A surround view camera solution for embedded systems

Buyue Zhang, Vikram Appia, Ibrahim Pekkucuksen,
Aziz Umit Batur, Pavan Shastry, Stanley Liu,
Shiju Sivasankaran, Kedar Chitnis
Embedded Processing, Texas Instruments
Dallas, TX 75243

Yucheng Liu
School of ECE
Purdue University
West Lafayette, IN 47906

***Abstract*—Automotive surround view camera system is an emerging automotive ADAS (Advanced Driver Assistance System) technology that assists the driver in parking the vehicle safely by allowing him/her to see a top-down view of the 360° surroundings of the vehicle. Such a system normally consists of four to six wide-angle (fish-eye lens) cameras mounted around the vehicle, each facing a different direction. From these camera inputs, a composite bird-eye view of the vehicle is synthesized and shown to the driver in real-time during parking. In this paper, we present a surround view camera solution that consists of three key algorithm components: geometric alignment, photometric alignment, and composite view synthesis. Our solution produces a seamlessly stitched bird-eye view of the vehicle from four cameras. It runs real-time on DSP C66x producing an 880×1080 output video at 30 fps.**

***Keywords*-surround view cameras; around view camera systems; multi-camera fusion; ADAS; geometric alignment; photometric alignment; composite view synthesis;**

I. Introduction

Figure 1. Illustration of a surround view camera system and the fish-eye image captured by each camera. Four fish-eye cameras with 180° FOV are mounted around the vehicle, each facing a different direction.

Automotive surround view cameras, also called around view cameras or surround vision monitoring system, is an emerging automotive ADAS technology that provides the driver a 360° surrounding of the vehicle [1], [2], [3], [4], [5]. Such systems normally consist of four to six fish-eye cameras mounted around the vehicle, for example, one at the front bumper, another at the rear bumper, and one under each side mirror. Figure 1 illustrates a surround view camera system and the captured fish-eye images. Most commercial surround view solution today suffer artifacts from geometric misalignment at stitching boundaries, and/or brightness and color inconsistency among different camera input due to lack of photometric correction [4]. As a result, the composite surround view looks unnatural. On the other hand, there are solutions in literature which claim seamless stitching [2], but are not designed or tested on an embedded platform to achieve real-time performance and quality.

In this paper, we describe a novel surround view camera solution that produces a seamlessly stitched bird-eye view. Our multi-algorithm surround view solution is designed for embedded systems. It has been implemented on DSP C66x and produces high definition (HD) output video at 30 fps. Moreover, our solution includes automatic calibration (geometric alignment) that does not require camera extrinsic parameters, but solely relies on a specially designed calibration chart, making re-calibration of the system efficient.

The organization of the paper is as follows: in Sec. II, we describe the three key algorithm components of our surround view solution. In Sec. III, we present the architecture of surround view solution, and its optimization on DSP C66x. Finally, results and performance of the proposed solution are given in Sec. IV.

II. Surround View Camera Solution

Our surround view camera solution consists of three key algorithm components: geometric alignment, photometric alignment, and composite view synthesis. Geometric alignment corrects fish eye distortion from the input videos and converts each input video frame from its respective perspective to a common bird-eye perspective. Photometric alignment corrects the brightness and color mismatch between adjacent views to achieve seamless stitching. Finally,

the synthesis algorithm generates the composite surround view after geometric and photometric correction.

A. Geometric Alignment Algorithm

Geometric alignment, also called calibration, is an essential component of the surround view camera system. This step includes both fish-eye lens distortion correction (LDC) and perspective transformation. For fish-eye distortion correction, we use a radial distortion model and removes fish-eye affect in original input frames by applying the inverse transformation of the radial distortion function. After LDC, we simultaneously estimate four perspective transformation matrices, one for each camera, to transform four input LDC corrected frames so that all input views are properly registered with the ground plane. We assume that the world is a 2D flat surface. Our algorithm is a calibration-chart-based approach. The content of the chart is designed to facilitate the algorithm in accurately and reliably finding and matching features. One particular chart design is shown in Fig. 2.

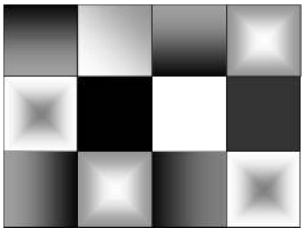

Figure 2. An example geometric calibration chart

During calibration of the surround view cameras, four calibration charts are placed around the vehicle. Each chart should be placed in the common FOV of two adjacent cameras, i.e., every pair of the adjacent cameras should "see" one common chart. After that, a frame from each camera is captured simultaneously.

