OPTICAL REVIEW Vol. 20, No. 5 (2013) 367–373

A Distortion Correction Algorithm for Fish-Eye Panoramic Image of Master–Slave Visual Surveillance System

Chenglin ZUO*, Yu LIU, Yongle LI, Wei XU, and Maojun ZHANG

College of Information System and Management, National University of Defense Technology, Changsha, Hunan 410073, China

(Received March 23, 2013; Accepted June 24, 2013)

A master–slave visual surveillance system is composed of one fish-eye panoramic camera and one dynamic pan–tilt–zoom (PTZ) dome camera. In order to make PTZ dome camera zoom into all targets of interest in panoramic image, the fish-eye panoramic camera is fixed inclining towards the gravity direction, which may cause more obvious distortion. This paper proposed a novel method for the distortion correction of captured panoramic image based on the midpoint circle algorithm (MCA). The method uses incremental calculation of decision parameters to determine the pixel positions along a circle circumference, and both of the vertical and horizontal are rectilinearised. Experimental results show that our correction method based on MCA is efficient and effective. In particular, due to its low computational cost, our method can be applied on embedded camera platform without any extra hardware resources.

Keywords: master–slave visual surveillance system, distortion correction, midpoint circle algorithm

1. Introduction

Video surveillance devices are already prevalent in people's daily life and have been applied in many fields. They can be seen in roads, banks, railway stations, airports and other important places. With the development of surveillance devices, the field of view (FOV) that can be monitored is becoming larger and larger, while the demand of high resolution image about interesting scene is also increased. One solution of that problem is the master–slave visual surveillance system. It is composed of one fish-eye panoramic camera and one dynamic pan–tilt–zoom (PTZ) dome camera, as shown in Fig. 1. The fixed fish-eye panoramic camera acts as the master one to capture the scene in a wide FOV, while the PTZ dome camera acts as the slave one to zoom into the targets of interest under the direction of captured panoramic image.

For traditional surveillance camera, it is usually fixed horizontally at a not high place to get a large FOV in horizontal direction, which makes the captured image with more vertical distortion but less horizontal distortion. On the contrast, the master–slave visual surveillance system often needs to be fixed at a high place to get a large FOV both in vertical direction and horizontal direction. In this case, if the fish-eye panoramic camera shoots horizontally, it can not capture some nearby scenes, such as the scene under it, as shown in Fig. 2(a). Simultaneously, some captured scenes in the vertical FOV are useless, such as the sky, which also can not be captured by the PTZ dome camera. In order to make sure that the fish-eye panoramic camera can capture more useful scenes nearby and the PTZ dome camera can zoom into all targets of interest in captured panoramic image, the fish-eye panoramic camera should be fixed inclining towards the gravity direction, as shown in Fig. 2(b). However, this structure brings a serious problem: there are serious depth differences in captured image. Therefore, in addition to

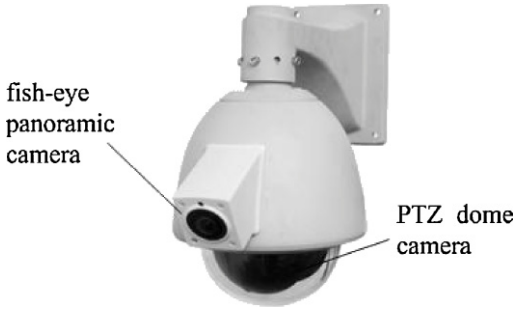


Fig. 1. (Color online) Practical master–slave visual surveillance system, which is composed of one fish-eye panoramic camera and one PTZ dome camera.

