

# 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.

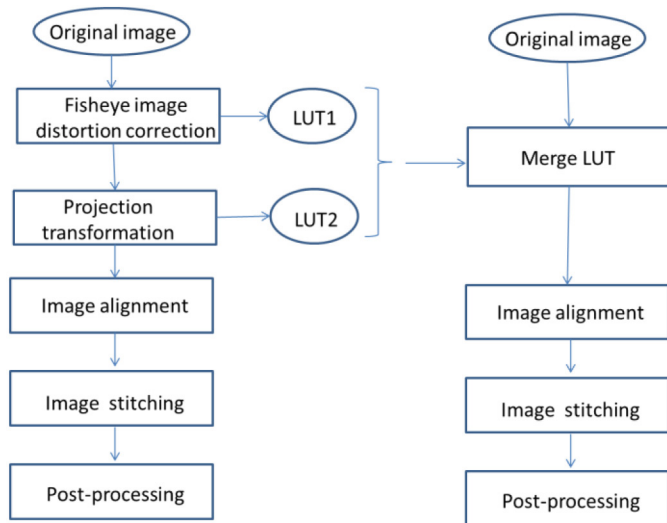


Figure 1. General algorithm flowchart

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)

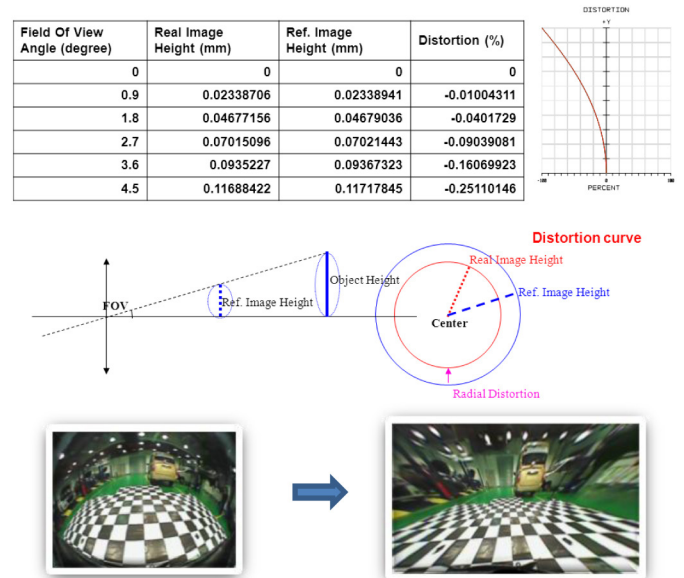


Figure 2. Distortion correction look-up table

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:

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

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  $I_0$  and  $I_1$  along the y-axis:

$$D(x_D, y_D) = I_0 * (y_{s+1} - y_s) + I_1 * (y_s - y_{s-1}) \quad (3)$$

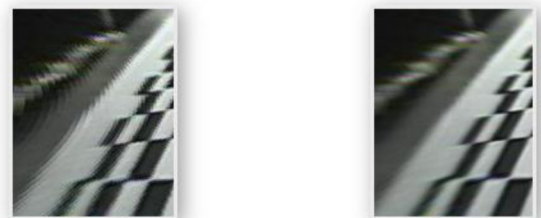


Figure 3. Interpolation based LUK for “water ripple” removal