing like seamless interpolation and brightness compensation are used to perfect our result. In general, the algorithm will create two look-up tables (LUT) at the fisheye image un-distortion step and projection transformation step respectively. We merge the two look-up tables together into one, using one table to realize original image to top-down view image mapping. In this way, the running time of embedded system is halved.

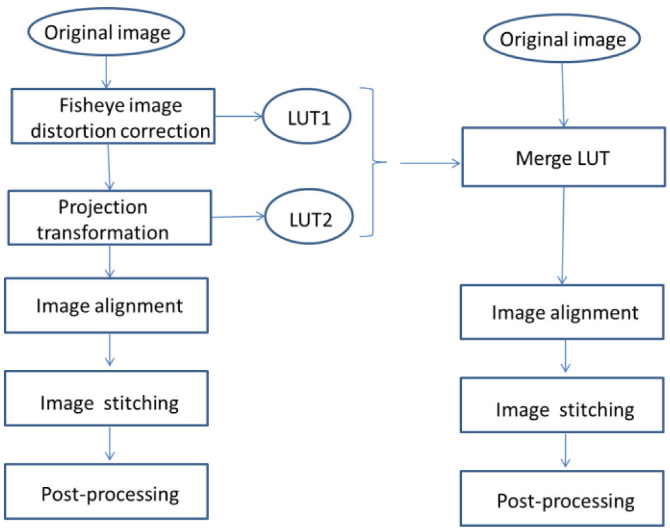
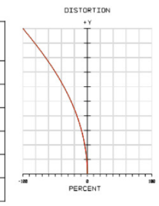


Figure 1. General algorithm flowchart

In our system, we are using 180 degree fisheye cameras. The radial distortion introduced by fisheye lenses is corrected by fisheye lens distortion curve provided by lens supplier (Figure 2). In order to speed up the calculation time, a look up table is created in advance. Please note that the table below is only part of the LUT table.

When we create a fisheye un-distortion loop-up table, we create a 2D interpolation based look-up table. In this way, we can successfully remove the "water ripple" which appears due to image transformation. (Figure 3)

Field Of View Angle (degree)	Real Image Height (mm)	Ref. Image Height (mm)	Distortion (%)
0	0	0	0
0.9	0.02338706	0.02338941	-0.01004311
1.8	0.04677156	0.04679036	-0.0401729
2.7	0.07015096	0.07021443	-0.09039081
3.6	0.0935227	0.09367323	-0.16069923
4.5	0.11688422	0.11717845	-0.25110146



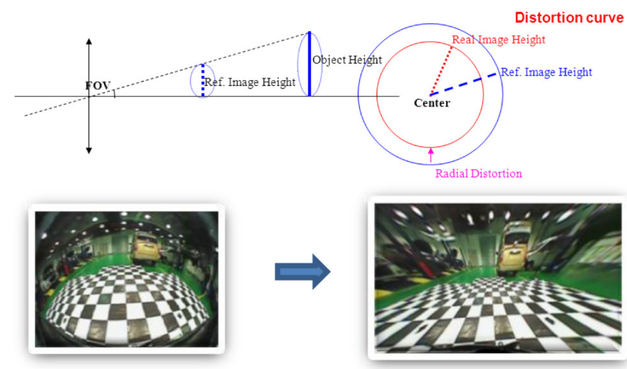


Figure 2. Distortion correction look-up table

The linear interpolation algorithm uses source image intensities at the four pixels:

$$(x_{s-1}, y_{s-1}), \ (x_{s+1}, y_{s+1}), \ (x_{s-1}, y_{s+1}), \ (x_{s+1}, y_{s-1}),$$

Which are closest to (x_s, y_s) in the source image:

$$x_{s-1} = int(x_s), x_{s+1} = x_{s-1} + 1, y_{s-1} = int(y_s)\ , y_{s+1} = y_{s-1} + 1 \quad (1)$$