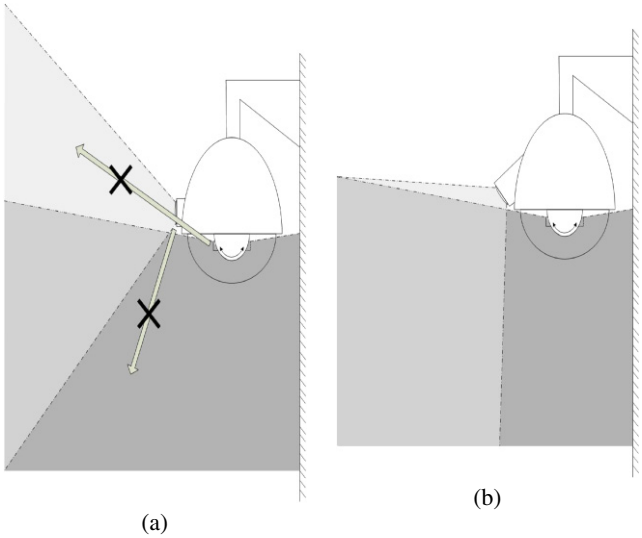


Fig. 2. (Color online) The difference of FOV between fish-eye panoramic camera and PTZ dome camera in the different structure. (a) The fish-eye panoramic camera shoots horizontally; (b) The fish-eye panoramic camera shoots inclining towards the gravity direction.

*E-mail address: zuochenglin3700@gmail.com

Fig. 3. (Color online) The fish-eye panoramic image captured by master–slave visual surveillance system, which has obvious distortion in horizontal direction.

serious vertical distortion, the captured fish-eye panoramic image may have more obvious horizontal distortion, which is unique to traditional fish-eye panoramic image. As shown in Fig. 3, the road and river have serious horizontal distortion. Hence, how to correct this particular type of distortion in captured fish-eye panoramic image becomes an urgent problem.

Most previous research focuses on constructing and figuring the internal reference model, which can express the mapping relationship between three-dimensional world and two-dimensional image.[1–3] Based on constructed internal reference model, the distortion image is mapped into three-dimensional spherical surface or parabolic surface. By using perspective projection method, the distortion can be corrected.

In this paper, a novel method based on the midpoint circle algorithm (MCA)[5] is proposed to correct the distortion in the fish-eye panoramic image captured by master–slave visual surveillance system. Firstly, by calculating the coordinate mapping between the column line image of the first corrected image and the arc line image of original fish-eye panoramic image based on MCA, the distortion in vertical direction is corrected. The first corrected image is called vertical correction image (VCI). Secondly, by means of the same principle, the coordinate mapping between the row line image of the second corrected image and the arc line image of VCI is calculated, and the distortion in horizontal direction is corrected too. The second corrected image is called horizontal vertical correction image (HVCI). Via adjusting the values of interrelated parameters, the distortion correction extent can be controlled in vertical and horizontal direction, respectively. In particular, due to its low computational cost, the proposed method can be applied on embedded camera platform in real time.

The remainder of the paper is organized as follows: Sect. 2 reviews related work. Section 3 describes the proposed distortion correction algorithm. In Sect. 4, experiments are implemented and the experimental results are shown. Finally, Sect. 5 concludes this article.

2. Related Work

Basu and Licardie[1] proposed the fish-eye transform (FET) model. This model was based on the observation that a fish-eye image has higher resolution in the foveal areas and lower resolution towards the peripheral areas. Devernay et al.[2] assumed the presence of straight lines in the scene. In his proposed method, distortion parameters are sought which lead to lines being imaged as straight in the corrected image. Kannala and Brandt[3] proposed a novel calibration method for fish-eye lens cameras that was based on viewing a planar calibration pattern. This method was proven suitable for different kinds of omnidirectional cameras as well as for conventional cameras. Wang et al.[4] presented a new model of camera lens distortion that utilized two angular parameters and two linear parameters. These parameters were used to determine the transform from ideal plane to real sensor array plane, which governs the lens distortion. Yu[6] proposed a lens geometric and photometric distortion correction method to obtain a high quality image. By using a simplified camera calibration technique, lens geometric coefficient can be estimated. Then, the photometric distortion was corrected by a nonlinear model fitting of a proposed photometric distortion model function. Ying et al.[7] used spherical perspective projection model to calibrate the fish-eye lenses. Based on the straight-line spherical perspective projection constraint, the mapping between a fisheye image and its corresponding spherical perspective image was determined. Once the mapping is obtained, the fish-eye lenses can be calibrated. However, since orthographic spherical perspective projection was used, their algorithms can only be applied to orthographic fish-eye cameras other than equidistant fish-eye cameras.[8,9] Li et al.[10] presented a embedded real-time fish-eye image distortion correction algorithm, which can be applied in IP network camera. However, this algorithm only aimed to correct the distortion in vertical direction. Moreover, methods that adapted the projection to content in the scene were also presented.[11–13] However, these methods requires human intervention, and the corrected image has to be cropped.