The first step of the algorithm is to apply LDC correction to each frame. Next, we perform initial perspective transformation to each LDC corrected frame. The parameters for the initial transformation can be obtained from camera placement specifications or estimated from the frame content itself. We used the latter approach. Next, Harris corner detection [6] is run in the image data in the overlapping area of adjacent views to find regions of interest. We filter raw Harris corner data to locate the strongest corners and then calculate BRIEF descriptor [7] of each corner feature to match corners from two cameras using BRIEF scores. The next step is to find the optimal perspective matrix for each frame that minimizes the distances between matched features. And finally we create a look-up-table (LUT) to encode both LDC and perspective transformation information. After the geometric LUT is obtained, it is saved to memory and used during composite view synthesis to create the final surround view output.

B. Photometric Alignment Algorithm

Due to different scene illumination, camera Auto Exposure (AE), and Auto White Balance (AWB), the color and brightness of the same object captured by different cameras can be quite different. As a result, the stitched composite image can have noticeable photometric difference between two adjacent views (i.e., camera input). The goal of photometric alignment for a surround view system is to match the overall brightness and color of different views such that the composite view appears as if it were taken by a single camera placed above the vehicle. To achieve that we design a global color and brightness correction function for each view such that the discrepancies in the overlapping regions of adjacent views are minimized.

Assuming proper geometric alignment is already applied to the input frames, the composite surround view is shown in Fig. 3. The composite surround view consists of data from all four input frames, view 1, 2, 3, and 4. The overlapping regions are the portion of the frames that come from the same physical world but are captured by two adjacent cameras, i.e., $O_{m,n}$, $m = 1, 2, 3, 4$, and $n \equiv (m+1) \bmod 4$. $O_{m,n}$ refers to the overlapping region between view m and view n, and n is the neighboring view of view m in clockwise order. At each location in $O_{m,n}$, there are two pixels available, i.e., the image data from view m and its spatial counter-part from view n. For photometric analysis, we used the image data in overlapping regions to estimate global photometric correction function.

For RGB input data format, we estimate a tone mapping function for each RGB color channel of each input camera by minimizing the total mean square error of the pixel value discrepancies in all the overlapping regions $O_{m,n}$, $m = 1, 2, 3, 4$, and $n \equiv (m + 1) \bmod 4$. The pixel value discrepancy is defined as the difference between a pixel value from camera m and that of its spatial counter-part from camera n. To reduce computation, we down-sample the overlapping regions by block-averaging before computing the errors. The tone mapping functions for all four cameras are jointly optimized for each color channel, but independently optimized for different color channels. To achieve photometric correction, we apply the optimal tone mapping functions to the input frames. For YUV input data format, we first convert the YUV data to RGB data with standard YUV to RGB conversion matrix, then estimate the optimal tone mapping functions for the RGB channels, apply tone mapping correction, and finally get the YUV output by converting the photometric corrected data from RGB back to YUV format using standard RGB to YUV conversion matrix.

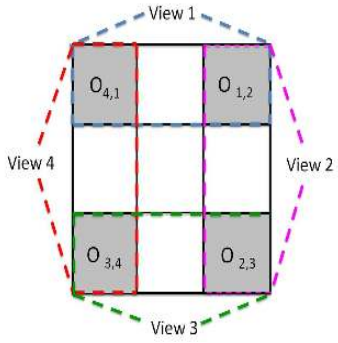

Figure 3. The views and overlapping regions in the composite surround view after geometric alignment. The composite surround view consists of data from all four input frames, view 1, 2, 3, and 4. $O_{m,n}$ is the overlapping region between view m and view n, and n is the neighboring view of view m in clock-wise order, $m = 1, 2, 3, 4$, $n \equiv (m + 1) \bmod 4$. Image data in the overlapping regions are used to compute tone mapping functions for photometric correction.

C. Surround View Synthesis

Synthesis function receives input video streams from four fish-eye cameras and creates a composite surround view. Synthesis creates the stitched output image using the mapping encoded in the geometric LUT. Figure 4 illustrates the view synthesis process. There are overlapping regions in the output frame, where image data from two adjacent input frames are required. In these regions, each output pixel maps to pixel locations in two input images.

