

SEAMLESS 3D SURROUND VIEW WITH A NOVEL BURGER MODEL

Lin Zhang, Juntao Chen, Dongyang Liu*, Ying Shen*, and Shengjie Zhao

School of Software Engineering, Tongji University, Shanghai, China

ABSTRACT

In recent years, the 3D surround view (3D-SV) system has become a hot research topic in the field of Advanced Driver Assistance Systems (ADAS). It can be used to form a stereoscopic view of the surrounding 3D environment by using 4 car-mounted surround cameras, and users can switch the viewpoints for virtual observation conveniently. However, there are still many problems in how to stitch calibrated images to the panoramic view and how to project the panorama to the surround view. In this paper, we introduce the graph cut algorithm and multi-band blending to the panorama stitching phase. In addition, we design a new hamburger-shaped 3D geometric model to be the carrier of the panorama for texture mapping. Our 3D-SV system can make drivers have an immersive visual experience. Experimental results show that the 3D-SV generated by our method is less distorted and looks more natural than the other competitors.

Index Terms— ADAS, surround view, graph cut, multi-band blending, texture mapping

1. INTRODUCTION

The first autonomous car appeared in 1980s, but vehicles without human intervention have not been put into commercial operation yet for the safety, technology and liability concerns [1] [2]. Instead, ADAS are the main way to improve the comfort and safety of driving. The 3D surround view (3D-SV) system is an important member of the ADAS family. Drivers can sit in the car and observe the surroundings with artificial 3D-SV synthesized by images captured by car-mounted cameras. Currently, the studies in the field of 3D-SV systems are quite sporadic and they usually follow a common pipeline: image capturing, image undistorting, panorama stitching and texture mapping. By investigating the literature, we find that the differences among current methods for generating 3D-SV mainly lie in their ways for panorama stitching and texture mapping.

For panorama stitching, the first inevitable step is to align adjacent views by estimating their homography matrix. Further more, in order to make the panorama look natural, the strategy to fuse the overlap regions also need to be studied. To this end, several different schemes have been proposed.

Gao *et al.* made use of the alpha fusion to fuse two adjacent images [3]. Lin *et al.* used Laplace pyramid fusion to make the fusion result more smooth [4]. Photometric alignment is another way to correct the mismatch in brightness and color between two adjacent views. In terms of texture mapping, traditional surround view systems project textures onto the ground plane and generate a bird's eye view [5] [6]. 3D-SV systems use 3D models as the carriers of 2D textures and various 3D models have been proposed, such as the ship model [3], the bowl model [7], and the cylinder model [4].

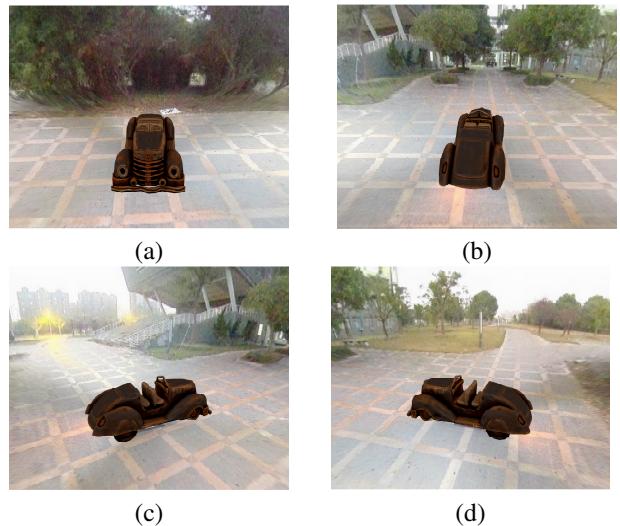


Fig. 1. Images rendered from different perspectives in our 3D-SV system. (a) front view, (b) back view, (c) left view, (d) right view.

In order to make the 3D-SV immersive, the panorama must be seamless. However, simply applying the alpha fusion or photometric alignment is not enough to make the panorama look seamless. These fusion techniques will cause the panorama to be discontinued and ghosted. Besides, current 3D models used for texture mapping usually make the 3D-SV look distorted. The ship model and bowl model have a mutation between the horizontal bottom and the arc-shaped wall and thus far objects will be distorted. The cylindrical model does not meet drivers' habits since their viewpoints will be limited. Actually, a well-designed 3D model is indispensable to make the 3D-SV more natural.

In this paper, we aim to find a more reasonable way to

*Corresponding authors: {1753575, yingshen}@tongji.edu.cn.

