An Efficient Hardware Architecture of Coordinate Transformation for Panorama Unrolling of Catadioptric Omnidirectional Images

Seung-Ho Lee*★

Abstract

In this paper, we present an efficient hardware architecture of unrolling image mapper of catadioptric omnidirectional imaging systems. The catadioptric omnidirectional imaging systems generate images of 360 degrees of view and need to be transformed into panorama images in rectangular coordinate. In most application , it has to perform the panorama unrolling in real-time and at low-cost, especially for high-resolution images.

The proposed hardware architecture adopts a software/hardware cooperative structure and employs several optimization schemes using look-up-table(LUT) of coordinate conversion. To avoid the on-line division operation caused by the coordinate transformation algorithm, the proposed architecture has the LUT which has pre-computed division factors. And then, the amount of memory used by the LUT is reduced to 1/4 by using symmetrical characteristic compared with the conventional architecture.

Experimental results show that the proposed hardware architecture achieves an effective real-time performance and lower implementation cost, and it can be applied to other kinds of catadioptric omnidirectional imaging systems.

Keywords: Catadioptric Omnidirectional Images, Panorama Unrolling, 1/4 LUT Generation

I. Introduction

Catadioptric omnidirecrtional images are captured using lenses and special mirrors (such as parabolic reflecting mirror or hyperboloid mirror)[1],[2],[3],[4]. Catadioptric omnidirectional imaging systems can provide 360 degrees view angle of surrounding scenes and be widely used for robot vision, supervision and other situations[5],[6],[7], where the omnidirectional images reduce the number of needed cameras. However, omnidirectional images usually have the problem of concentric circles resulting in distortion. In order to process in a similar way as using common lenses, they need to be unrolled to planar images. Generally, real-time image processing requires enormous throughput rate and huge amount of operations. To satisfy this limitation, some LUT-based approaches were proposed on FPGA [8],[9]. For example, Lidong, et al.[10] proposed a hardware module to transform omnidirectional images into rectangular panorama images by a performing ray-trace coordinate mapping algorithm. However, though high speed performance is achieved by using LUT, the LUTs result in huge memory usage. In this paper, we propose a hardware design which can transform the omnidirectional circularity images with the size of 1280*1024 into the rectangular panorama images with the size of 3200*768, and the size of LUTs has been reduced to 3KB.

The paper is organized as followed: Section 2 describes the main ideas for the low-cost hardware

* Department of Electronic Engineering, Hanbat National University
★ Corresponding author
Manuscript received Mar. 5, 2011 ; received Mar. 29, 2011

implementation. Section 3 describes the Quad LUT generation and a 9-point interpolation of proposed hardware design is detailed. Finally, experimental results are introduced in section 4 and conclusions are commented in section 5.

II. PROPOSED HARDWARE STRUCTURE

As shown in Fig. 1, the proposed hardware design includes four modules:

1) The LUT generation module calculates the corresponding coordinates in omnidirectional image using transformation equations by software and generates the look-up tables (LUTs).

2) The coordinate calculation module computes four coordinates at one time using the symmetrical characteristics, and computed coordinates divide into integer part which is used to generate address to read data from input frame memory and fraction part which is used to do interpolation.

3) RGB to YCbCr module transforms input RGB data to YCbCr data in the 4:2:2 format and store them into dual port RAM.

4) The interpolation module removes aliasing and enhances image quality by 9-point interpolation algorithm.

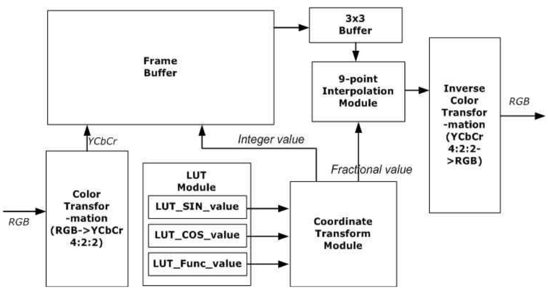
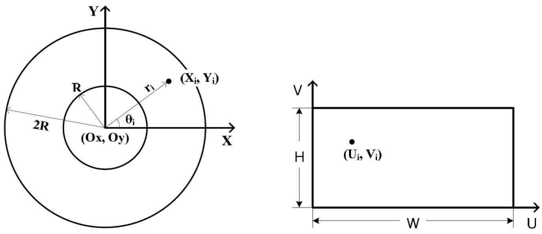