(a)

(b)

Fig. 4. (Color online) (a) The panoramic image captured by conventional fish-eye panoramic camera that shoots horizontally. The buildings and traffic signal poles on the both sides of the image all have vertical distortion more or less; (b) the marked image, in which the distortion has been marked.

Nevertheless, few methods were proposed to correct the particular type of distortion in the fish-eye panoramic image captured by master–slave visual surveillance system. This paper draws inspiration from MCA and applies this algorithm to correct the distortion.

3. Distortion Correction

Figure 4(a) shows the image captured by conventional fish-eye panoramic camera, which shoots horizontally. The buildings and traffic signal poles on the both sides of the image all have obvious vertical distortion, which are marked in Fig. 4(b), while almost no distortion occurs in horizontal direction. This type of distortion can be corrected easily with existing algorithms. However, for the particular type of distortion in horizontal direction shown in Fig. 3, existing algorithms are ineffective. So, method that can correct this horizontal distortion is needed.

MCA is to determine the pixel positions along a circle circumference based on incremental calculation of decision parameters. Compared with the Cartesian equation involving square root calculations and the polar coordinates parametric equations containing trigonometric calculations, MCA involves only simple integer operations.[5] As mentioned in Strand,[14] a straight line in world space can be projected to a corresponding circle on the fish-eye image plane. It means the coordinate mapping process between corrected image and fish-eye panoramic image can be calculated directly based on MCA.

The whole process of proposed distortion correction algorithm can be divided into two steps: Firstly, correct the distortion of fish-eye panoramic image in vertical direction; Secondly, based on the corrected image, correct its distortion in horizontal direction.

3.1 *Vertical distortion correction*

Figures 5(a) and 5(b) show the structure of captured fish-eye panoramic image and VCI in vertical direction, whose width and height both are W and H, respectively.

The first column line image EE′ in VCI is corresponding to the first arc line image ACA′ in the fish-eye panoramic image. For an arbitrary point P on EE′, there is a corresponding point P′ on ACA′. Since the distortion is corrected only in vertical direction, ordinate value of point P is the same with its corresponding point P′. In fish-eye panoramic image, OA does not project to VCI, so its length L is called vertical distortion redundancy length (VDRL).

For an arbitrary column line image FF′ in VCI, assume its corresponding arc line image in fish-eye panoramic image is BDB′. The point $F(X_f, Y_f)$ on FF′ has its corresponding point $B(X_b, Y_b)$ on BDB′. So, following equation is derived:

$$X_b = L + X_f \cdot \frac{W - 2L}{W}, \tag{1}$$

$$Y_b = Y_f = 0. \tag{2}$$

Based on MCA, the radius of the circle that arc line image BDB′ places is derived as follows:

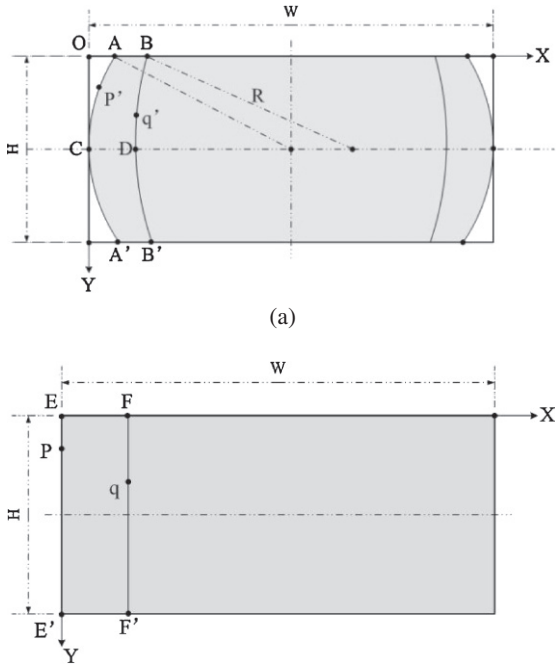


Fig. 5. (a) The structure of captured fish-eye panoramic image in vertical direction; (b) The structure of VCI in vertical direction.

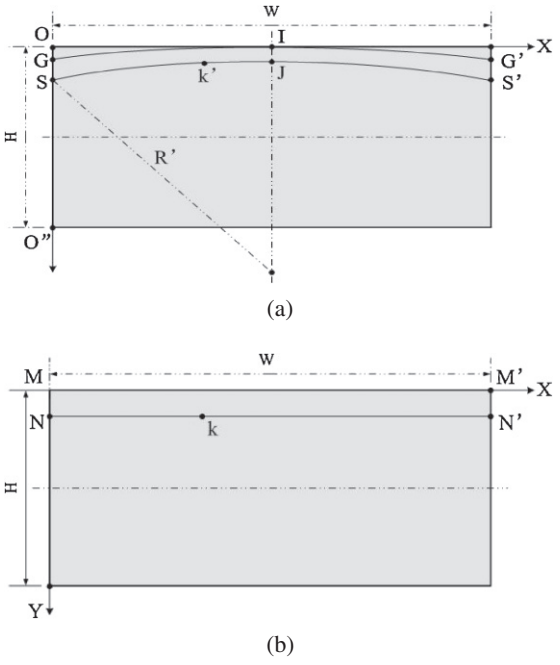


Fig. 6. (a) The structure of VCI in horizontal direction; (b) The structure of HVCI in horizontal direction.

$$R = \frac{(H/2)^2 + X_b^2 - X_f^2}{2 \cdot (X_b - X_f)} - X_f \quad \left(X_f \le \frac{W}{2}\right), \tag{3}$$

$$R = \frac{(H/2)^2 + (W - X_b)^2 - (W - X_f)^2}{2 \cdot (X_f - X_b)} - (W - X_f) \quad \left(\frac{W}{2} < X_f \le W\right). \tag{4}$$

With regard to an arbitrary point $q(X_q, Y_q)$ on FF′, assume its corresponding point on BDB′ is $q'(X_0, Y_0)$. So, following equations are derived:

$$X_0 = R + X_q - \sqrt{R^2 - \left(\frac{H}{2} - Y_0\right)^2} \quad \left(X_q \le \frac{W}{2}\right), \tag{5}$$

$$X_0 = R + X_q + \sqrt{R^2 - \left(\frac{H}{2} - Y_0\right)^2} \quad \left(\frac{W}{2} < X_q \le W\right), \tag{6}$$

$$Y_0 = Y_q. \tag{7}$$

Based on above derived equations, all points in VCI find corresponding points in fish-eye panoramic image after traversing entire VCI, so the distortion in vertical direction can be corrected. In particular, by adjusting the value of VDRL, the extent of distortion correction can be controlled.

3.2 *Horizontal distortion correction*

The distortion correction in horizontal direction uses the same method. Figures 6(a) and 6(b) show the structure of VCI and HVCI in horizontal direction, respectively.