First, the intensity values are interpolated along the x-axis to produce two intermediate results I_0 and I_1 (see Figure 3. Linear Interpolation):

$$I_0 = S(x_s, y_{s-1}) = S(x_{s-1}, y_{s-1}) * (x_{s+1} - x_s) + S(x_{s+1}, y_{s-1}) * (x_s - x_{s-1})$$

$$I_1 = S(x_s, y_{s+1}) = S(x_{s-1}, y_{s+1}) * (x_{s+1} - x_s) + S(x_{s+1}, y_{s+1}) * (x_s - x_{s-1}) \quad (2)$$

Then, the sought-for intensity $D(x_D, y_D)$ is computed by interpolating the intermediate values I0 and I1 along the y-axis:

$$D(x_D, y_D) = I_0 * (y_{s+1} - y_s) + I_s * (y_s - y_{s-1}) \quad (3)$$

Figure 3. Interpolation based LUK for "water ripple" removal

According to the projection transformation algorithm, we can get a top-down view projection look-up table. By specifying the feature points in a reference image (X_d, Y_d) of the ground plane and their corresponding points in image (X_r, Y_r), we can establish a holography matrix and get a top-down view using the projection transformation. The detailed projection matrix is illustrated in equation 4 and equation 5. (Figure 4)

$$\begin{pmatrix} x_d \\ y_d \\ z_d \end{pmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} * \begin{pmatrix} x_r \\ y_r \\ z_r \end{pmatrix}, we\ set\ a_{33} = 1 \tag{4}$$

$$\begin{pmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{21} \\ a_{22} \\ a_{23} \\ a_{31} \\ a_{32} \end{pmatrix} = \begin{pmatrix} X_{r,1} & Y_{r,1} & 1 & 0 & 0 & 0 & -X_{r,1}*X_{d,1} & -Y_{r,1}*Y_{d,1} \\ 0 & 0 & 0 & X_{r,1} & Y_{r,1} & 1 & -X_{r,1}*X_{d,1} & -Y_{r,1}*Y_{d,1} \\ X_{r,2} & Y_{r,2} & 1 & 0 & 0 & 0 & -X_{r,2}*X_{d,2} & -Y_{r,2}*Y_{d,2} \\ 0 & 0 & 0 & X_{r,2} & Y_{r,2} & 1 & -X_{r,2}*X_{d,2} & -Y_{r,2}*Y_{d,2} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ X_{r,n} & Y_{r,n} & 1 & 0 & 0 & 0 & -X_{r,n}*X_{d,n} & -Y_{r,n}*Y_{d,n} \\ 0 & 0 & 0 & X_{r,n} & Y_{r,n} & 1 & -X_{r,n}*X_{d,n} & -Y_{r,n}*Y_{d,n} \end{pmatrix} \begin{pmatrix} X_{d,1} \\ Y_{d,1} \\ X_{d,2} \\ Y_{d,2} \\ \cdots \\ X_{d,n} \\ Y_{d,n} \end{pmatrix} \tag{5}$$

Figure 4. Project_LUT Transformation

In our work, we combine above two look-up tables into one. In this way, we can directly project the source fisheye image to top-down view image. This process reduces computational cost into half.

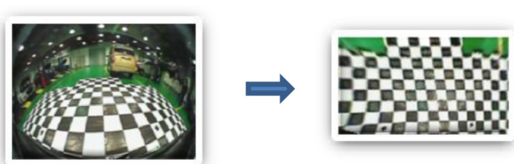


Figure 5. Merge into one projection LUT

Seamless Image Stitching

The system requires stitch together the images which are captured by four fisheye cameras around the vehicle. After correcting the distortion of four fisheye images and registering all images on a planar surface, we want to smooth the seam of adjacent images away to generate a high-quality composite result. In order to make the fused image with visual consistency to where there are no obvious seams, we use the image fusion method. This method is based on pixel value weighted average smooth transition, which is simple and easy to achieve, and can basically meet the general application requirements.