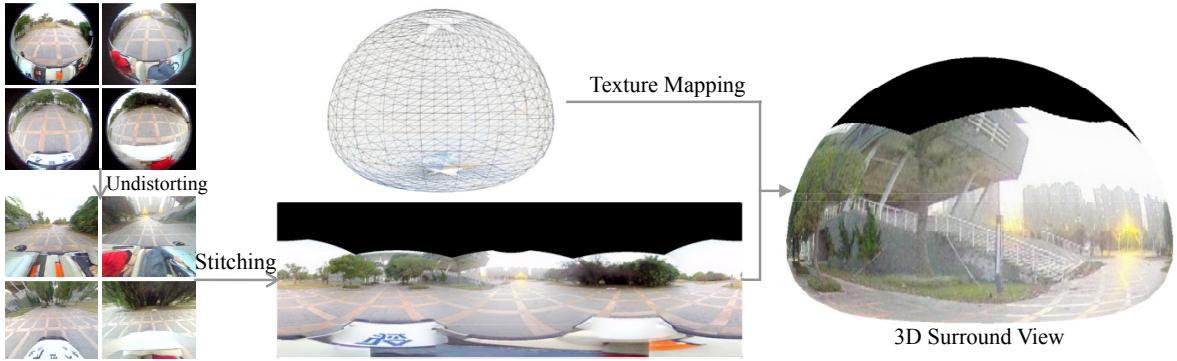


Fig. 2. The flowchart of the system.

implement 3D-SVs for ADAS. Fig. 1 shows several 3D-SV images from different viewpoints generated by our system. Our contributions are twofold. First, inspired by [8], we introduce the graph cut algorithm and multi-band blending to the panorama stitching phase. The graph cut algorithm is effective in image matting. Adjacent images can be stitched along an optimal seam. In this way, objects, textures and lines can be ensured to stay integrated. After finding optimal seams, adjacent images are fused with the multi-band blending strategy, which can make the transition between two adjacent views smooth. Second, we propose a new 3D model for texture mapping, which is named as “Burger Model” since its shape looks like a burger. With the Burger model, our 3D-SV looks immersive and less distorted.

The organization of the paper is as follows. In Sect. 2, we describe the implementation details of the system. In Sect. 3, we present our experimental results and have a brief discussion. Finally, Sect. 4 concludes the paper.

2. SURROUND VIEW FUSION AND MAPPING

The flowchart of our system is shown in the Fig. 2. Firstly, images are captured from 4 car-mounted fisheye camera (larger than 180° coverage), which can cover all viewpoints of the surroundings. Then, the fish-eye images are undistorted by Zhang’s algorithm [9]. Next, all undistorted images are fused into a seamless panorama by the graph cut algorithm and multi-band blending. Finally, the 2D panorama is projected onto the 3D model and the 3D-SV can be generated for virtual observation.

Since the study for undistorting images is relatively mature, only “panorama stitching” and “texture mapping” are discussed in-depth in this paper.

2.1. Panorama stitching

Undistorted images will be first stitched into a panorama. This operation consists of three main steps.

First, extract features from each image. In our implementation, the SURF feature detector [10] serves as the feature

detection algorithm. Then, features are matched to estimate homography matrices for every two adjacent images.

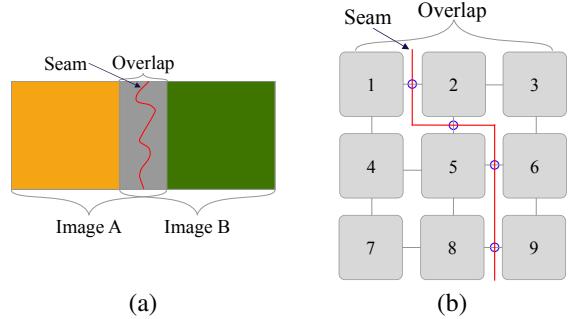


Fig. 3. A seam between two images. (a) is schematic diagram of a seam and (b) demonstrates the min-cut problem in the overlapping region.