In HVCI, an arbitrary row line image has its corresponding arc line image in the VCI, such as the first row line image MM′ and its corresponding first arc line image GIG′. As VDRL in fish-eye panoramic image, there also is horizontal distortion redundancy length (HDRL) L' in VCI, which is the length of OG. Because the distortion is corrected only in horizontal direction, abscissa value of a point in HVCI is the same with its corresponding point in VCI.

For an arbitrary row line image NN′ in HVCI, assume its corresponding arc line image in VCI is SJS′. A point $k(X_k, Y_k)$ on NN′ has its corresponding point $k'(X_1, Y_1)$ on SJS′. So, following equations are derived:

$$R' = \frac{\left(\frac{W}{2}\right)^2 + \left(L' + Y_k \cdot \frac{H - 2L'}{H}\right)^2 - Y_k^2}{2 \cdot \left(L' + Y_k \cdot \frac{H - 2L'}{H} - Y_k\right)} - Y_k \quad \left(Y_k \le \frac{H}{2}\right), \tag{8}$$

$$R' = \frac{\left(\frac{W}{2}\right)^2 + \left[H - \left(L' + Y_k \cdot \frac{H - 2L'}{H}\right)\right]^2 - (H - Y_k)^2}{2 \cdot \left[Y - \left(L' + Y_k \cdot \frac{H - 2L'}{H}\right)\right]} - (H - Y_k) \quad \left(\frac{H}{2} < Y_k \le H\right). \tag{9}$$