In the overlapping regions, we can either blend image data from the two adjacent images or we can make a binary decision to use data from one of the two images. In this paper, we show results using the standard alpha-blending technique. Most of the color and brightness mismatch between adjacent views are already removed by the Photometric Alignment described in Sec. II-B. Alpha-blending is applied to the photometric corrected pixels to eliminate any residual seam boundaries and make the seams completely invisible. The alpha-blend weights are pre-stored in another LUT, which we refer to as the blending LUT. Output pixels are generated by a linear-combination of the corresponding input pixel values weighted by the respective blending weights. In the non-overlapping regions, to generate an output pixel, only one input pixel is fetched based on the geometric LUT. We then apply the proper tone mapping obtained through Photometric Alignment to the input pixel to get the final output pixel value.

III. Embedded Implementation

A. Architecture of the Proposed Solution

The architecture of our surround view solution is designed to meet the performance and memory requirements for embedded system. The flow diagram of the proposed solution is shown in Fig. 5. The geometric alignment analysis (block 101) receives input fish-eye camera images and generates the geometric LUT as described in Sec. II-A. The output geometric LUT (block 201) depends only on the location of the cameras and does not change significantly after the initial installation. Thus, geometric alignment function (block 101) is called only once by the system framework when the system is powered up. After completion, geometric LUT (block 201) is saved in the memory.

The Synthesis function (block 103) runs every frame. It takes four inputs: 1) the fish eye frames from the four cameras, 2) the geometric LUT (block 201), 3) the photometric LUT (block 203), i.e., the tone mapping functions, and 4) the blending LUT (block 202). The Synthesis function has two outputs: 1) the composite surround view frame, and 2) the statistics for photometric function (block 204). Statistics required by photometric function are the block averages of the image data in the overlapping regions of input frames. Ideally, the statistics should be collected by the photometric alignment function (block 102). This requires accessing input frames twice for each output (once for synthesis and once for photometric correction). To reduce memory bandwidth, we collect these statistics in synthesis function (block 103) for the current frame n, and use the statistics for photometric correction in the consecutive frame (n+1). Such a design limits all pixel-level, computationally-intensive operations required in each frame to the synthesis function block. It leads to a one frame latency, but this has not been an issue for image quality in our testing.

Finally, the photometric alignment analysis function (block 102) takes statistics (block 204) as the input, and generates photometric LUTs (block 203), i.e., the tone mapping functions for each camera. The photometric LUTs map an input value between 0 and 255 to an output value in the same range to compensate for both color and brightness mismatch among the four input frames.

B. DSP Optimization

Due to fish-eye warping, the access pattern for mapping output pixels to input image is not linear as illustrated by the dotted red line in Fig. 6 (a) and (b). Thus, the standard linear cache access sequence for the DSP to fetch data from external memory is not optimal for creating the composite surround view frame. In real-time implementation, the DSP fetches successive pixels along the horizontal line from the input image into the internal memory to speed-up access. With pixels following the curvature in the fish-eye image, there are several cache misses. Thus the processor has to wait for the relevant input pixel to be fetched into the internal memory.

To overcome the issue of cache-misses, we use a block-based Direct Memory Access (DMA) pattern. To enable this, we divide the output image into several blocks and process each output block independently. Fig. 6 shows an example of how the output blocks are mapped to one of