Second, to reduce the discontinuity between images, an image synthesis technique [11] is modified to find out a seam for every two adjacent views. Fig. 3(a) shows how the seam works. After projecting one image to another image’s plane, these two images would have an overlapping region. The seam determines which image contributes pixels at different locations in the overlapping region. To be more specific, finding out the optimal seam is considered as a minimum cut problem as demonstrated in Fig. 3(b). A path cuts the pixels of the overlapping region into two disjoint subsets. Every pixel is a vertex and connected with nearby pixels. To define the weight of each edge, the cost of separating two neighboring pixels needs to be defined. Suppose that $A(s)$ and $B(s)$ are the vectors of the intensity at the position s in images A and B . A cost function M [11] can be defined as,

$$M(s, t, A, B) = \|A(s) - B(s)\| + \|A(t) - B(t)\|$$

where $\|\cdot\|$ denotes an appropriate norm. Actually, it represents the difference between two adjacent pixels s and t . With all edges weighted, a min-cut of this graph can be found by using algorithms like Stoer-Wagner [12] and it is considered is the optimal seam of the overlapping region.

As concluded [13], since parallax is always inevitable, images do not need to be perfectly aligned over the whole overlapping region for stitching. Instead an optimal seam can help the fused image look natural and smooth in visual experience.



Fig. 4. The effect of multi-band blending. (a) is a fused image without multi-band blending and (b) is (a)'s counterpart with multi-band blending.

Third, image blending is applied on the stitched images. Although an optimal seam has been found, the transition around the seam is not continuous. Image blending can make this transition more smooth. In this paper we use the multi-band algorithm [14] for blending. Compared to traditional feathering or alpha blending [15], the multi-band algorithm takes both details near the seam and large scale features such as illumination into consideration. So how to fuse high-frequency features and low-frequency features? The pyramid representation is a good choice. Let $I(x, y)$ be the original image, the level $n + 1$ of the Gaussian pyramid G can be defined as,

$$G_{n+1}(i, j) = 4 \sum_{m=-2}^2 \sum_{n=-2}^2 W(m, n) G_n\left(\frac{i-m}{2}, \frac{j-n}{2}\right)$$

where $W(m, n)$ is a 5×5 Gaussian kernel and G_0 is the original image. G_{n+1} is $1/4$ downsampled G_n convolved with a Gaussian kernel. Finally, several levels of the pyramid [$G_0, G_1, G_2, \dots, G_n$] can be calculated. Since Laplacian of Gaussian (LoG) can be approximated with difference of Gaussian (DoG), the Laplacian pyramid can be defined as:

$$L_n(i, j) = G_n(i, j) - \text{expand}(G_{n+1})(i, j)$$

where $\text{expand}(G)$ means resizing the width and height of G twice. Similarly, several levels of the pyramid [$L_0, L_1, L_2, \dots, L_{n-1}$] can be calculated.

Then, the two LoG pyramids are fused by using feathering blending. Then, all fused levels are expanded to the same size of L_0 . Add all expanded levels and we finally get the fused overlapping region.

2.2. Texture mapping

For texture mapping, a novel “burger”-shaped 3D model is proposed in this paper as shown in Fig. 5. The burger model

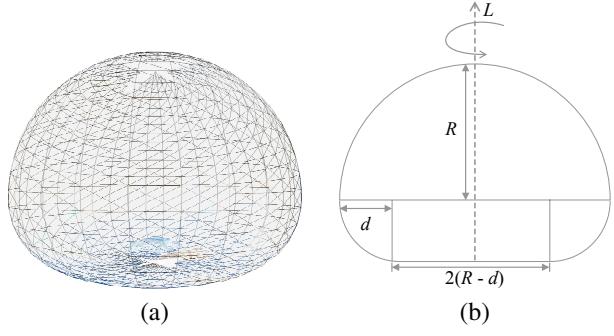


Fig. 5. 3D model and its sectional view. (a) burger model, (b) maximum cross section.

is a closed surface of revolution created by rotating a curve around the axis of rotation L in Fig. 5(b). The curve consists of a semicircle with radius R , two 90° arcs with radius d and a line segment of length $2(R - d)$. The maximum cross section is shown in Fig. 5(b). Thus, the transition from the top hemisphere to the bottom part is not only continuous but also smooth, while the ship model [3] is only continuous but non-differentiable.

So far, there is only one task remaining, mapping the 2D texture to the the 3D burger model, for which we adopt the object centroid mapping scheme [16].

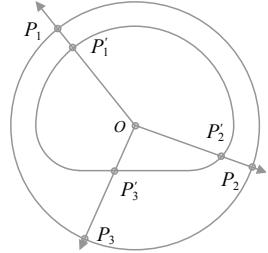


Fig. 6. Object centroid texture mapping.