Set image $I_1(x, y)$ and image $I_2(x, y)$ are two adjacent images which need to do seamless image stitching, (x, y) is the coordinate of the pixel. The fused image pixel value is:

$$I(x,y) = \begin{cases} I_1(x,y) & (x,y) \in I_1\ and (x,y) \notin I_2 \\ I_1(x,y) * \alpha_{i,j} + I_2(x,y) * (1-\alpha_{i,j}) & (x,y) \in I_1\ and (x,y) \in I_2 \\ I_2(x,y) & (x,y) \notin I_1\ and (x,y) \in I_2 \end{cases} \tag{6}$$

Where α is the weight factor, which can be calculated as:

$$\alpha_{x,y} = k\frac{\max(r-x,x) * \max(c-y,y)}{rc} \tag{7}$$

Where r and c represent the total number of pixels which is the length and width of the overlap region of the two input images contains. k is the gain, we set it to 0.1.*max* (x, y) compares the input x and y, output both relatively bigger one.

Figure 6 shows the result of seamless stitching. The left image is the seam stitching result. The middle is the weighted image, the right one is the stitching image.

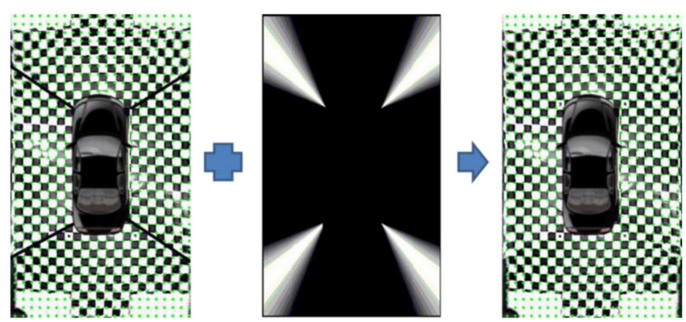


Figure 6. Seamless stitching based on weighted average

Brightness Balance

In the surround view system, four images are captured at different conditions and need to be combined into one image, which creates a challenge issue that balances exposure distinction of each image. We can use the gain compensation algorithm to minimize global intensity differences.

For an image sequence S_1, S_2, S_3, S_4, suppose S_{i-1} and S_i are two adjacent images. In the overlapping area between the two adjacent images, we compute a correction coefficient for image S_i as [11]:

$$\alpha_{c,j} = \frac{\sum_p (C_{c,i-1}(p))^r}{\sum_p (C_{c,i}(p))^r} \quad c \in \{R,G,B\}\ \ i = 1,2,3,4 \tag{8}$$

Where $C_{c,j}(p)$ is the gamma-corrected color value of pixel p in color channel c. the gain compensation coefficient for the first image is set to one. $\overline{I_j}$ is the average color value of image j.

We calculate the overall gain parameters for the images by minimizing a global error function, which calculates the sum of intensity differences of all the overlapping pixels.

To avoid cumulative errors, we want to adjust the color for each image as little as possible to avoid image saturation, and create the following objective function[11]::

$$\min_{g_c} \sum_i^n (g_c \alpha_{c,j} - 1)^2 \qquad c \in \{R, G, B\} \tag{9}$$

g_c is a global compensation coefficient for color channel *c*. *n* is 4. In this approach, the images can be adjusted into similar exposure.

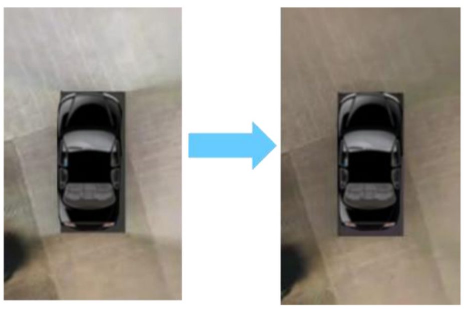


Figure 7. Brightness balance

Parking Guidance for Surround View System

The 360°Surround View System with Parking Guidance system has two different parking modes: parallel parking mode and vertical parking mode. By using pre-calibrated camera and vehicle data, the system calculates intended vehicle direction and position in real-time for each movement of steering wheel. Each parking mode has several parking phases. At each parking phase, different guide lines are drawn on the corresponding position of image frame to tell the driver what he should do next step. By following the visual instruction of our system, a driver can complete a parking easily.