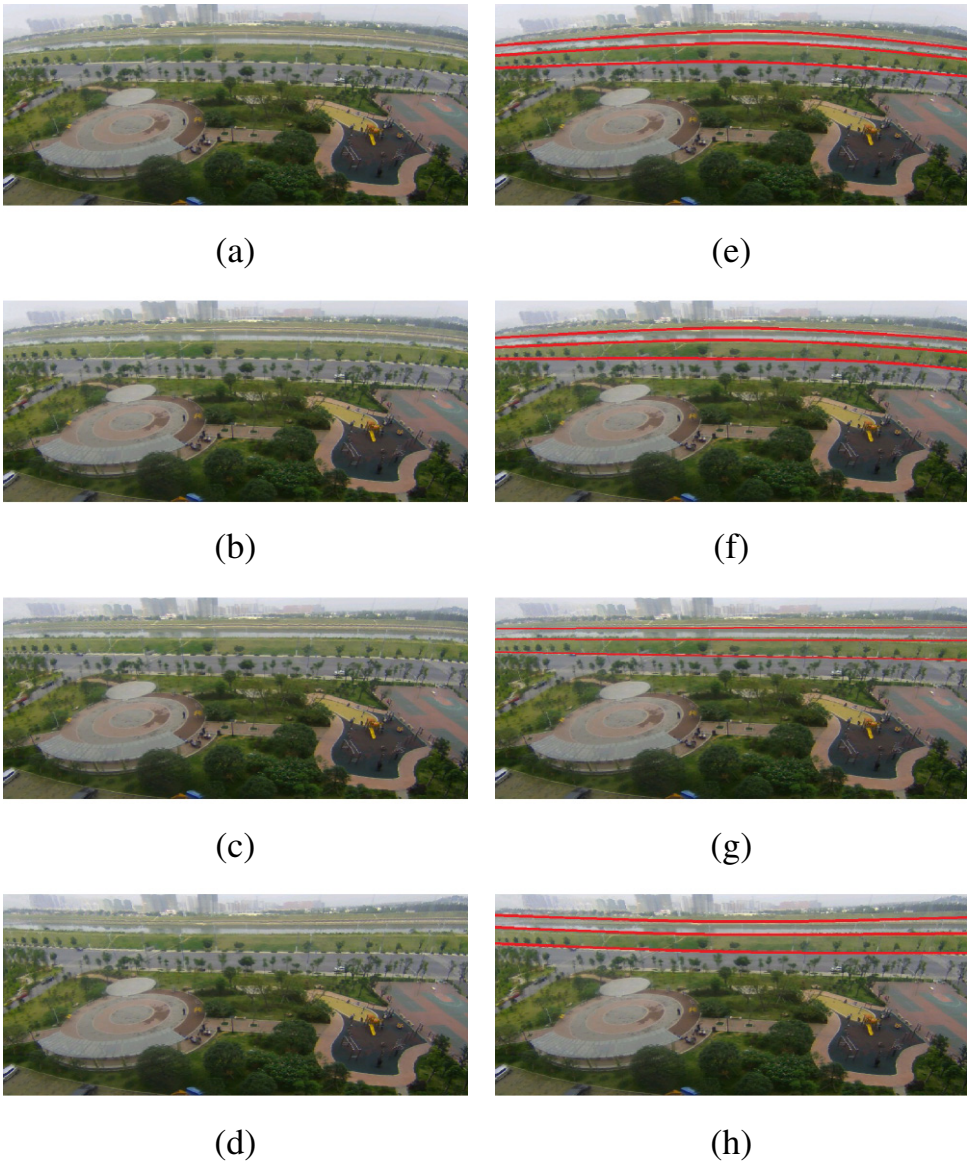


Fig. 7. (Color online) Different images after distortion correction by using proposed method with different HDRL values. (a)–(d) show the corrected images when HDRL values are 60, 90, 120, and 140, respectively. (e)–(h) show the marked images after distortion correction with above HDRL values, in which the distortion corrected results are marked.

So, the coordinate values of point k' can be derived as follows:

$$Y_1 = R' + Y_k - \sqrt{(R')^2 - \left(\frac{W}{2} - X_1\right)^2} \quad \left(Y_k \le \frac{H}{2}\right), \tag{10}$$

$$Y_1 = R' + Y_k + \sqrt{(R')^2 - \left(\frac{W}{2} - X_1\right)^2} \quad \left(\frac{H}{2} < Y_k \le H\right), \tag{11}$$

$$X_1 = X_k. \tag{12}$$

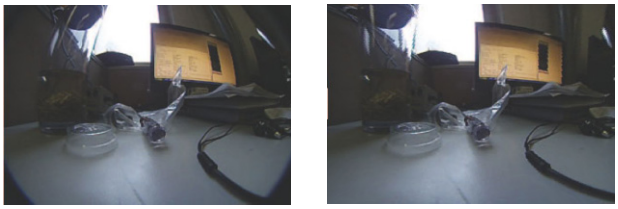
Similarly, after traversing entire HVCI, all points find corresponding points in VCI, so the distortion in horizontal direction can be corrected. In particular, by adjusting the value of HDRL, the extent of distortion correction can also be controlled.

After correcting the distortion in vertical and horizontal directions respectively, this particular type of distortion in the fish-eye panoramic image captured by master–slave visual surveillance system can be corrected effectively. It is worth mentioning that this method has a very fast correction speed due to its low computational cost.

4. Experiments and Results

In order to demonstrate proposed distortion correction algorithm feasible, practical experiments are implemented. Figure 3 shows the fish-eye panoramic image captured by master–slave visual surveillance system, whose size is 2560×896. Figures 7(a)–7(d) show the images after

Fig. 8. (Color online) The distortion corrected result of Fig. 4(a) by using proposed method.



(a) (b)

Fig. 9. (Color online) (a) An indoor image captured by conventional fish-eye panoramic camera; (b) The distortion corrected result by using proposed method.

Table 1. Comparison of time consuming.

Image resolution	Time consuming of Ying's algorithm[7] (ms)	Time consuming of proposed algorithm (ms)
640 × 480	487	61
1280 × 720	1434	263
1920 × 1080	3205	326
2560 × 896	3695	471

Table 2. Comparison of calculation resource consuming.