Firstly, map the 2D texture to a sphere. As we know, texture coordinates usually range from $(0, 0)$ to $(1, 1)$. However, a sphere has three demensions. How to map from 2D to 3D? A commonly-used way is to use latitude and longitude to represent the position of an object on our earth. A geographic coordinate system [17] can describe every position on the earth’s surface specifically since it has only two demensions too. More precisely, this coordinate system is more like a spherical coordinate system with a fixed radial distance (a unit length) and it uses a polar angle ($\theta \in [0, \pi]$) and an azimuth angle ($\phi \in [0, 2\pi]$) to describe the position of a point. Hence a simple linear mapping can be performed from the radius fixed spherical coordinate system $[1, \theta, \phi]$ to the texture coordinate system $[u, v]$. The texture coordinates $[u, v]$ can be defined as:

$$u = \theta/\pi$$

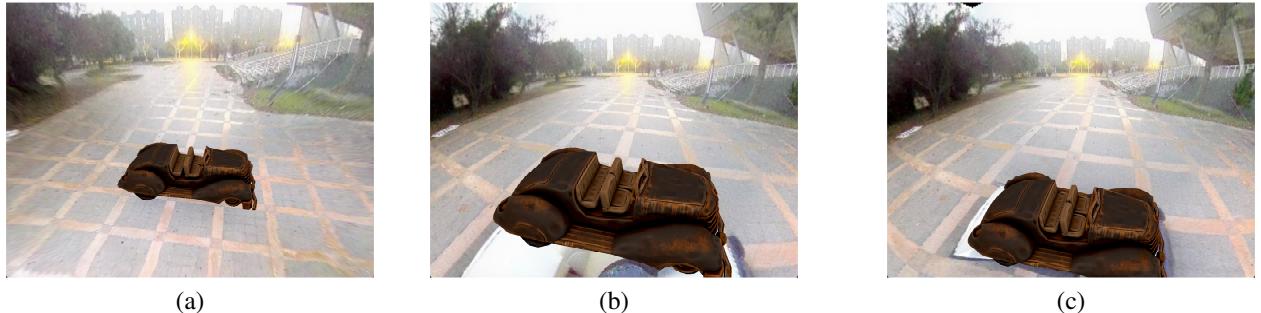


Fig. 7. Surround views generated by different models. (a) burger model, (b) sphere model [7], and (c) ship model [3].

$$v = \phi/2\pi$$

Since meridians near two poles are more intensive than those near the equator, texture near the poles will be highly distorted. Fortunately, we do not really care about what there is above us or what there is under the vehicle.

Secondly, map the texture from sphere to our burger model, using the real spherical coordinate system. The sphere model is concentric with the hemisphere of our burger model and their centre is denoted by O . Fire several bundles of rays from O . They intersect with sphere at P_1 , P_2 and P_3 , and intersect with the burger model with P'_1 , P'_2 and P'_3 . P'_1 , P'_2 and P'_3 have the same texture coordinates with P_1 , P_2 and P_3 . In this case, our texture mapping is single-valued.

3. EXPERIMENTAL RESULTS

Our 3D-SV system is implemented on an experimental vehicle. Four fisheye cameras are mounted on the vehicle and each has a 184° FOV. With our experimental settings, the panorama is fused at a speed of 20 fps on 2.7 GHz Intel Core i5. The graph cut algorithm runs only once in a period of time and the parameter can be used to stitch a series of frames and update the stitching seam every 50 frames. It takes about 0.5 second to find the stitching seams but this can be computed in parallel with panorama stitching. Hence the 3D-SVs can be generated in real time.

Fig. 7 presents the surround view generated by the burger model, sphere model [7] and ship model [3]. We can have an intuitive feeling of the difference between these two models and the proposed burger model. It can be seen that the surround view generated with buger model looks clear and natural. What's more, it is seamless and continuous. The ship model has an obvious mutation between its bottom floor and its arc-shaped wall. Moreover, with the ship model, far objects appear highly distorted. This can mislead drivers seriously. The sphere model is centrosymmetric and the vehicle is at the center of the sphere. Consequently, the ground plane will be projected to the lower hemisphere and thus it will be severely distorted.

As for panorama stitching, we compare the panorama fused with graph cut and multi-band blending and the one

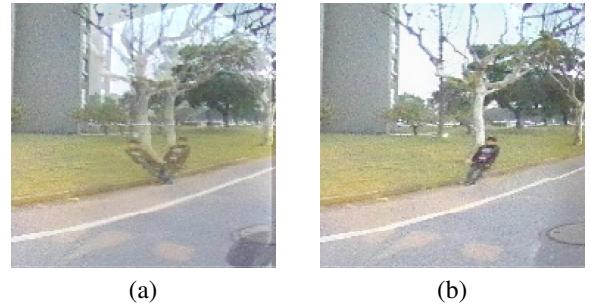


Fig. 8. Panoramas generated by different methods. (a) is the panorama stitched by alpha blending; (b) is the one stitched using the graph cut algorithm and multi-band blending.

with only simple alpha blending. The viewpoints of adjacent cameras will cause apparent parallax and usually the stitched panorama will be ghosted and kind of blurry. As shown in Fig. 8, the tree and the passer-by both have a ghosted look, while our fusion scheme solves the parallax problem very well.