Fig. 1 . Proposed Hardware Structure

III. HARDWARE OPTIMIZATION AND ACCELERATION

3-1. Quad Omnidirectional Image Transformation

Because the calculation of omnidirectional image transformation function is not the emphasis of this paper, we adopt the algorithm of [8] to get the function directly.



a) Omnidirectional Image b) Rectangular image

Fig. 2. Relationship between Omnidirectional and Rectangular Images

As shown in Fig. 2, (a) is the omnidirectional image need to be unrolled, (b) is the panorama image corrected. The panorama unrolling is defined as the transformation that moves the pixels of (a) to the corresponding location in (b). The relationship between the parameters (ri, Θi) and (Ui, Vi) can be described as (1) and (2).

$$ri = R \cdot Vi/H + R \quad (1)$$

$$\theta i = 2 \cdot \pi \cdot Ui/W \quad (2)$$

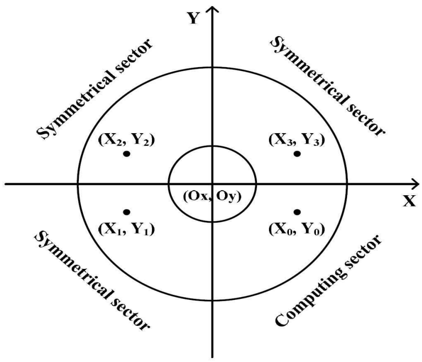
Based on (1) and (2), the coordinate mapping relationship between the pixels (Xi, Yi) and the pixels (Ui, Vi) can be derived as (3).

$$\begin{aligned} Xi &= ri \cdot \cos\theta i + Ox \\ &= (R \cdot Vi/H + R) \cdot \cos(2 \cdot \pi \cdot Ui/W) + Ox \\ Yi &= ri \cdot \sin\theta i + Ox \\ &= (R \cdot Vi/H + R) \cdot \sin(2 \cdot \pi \cdot Ui/W) + Oy \end{aligned} \quad (3)$$

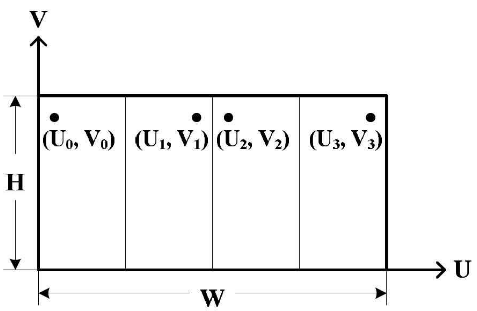
In order to reduce the memory size and realize the real-time image processing, we proposed a Quad transformation algorithm. Because of the characteristics of symmetric in trigonometric function, we can reuse the transformed value of 0~90 degree in the domain of omnidirectional image for other degrees. The Quad transformation algorithm is illustrated as Fig. 3. This algorithm includes four steps: First, define the center of the omnidirectional circularity image. And define the resolution of the rectangular panorama image. Second, divide the omnidirectional circularity image into four symmetrical regions and partition the rectangular panorama image into four rectangular regions as Fig. 3. Third, based on any one of these sectors, compute the coordinate according to (3). Fourth, the other three sectors are calculated by the principle of symmetrical coordinates as (4).

$$\begin{aligned} X_1 &= X_0 + 2Ox, & Y_1 &= Y_0 \\ X_2 &= -X_0 + 2Ox, & Y_2 &= -Y_0 + 2Oy \\ X_3 &= X_0, & Y_3 &= -Y_0 + 2Oy \end{aligned} \tag{4}$$

Obviously, because four coordinates are calculated at once, it is possible to operate the transformation fastly and save memory size used by LUT.



a) Four Sectors in Omnidirectional Image



b) Four Sectors in Rectangular Image

Fig. 3. Four Symmetrical Partition

3-2. Quad-LUT Generation

Because the equation of transformation consist of complex trigonometric functions, multiplications and divisions, it is very difficult to implement real-time unrolling of omnidirectional images on FPGA. Generally, a pre-calculated look-up table (LUT) can be used to reduce the computational complexity. there should be two tables to realize the implementation, LUT_X for values of X and LUT_Y for values of Y. If the values of U and V are given, it is easy to get the corresponding values of U and V by table_U and table_V.