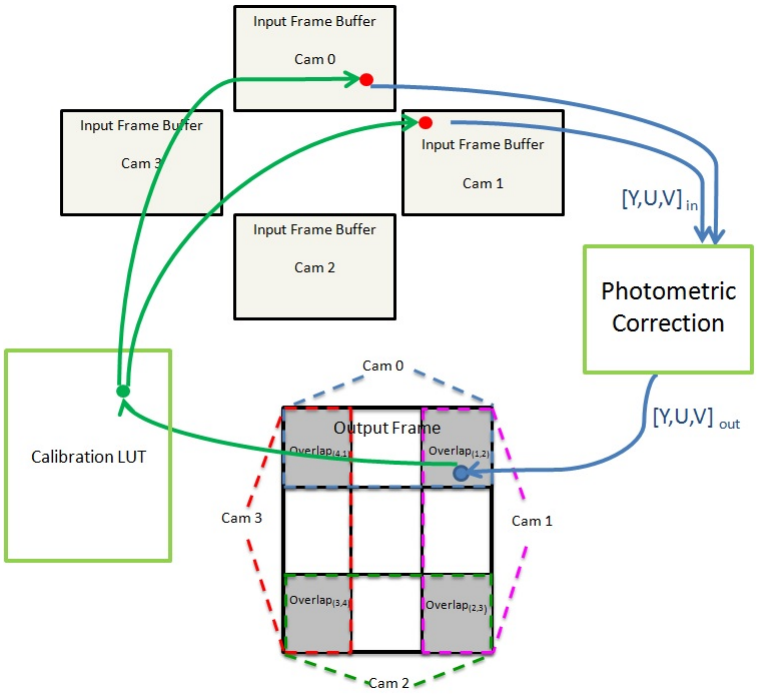

Figure 4. Illustration of the composite view synthesis process: to generate an output pixel: either two pixels (if the output is in the overlapping region) or a single pixel (if output is in non-overlapping region) are fetched from input frames through looking-up the geometric LUT. Each entry in the geometric LUT specifies the camera ID (i.e., index of the input camera(s)) and the coordinates in the input frame for generating the output pixel at current location. After the input pixels are fetched, we apply photometric correction and blending to these pixels to generate the final output pixel.

the input fish-eye images. For a given output block, the Geometric Alignment Algorithm generates the bounding box that tightly encloses the pixels in the input image required for synthesizing that block and encodes this information in the geometric LUT. Figure 6 illustrates the DMA blocks mapping between output and input images. For each output pixel, the location for input pixel access is stored in the LUT as an offset from the head of the corresponding input block. Since the offset location from the head of the block can be encoded with fewer bytes compared to the offset from the head of the entire input image, we further reduce the size of the LUT, therefore reducing memory bandwidth.

To process a given output block, the photometric LUT and the corresponding block from the blending LUT are also fetched into internal memory along with the corresponding input blocks. Thus, all the data required for processing the entire output block is available in the internal memory. Furthermore, we utilize a ping-pong DMA access pattern. When one block is being processed, the data necessary for the next block is brought into the internal memory simultaneously. This ensures that we minimize the processor idle time in waiting for data to be fetched into the internal memory.

As mentioned in the Sec. III-A, we also collect the block averages for the photometric alignment analysis during synthesis to save another memory access of the input image pixels. With these implementation strategies, we achieved real-time video frame rates for high-resolution output images.

IV. Results

Our surround view solution is implemented on an Automotive-grade ADAS SoC with two DSP C66x cores. Our system consists of four fish-eye cameras, each having a 180° FOV and 720p (1280×720) resolution. The four cameras are mounted on a toy jeep as shown in Fig. 7. From these four fish-eye videos, a composite surround view is synthesized at 30 fps. The composite surround view has a dimension of 880×1080. The 880×1080 output resolution was chosen to match our display constraints; other output resolutions can be achieved in a similar manner.

The geometric alignment algorithm runs on one DSP C66x ($600Mhz$), but is called only once at system powering up. It takes about 5 seconds to finish and consumes the entire DSP during this time. The synthesis algorithm (with de-warping and blending operation) runs every frame on a second C66x ($600Mhz$) core. It takes about 75% loading of the $600Mhz$ DSP for 880×1080 output resolution. The DSP loading for Synthesis is a function of the stitched output resolution. The photometric alignment algorithm, which uses image statistics collected during Synthesis runs every frame