Parallel parking guidance mode helps the driver park the host vehicle in a parallel parking space. The main process is:

Step1. Parking Mode Selection

Once the parallel parking mode is selected, there will be a polygon and a flag around the target parking space on the screen, please refer to Figure 8 (a) - Figure 8 (b).

The polygon is used for the three following purposes: First, identify the size of the target parking space. The size of the polygon is the minimum size required for parking in our PGS system. If the size of the target parking space is smaller than that of the polygon, it means that the host vehicle cannot be parked into this space. Second, identify whether the host vehicle will crash with other vehicles (vehicles in front and rear) and objects when it is parking. If any part of the object is located inside of the polygon, the host vehicle is going to crash with the object and the driver has to adjust the start position of host vehicle. Third, identify whether the host vehicle is parallel to the target parking space. The flag located at inner corner of the polygon is used to help the driver to identify position of the polygon more easily.

Step2. Go to Start Parking Position

Driver drives the car forward and stops the host vehicle blue arc line beside the rear wheel reaches to the rear wheel of the vehicle at the front of destination parking spot. Figure 8(c) - Figure 8(d).

Step3. 1st Steering Wheel Turning

Turn steering wheel to right side until the steering icon turns to green color. Figure 8(e) -Figure 8(f).

Step4. Back up

Hold the steering wheel and back up the host vehicle until the yellow guide arc on the screen align to the roadside, please refer to Figure 8 (g) - Figure 8 (h).

Step5. 2nd Steering Wheel Turning

Stop the car, and turn steering wheel reversely until the steering wheel icon turns to green again, the yellow guide arc disappears, and the static and dynamic overlay appeared. Continue back-up and stop when the static overlay is parallel to the rear vehicle, please refer to Figure 8 (i).

Vertical parking guidance mode helps the driver park the host vehicle in an vertical parking space. The main process is:

Step1. Parking Mode Selection

Once the vertical parking mode is selected, there will be a rectangle appearing on the parking path on the screen, please refer to Figure 9 (a) - Figure 9 (b).

Step2. 1st Steering Wheel Turning

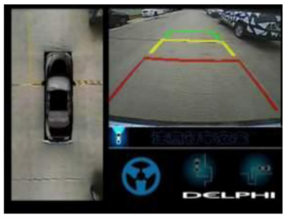
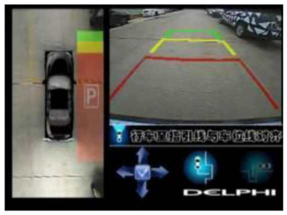
Turn steering wheel left side until the steering icon turns to green color, please refer to Figure 9 (c) - Figure 9(d).

Step3. Go Forward

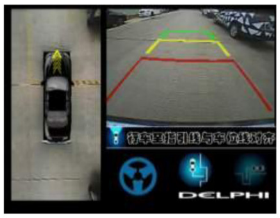
Driver drives the car forward and stops the host vehicle red arc line behind host vehicle match destination parking spot, Please refer to Figure 9 (e).

Step4. 2nd Steering Wheel Turning

Turn steering wheel right side until the steering icon turns to green color and the static and dynamic overlay appeared on the screen (Figure 9 (f)). Hold the steering wheel and reverse the host vehicle carefully (Figure 9 (g)). Continue back-up and stop when the static overlay parallel to the side vehicle (Figure 9 (h)).