However, the size of LUT is increased by the size of rectangular panorama images. We use the rectangular panorama images with the size of 3200*768 and also use 10 bits to record the floating point values. The total size of LUT is too large as 3200*768*10bits*2 = 6.1MB to use on the FPGA.

The integration of look-up table and calculation is proposed in our implementation. Formula (3) can be expressed as follows:

$$\begin{aligned} Xi &= FUNC(V) \cdot COSTABLE(U) + Ox \\ Yi &= FUNC(V) \cdot SINTABLE(U) + Oy \end{aligned} \tag{5}$$

Where

$FUNC(V) = R \cdot Vi / H + R$

$COSTABLE(U) = \cos(2 \cdot \pi \cdot Ui / W)$

$SINTABLE(U) = \sin(2 \cdot \pi \cdot Ui / W)$.

We generate three look-up tables: SIN_table(U), COS_table(U) and FUNC(V). Then, we can calculate the values of U and V by using only one multiplication and one addition or subtraction. In order to reduce the size of LUT, we store Quad parts of the whole table by utilizing the symmetrical characteristic mentioned above. Quad LUT generation algorithm stores only 1/4 size of the rectangular panorama image using symmetrical characteristic of omnidirectional circularity image. As a result, the total size of LUTs has been reduced to (800*2+768)*10bits = 3KB.

3-3. 9-point Interpolation

After coordinate calculation, the coordinates of pixels are usually located among the actual pixels, so the interpolations are used to improve the output image quality.

The nearest neighbor interpolation algorithm is very fast, but results in grainy image. More complex interpolation methods with floating point are also not suitable for hardware implementation efficiently. By tradeoff with speed and performance, we use a 9-point interpolation[11].

The 9-point interpolation algorithm is illustrated as Fig. 4. Among the 2*2 interpolation window, there are five interpolated pixels, except for the actual pixels. All interpolated pixels are calculated by taking the average of its adjacent actual pixels. For example, the value of the interpolated pixel (x, y+0.5) is the average of the values of the actual pixels (x, y) and (x, y+1). The value of (x+0.5, y+0.5) is the average of the values of the actual pixels (x, y), (x+1, y), (x, y+1) and (x+1, y+1). The red triangle in Fig. 4 is mapped to interpolated pixel (i+0.5, j).

It means that, 9-point interpolation only involves averages of either two or four quantities. Division by two or four can be calculated by right-shifting the sum by one or two bits. This algorithm simplifies computation without significant sacrifice in the quality of the rectangular panorama image, and it is also easy for hardware implementation.

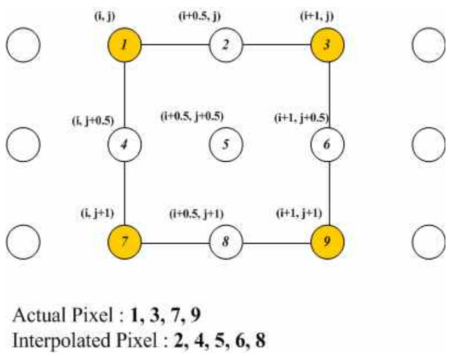


Fig. 4. Illustration of the 9-point Interpolation Algorithm

Ⅳ. EXPERIMENTAL RESULT

In our experiment, the design described in Verilog HDL, and catadioptric omnidirectional image sensor from INEW DIGITAL and Xilinx Vertex-5 xc5vlx330t device are used. As Fig. 5 and Fig. 6 shows the catadioptric omnidirectional camera, and the omnidirectional image with the size of 1280*1024 is transformed into rectangular panorama image with the size of 3200*768 in real-time. All of memory used in simulation is on-chip block memory of FPGA. The memory usages are shown in Table. 1 and Table 2.