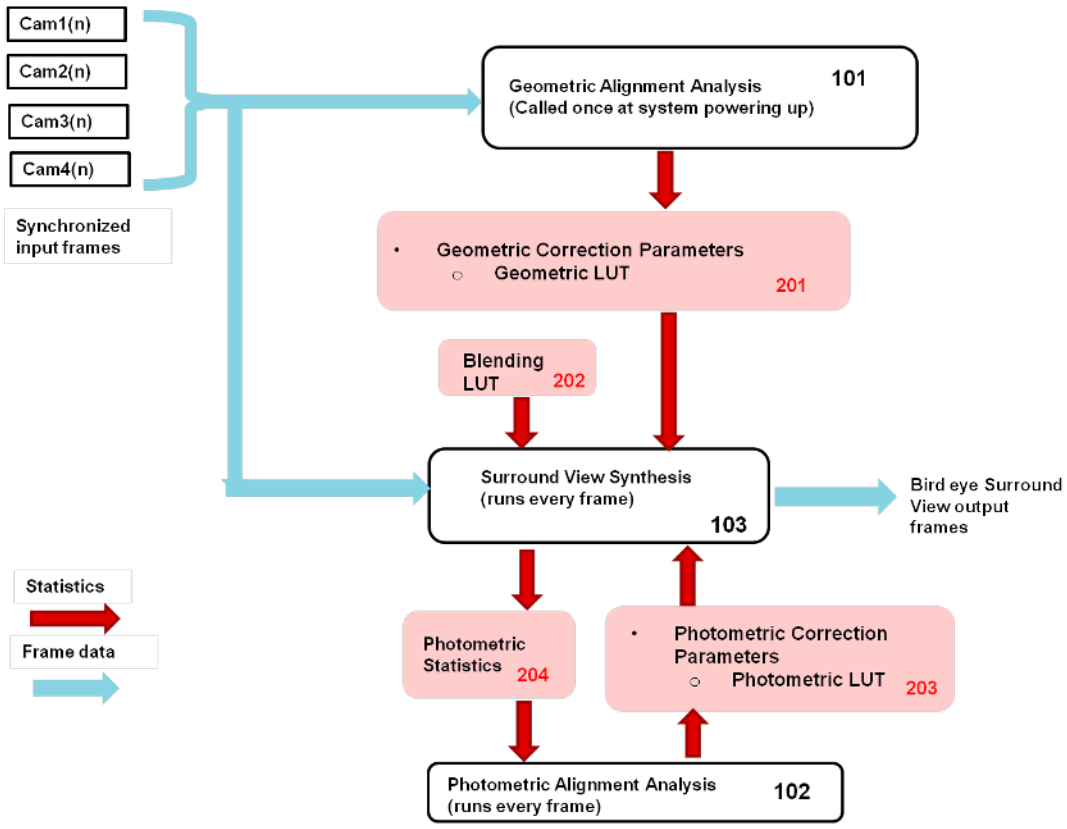

Figure 5. Flow diagram of the proposed surround view camera solution

on the first DSP utilizing 3% of its capacity. The composite surround view with varying algorithm complexities are shown in Fig. 8: (a), without geometric alignment and photometric alignment; (b), with geometric alignment, but without photometric alignment, and finally, (c), the output with our proposed geometric and photometric alignment algorithms. The proposed surround view solution produces a seamlessly stitched composite view as if it were taken by a camera above the car. A video of the live performance of our real-time surround view prototype can be found at ¡Surround View Demo at CES2014¿.

V. Conclusion

In this paper, we presented a complete real-time surround view solution that is ready for ADAS applications. We described the three main components for our surround view solution: 1) Geometric Alignment, 2) Photometric Alignment, and 3) Composite View Synthesis. We presented the design and architecture of the entire solution. Furthermore, we described in detail the techniques used to optimize our solution for real-time performance on DSP C66x. With the proposed solution, we achieved high quality stitched HD video output at 30 fps.

References

[1] C. C. Lin and M. S. Wang, "A vision based top-view transformation model for a vehicle parking assistant," *Sensors*, vol. 12, pp. 4431–4446, 2012.

[2] Yu-Chih Liu, Kai-Ying Lin, and Yong-Sheng Chen, "Bird's-eye view vision system for vehicle surrounding monitoring," *International Workshop on Robot Vision*, pp. 207–218, 2008.

[3] Frank Nielsen, "Surround video: a multihead camera approach," *Vis. Comput.*, vol. 21, pp. 92–103, 2005.

[4] Kapje Sung, Joongryoul Lee, Junsik An, and Eugene Chang, "Development of image synthesis algorithm with multi-camera," *2012 IEEE Vehicular Technology Conference (VTC Spring)*, pp. 1–5, 2012.

[5] Din Chang Tseng, Tat Wa Chao, and Jiun Wei Chang, "Image-based parking guiding using ackermann steering geometry," *Appl. Mech. Mater.*, vol. 437, pp. 823–826, 2013.

[6] C. Harris and M. Stephens, "A combined corner and edge detector," *Proc. Alvey Vision Conf.*, pp. 147–151, 1988.

[7] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: Binary robust independent elementary features," *Proc. European Conf. on Comput. Vis. (ECCV)*, 2010.