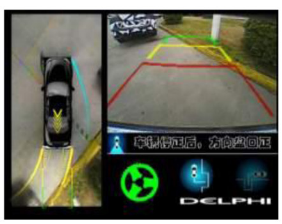



(a) Go to parking spot (b) Identify parking spot




(c) Drive forward (d) Stop at first turn point

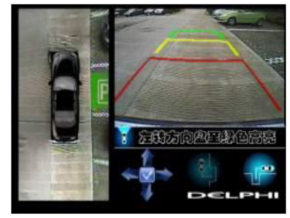

(e)Turn steering wheel to right (f) steering icon turns green

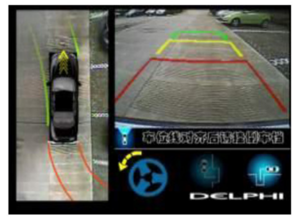
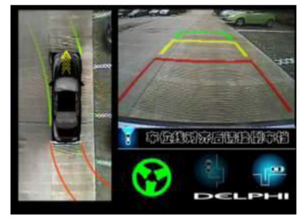
(g) Back up and stop reversely (h) Turn steering

(i) Back up and Stop the vehicle

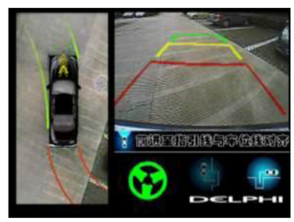
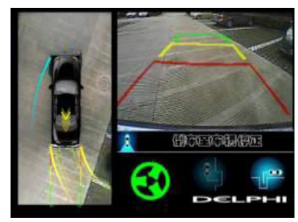
Figure 8. Parallel Parking Steps

(a) Go to parking spot (b) Identify the parking spot

(c) 1st wheel turning (d) Steering icon turns green

(e) Go forward until the red line (f) 2nd wheel turing


(g) Back up (h) Stop the vehicle

Figure 9. Vertical Parking Steps

CONCLUSIONS

In this paper, we have presented a 360 degree surround view system which can provide the top-down view image of vehicle surrounding and parking guidance system based on it.

Experiments have been carried out in a closed-loop environment by applying the proposed algorithm on a set of test vehicles. The experimental results indicate promising performance of our parking guidance featured 360 degree Surround View System with no collision hazard to host vehicle or to nearby vehicles.

The described algorithm has been integrated in a vision based system by Delphi.

REFERENCES

1. Ehlgen T., Pajdla T., and Ammon D., “Eliminating blind spots for assisted driving”. IEEE Trans. on Intelligent Transportation Systems, Vol. 9(4), pp. 657-665, 2008.
2. Liu Y.-c., Lin K.-Y., and Chen Y.-S., “Bird's-eye view vision system for vehicle surrounding monitoring”. Robot Vision, pp. 207-218. 2008.
3. Xiong Y. and Pulli K., “Gradient domain image blending and implementation on mobile devices”. In MobiCase, 2009.
4. Gracias N., Mahoor M., Negahdaripour S., and Gleason A.. “Fast image blending using watersheds and graph cuts”, Image Vision Comput., 2009.
5. Xiong Y. and Pulli K., “Fast image labeling for creating high resolution panoramic images on mobile devices”, In IEEE Int. Symp. on Multimedia, 2009.
6. Ha S. J., Koo H., Lee S. H., Cho N. I., and Kim S. K., “Panorama mosaic optimization for mobile camera systems. Consumer Electronics”, IEEE Transactions on, 2007.
7. Ha S. J., Lee S. H., Cho N. I., Kim S. K., and Son B., “Embedded panoramic mosaic system using auto-shot interface”, Consumer Electronics, IEEE Transactions on, 2008.
8. Pham B. and Pringle G., “Color correction for an image sequence”, Computer Graphics and Applications, IEEE, pp. 38-42, 1995.
9. Ha Seong Jong, Koo Hyung, Lee Sang Hwa, Cho Nam Ik, and Kim Soo Kyun, “Panorama mosaic optimization for mobile camera systems”, Consumer Electronics, IEEE Transactions on, 2007.
10. Brown Matthew and Lowe David G., “Automatic panoramic image stitching using invariant features”, International Journal of Computer Vision, pp. 59-73, 2007.
11. Xiong Yingen and Pulli Kari, “Color Correction for Mobile Panorama Imaging”, Proceedings of the First International Conference on Internet Multimedia Computing and Service, pp. 219-226, 2009.