Table 1. Memory Usage of Proposed Architecture

Memory Attribute		Memory Size(bits)	Total Memory Size(Kbits)
Input Buffer	Y	1280*1024*8	20971
	Cb	640*1024*8	
	Cr	640*1024*8	
3x3 Buffer	Y	640*8	15
	Cb	640*8	
	Cr	640*8	
Output Buffer	Y	1280*1024*8	20971
	Cb	640*1024*8	
	Cr	640*1024*8	

Table 2. Comparison with Conventional Method

	Conventional Method[10] (32bits)	Proposed Method (32bits)	Proposed Method (10bits)
LUT Size (KB)	28KB	9KB	3KB

Fig 5. Catadioptric Omnidirectional Imaging System


Fig 6. Rectangular Panorama Image

Ⅳ. CONCLUSION

This paper proposed a low-cost hardware structure for panorama unrolling of catadioptric omnidirectional images. The design can realize a transformation from the omnidirectional images with the size of 1280*1024 into the rectangular panorama images with the size of 3200*768 by using Quad look-up tables and the 9-point interpolation. The module can be applied to other transformation algorithms and be directly used in low-cost catadioptric omnidirectional imaging systems.

REFERENCES

[1] K. Yamazawa, Y.Yagi, and M. Yachida, "Omnidirectional Imaging with Hyperboloidal Projection," Proc. 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'93), vol. 2, 1993, pp. 1029-1034.
[2] S. K. Nayar, "Catadioptric Omnidirectional Camera," Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'97), 1997, pp. 482-488.
[3] Sooyeong Yi, and N. Ahuja, "An Omnidirectional Stereo Vision System Using a Single Camera," Proc. 18th International Conference on Pattern Recognition (ICPR'2006), vol. 4, 2006, pp. 861-865.
[4] Jiyong Zeng, and Xianyu Su, "Hyperboloidal Catadioptric Omnidirectional Imaging System," Acta Optica Sinica, vol. 23, No. 9, 2003, pp. 1138-1142.
[5] Y. Onoe, N. Yokoya, K. Yamazawa, and H. Takemura, "Visual Surveillance and Monitoring System Using an Omnidirectional Video Camera," Proc. Fourteenth International Conference on Pattern Recognition (ICPR'1998), vol. 1, 1998, pp. 588-592.
[6] T. Ehlgen, T. Pajdla, and D. Ammon, "Eliminating Blind Spots for Assisted Driving," IEEE Transactions on Intelligent Transportation Systems (TITS'2008), vol. 9, No. 4, 2008, pp. 657-665.
[7] T. Ehlgen, M. Thorn, and M. Claser, "Omnidirectional Cameras as Backing-Up Aid," Proc. IEEE 11th International Conference on Computer Vision (ICCV'2007), 2007, pp. 1-5.
[8] A. Gardel, A. Hernandez, R. Bravo, and R. Mateos, "Correction of Omnidirectional Camera Images using Reconfigurable Hardware," Proc. the 32nd Annual Conference of IEEE Industrial Electronics Society (IECON'06), 2006, pp. 3403-3407.
[9] Wang Wei, and Cao Zongsheng, "A Cylindrical Virtual Space Based Catadioptric Real-time Panorama Imaging System," Proc. IEEE Conference on Robotics, Automation and Mechatronics (RAMECH'2006), 2006, pp. 1-5.
[10] Lidong Chen, Maojun Zhang, Bin Wang, Zhihui Xiong, and Gang Cheng, "Real-time FPGA-based Panoramic Unrolling of High- Resolution Catadioptric Omnidirectional Images," Proc. International Conference on Measuring Technology and Mechatronics Automation(ICMTMA'2009), vol. 1, 2009, pp. 502-505.
[11] Altera, Corp., "A Flexible Architecture for Fisheye Correction in Automotive Real-View Cameras," White Paper, 2008

BIOGRAPHY

Seung-Ho Lee(Member)

1986 : BS degree in Electronic Engineering, Hanyang University
1989 : MS degree in Electronic Engineering, Hanyang University
1994 : Ph. D degree Electronic Engineering,Hanyang University
1994 ~ Present : Professor, Department of Electronic Engineering, Hanbat National University