Image resolution	CPU utilization of Ying's algorithm[7] (%)	Memory utilization of Ying's algorithm[7] (kb)	CPU utilization of proposed algorithm (%)	Memory utilization of proposed algorithm (kb)
640 × 480	20	3284	2	1516
1280 × 720	25	9524	6	1992
1920 × 1080	25	19604	9	2336
2560 × 896	25	25560	16	2692

distortion correction by using our proposed method with different HDRL values, and the correction results are marked in Figs. 7(e)–7(h), respectively.

In Figs. 7(a) and 7(b), HDRL values are 60 and 90, respectively. The distortion in Fig. 7(a) has been corrected a little, but not visibly. Although the distortion correction result in Fig. 7(b) is slightly better than that in Fig. 7(a), both are not ideal. In Fig. 7(c), HDRL value is 120, and the distortion in horizontal direction is almost corrected. The result shows that this value setting can obtain desired distortion correction effect. In Fig. 7(d), HDRL value is 140. Obviously, this value setting is overdone and causes over-distortion. It is worth mentioning that the ideal HDRL value is different for different fish-eye panoramic image.

Moreover, this method can also be used to correct the distortion of conventional fish-eye panoramic image. Figure 8 shows the corrected result of Fig. 4(a). Because it almost has no horizontal distortion in conventional image, the distortion correction is only done in vertical direction. So, it just needs to adjust the VDRL value, and its value is 70. Figure 9(a) shows another indoor image, and its corrected result is shown in Fig. 9(b).

Table 1 shows the comparison of time consuming on dealing with different resolution images with the algorithm proposed by Ying et al.[7] The comparison was conducted with single-threaded implementation on a 3.10 GHz Intel Core i3-2100 CPU and 2.00 GB RAM computer by using Microsoft Visual Studio 2008 software. The result shows that the proposed algorithm has much less time consuming than Ying's algorithm on dealing with different resolution images. In addition, the calculation resource consuming of Ying's algorithm and proposed algorithm on dealing with different resolution images in the comparison, such as the utilization of CPU and memory, is also shown in Table 2. Obviously, due to its low computational cost, much less resource is required by the proposed algorithm.

Finally, we analyze the computation complexity of proposed algorithm in Table 3. On the embedded camera platform, the simple addition, subtraction, multiplication and division operations are quite easy to implement without much time consuming. However, the square root and trigonometric operations are hard to implement because of their complexity. Fortunately, there is only one square root operation in our algorithm, but no trigonometric operation.

Table 3. Computation complexity of proposed algorithm.

Operation	Time
Addition	3
Subtraction	7
Multiplication	6
Division	2
Square root	1
Trigonometric	0

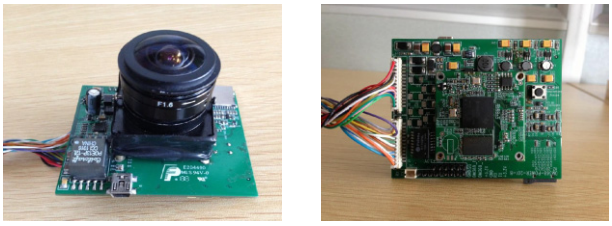


Fig. 10. (Color online) The structure of the designed embedded IP camera.

Table 4. Performance of proposed algorithm in embedded IP camera.

Image resolution	Time consuming (ms)
640 × 480	35
1280 × 720	70
1920 × 1080	120

Generally, there are many methods to calculate the square root with less cost on the embedded platform, such as Newton-Raphson method,[15] SRT-redundant method,[16] non-redundant method,[17] and some other methods.[18,19] Because there is only one square root operation in our algorithm, the time consuming is still not much. More importantly, all the operations of our algorithm are on integer numbers but not floating point numbers, which means faster calculation speed on the embedded platform.