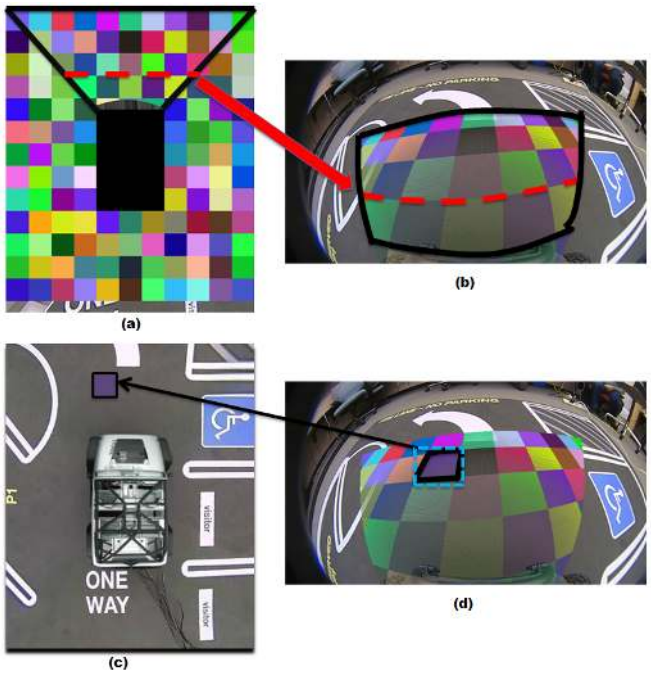

Figure 6. Illustration of DMA blocks mapping between output and input images. (a) and (b) show the mapping of a portion of the output surround view image to one of the input images, with the corresponding blocks overlaid. The dotted red line indicates the warped access required in the input image to fetch a horizontal line in the output frame. (c) and (d) show the mapping of one DMA block from output image to input image. The output DMA block and its corresponding input pixels are highlighted by black borders. The bounding box for the input pixels, i.e., the input DMA block is highlighted by the dotted cyan lines.

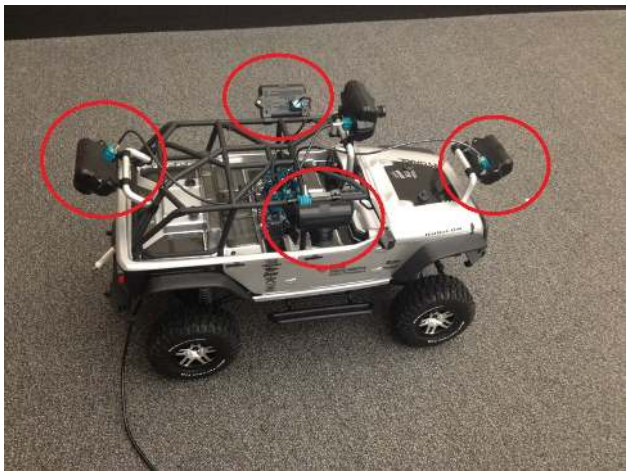

Figure 7. A toy jeep with surround view cameras. The cameras are highlighted with red circles.

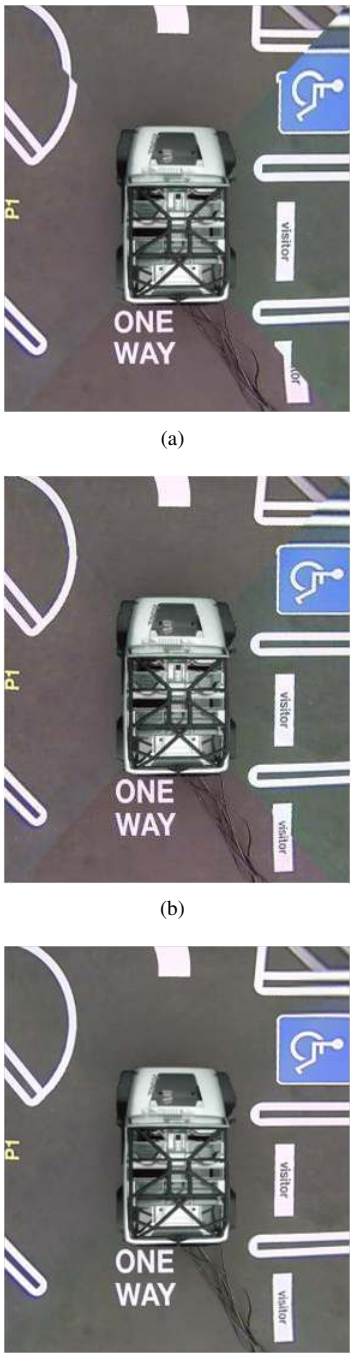

Figure 8. The composite surround view synthesized from four fish-eye frames shown in Fig. 1: (a), without proper geometric alignment or photometric alignment; (b), with proper geometric alignment, but without photometric alignment, and (c), with our proposed geometric and photometric alignment algorithms. Without proper geometric alignment, misalignment at view boundaries are very noticeable. Without proper photometric alignment, the stitched surround view suffer color and brightness inconsistency from view to view. In (c) we achieve high-quality seamlessly stitched result.