Our 3D-SV system performs quite well while the vehicle is moving. Observers can switch the viewpoints freely and it looks like that observers are watching the vehicle in the real world. A demo video is provided as the supplementary material.

4. CONCLUSIONS

In this paper, we introduce a new approach to generate 3D-SV. By using the graph cut algorithm to find an optimal seam and the multi-band blending to smooth the overlapping regions between two adjacent views, the panorama becomes reliable and seamless. In addition, we come up with a new texture carrier, the burger model. Experimental results corroborate that our proposed system can generate seamless and less distorted 3D-SV images.

5. ACKNOWLEDGMENT

This research was funded in part by the Natural Science Foundation of China under Grant No. 61672380 and in part by the National Key Research and Development Project under Grant No. 2017YFE0119300.

6. REFERENCES

- [1] N. Kalra and S. M. Paddock, “Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability?,” *Transportation Research Part A: Policy and Practice*, vol. 94, pp. 182 – 193, 2016.
- [2] M. Schellekens, “Self-driving cars and the chilling effect of liability law,” *Computer Law & Security Review*, vol. 31, no. 4, pp. 506 – 517, 2015.
- [3] Y. Gao, C. Lin, Y. Zhao, X. Wang, S. Wei, and Q. Huang, “3-D surround view for advanced driver assistance systems,” *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 1, pp. 320–328, 2018.
- [4] M. Lin, G. Xu, X. Ren, and K. Xu, “Cylindrical panoramic image stitching method based on multi-cameras,” in *CYBER*, 2015, pp. 1091–1096.
- [5] L. Zhang, J. Huang, X. Li, and L. Xiong, “Vision-based parking-slot detection: A DCNN-based approach and a large-scale benchmark dataset,” *IEEE Trans. Image Processing*, vol. 27, no. 11, pp. 5350–5364, 2018.
- [6] M. Yu and G. Ma, “360° surround view system with parking guidance,” *SAE Int. J. Commer. Veh.*, vol. 7, pp. 19–24, 2014.
- [7] B. Zhang, V. Appia, I. Pekkucuksen, Y. Liu, A. U. Batur, P. Shastray, S. Liu, S. Sivasankaran, and K. Chitnis, “A surround view camera solution for embedded systems,” in *CVPR*, 2014, pp. 676–681.
- [8] Matthew Brown, David G Lowe, et al., “Recognising panoramas,” in *ICCV*, 2003, vol. 3, p. 1218.
- [9] Z. Zhang, “A flexible new technique for camera calibration,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, pp. 1330–1334, 2000.
- [10] H. Bay, T.uytelaars, and L. Van Gool, “Surf: Speeded up robust features,” in *ECCV*, 2006, pp. 404–417.
- [11] V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick, “Graphcut textures: Image and video synthesis using graph cuts,” *ACM Trans. Graph.*, vol. 22, no. 3, pp. 277–286, 2003.
- [12] M. Stoer and F. Wagner, “A simple min-cut algorithm,” *J. ACM*, vol. 44, no. 4, pp. 585–591, 1997.
- [13] F. Zhang and F. Liu, “Parallax-tolerant image stitching,” in *CVPR*, 2014, pp. 3262–3269.
- [14] P. J. Burt and E. H. Adelson, “A multiresolution spline with application to image mosaics,” *ACM Trans. Graph.*, vol. 2, no. 4, pp. 217–236, 1983.
- [15] M. Uyttendaele, A. Eden, and R. Skeliski, “Eliminating ghosting and exposure artifacts in image mosaics,” in *CVPR*, 2001, vol. 2, pp. 509–516.
- [16] E. A. Bier and K. R. Sloan, “Two-part texture mappings,” *IEEE Comput. Graph. Appl.*, vol. 6, no. 9, pp. 40–53, 1986.
- [17] A. G. McNish, “Geomagnetic coordinates for the entire earth,” *Terrestrial Magnetism and Atmospheric Electricity*, vol. 41, no. 1, pp. 37–43.