Based on above analysis, we design a simple embedded IP camera, which can meet the requirement of our algorithm on time consuming. TI TMS320DM8168 is employed as the processor, which has 1200 MHz clock rate and 2400 MIPS computing speed. In addition, it has three high definition video image co-processors (HDVICP) engines to support a range of encode, decode and other video quality operations, and is capable of 1080 p and 60 fps H.264 video processing, etc. The image sensor is Aptina Micron MT9P031 which has a 1/2.5 in. 5 Mp digital sensor. It has column and row binning modes to improve image quality when resizing image size without changing FOV. It is suitable for the application of high resolution network cameras with wide FOV cameras. The lens is Daiwon DW1634D, whose horizontal angle of view is 172.2° and vertical angle of view is 126.2°. Figure 10 shows the designed IP camera. Our distortion correction algorithm is then embedded into the IP camera, and the practical performance is shown in Table 4. The results show that our method is effective to handle with embedded IP camera's fish-eye distortion problem in real-time when the resolution of captured image is 640 × 480. However, when the resolution of captured image becomes larger, the algorithm appeared to be inadequate.

5. Conclusion

In this paper, we propose a novel algorithm to correct the particular type of distortion in fish-eye panoramic image captured by master–slave visual surveillance system based on MCA. Firstly, by calculating the coordinate mapping between fish-eye panoramic image and VCI, the distortion in vertical direction is corrected. Secondly, based on the same principle, the distortion in horizontal direction is corrected. By adjusting the values of VDRL and HDRL, the distortion correction extent can be controlled. As experimental results show, the proposed algorithm is efficient and effective. More importantly, the proposed algorithm can be applied on embedded camera platform without any extra hardware resources due to its low computational cost.

Acknowledgements

This research was partially supported by National Natural Science Foundation (NSFC) of China under project Nos. 61175006 and 61175015.

References

1) A. Basu and S. Licardie: Pattern Recognit. Lett. **16** (1995) 433.
2) F. Devernay and O. Faugeras: Mach. Vision Appl. **13** (2001) 14.
3) J. Kannala and S. S. Brandt: IEEE Trans. Pattern Anal. Mach. Intell. **28** (2006) 1335.
4) J. Wang, F. Shi, J. Zhang, and Y. Liu: Pattern Recognition **41** (2008) 607.
5) D. D. Hearn and M. P. Baker: *Computer Graphics with OpenGL* (Pearson Education, Englewood Cliffs, NJ, 2004) 3rd ed., p. 103.
6) W. Yu: IEEE Trans. Consum. Electron. **49** (2003) 894.
7) X. Ying, Z. Hu, and H. Zha: Proc. ACCV, 2006, p. 61.
8) C. Hughes, R. McFeely, P. Denny, M. Glavin, and E. Jones: Image Vision Comput. **28** (2010) 538.
9) C. Hughes, P. Denny, M. Glavin, and E. Jones: IEEE Trans. Pattern Anal. Mach. Intell. **32** (2010) 2289.
10) Y. Li, M. Zhang, Y. Liu, and Z. Xiong: Proc. SMC, 2012, p. 2224.
11) R. Carroll, M. Agrawala, and A. Agarwala: ACM Trans. Graph. **28** (2009) 1.
12) J. Kopf, D. Lischinski, O. Deussen, D. Cohen-Or, and M. Cohen: Comput. Graph. Forum **28** (2009) 1083.
13) J. Wei, C. F. Li, S. M. Hu, R. R. Martin, and C. L. Tai: IEEE Trans. Visualization Comput. Graphics **18** (2012) 1771.
14) R. Strand and E. Hayman: Proc. BMVC, 2005, p. 1.
15) H. Kabuo, T. Taniguchi, A. Miyoshi, H. Yamashita, M. Urano, H. Edamatsu, and S. Kuninobu: IEEE Trans. Comput. **43** (1994) 43.
16) T. Lang and P. Montuschi: Proc. IEEE Symp. Computer Arithmetic, 1995, p. 124.
17) G. Knittel: Comput. Graph. **19** (1995) 261.
18) Y. Li and W. Chu: Proc. IEEE FCCM, 1997, p. 226.
19) R. M. Fosler: Microchip Technology Inc. TB040 (2000).