




On the design and implementation of a dual fisheye camera-based surveillance vision system

Raghad Al-Harasis¹ · Belal H. Sababha² 

Received: 21 January 2018 / Revised: 3 March 2019 / Accepted: 18 March 2019 /
Published online: 22 April 2019
© Springer Science+Business Media, LLC, part of Springer Nature 2019

Abstract

Image processing and computer vision have been a focus of researchers for decades in various application domains. This research is continuously rising with the rise of Artificial Intelligence in the fourth industrial revolution. One of the important digital image processing applications is to produce panorama images. The wide range of view a panorama image provides can be used in a variety range of applications which may include surveillance applications and remote robot operations. A panorama image is a combination of several individual natural looking images into a composite one to provide a wide field of view that may reach 360 degrees horizontally without any distortion. Wide-angle lenses provide a wide field of view, but using them alone does not necessarily make a panorama image. In this work the design and implementation of a wide-angle stereo vision system that suites many real-time applications is proposed. The system makes use of two wide-angle fisheye cameras where each camera covers around 170 degrees field of view. The horizontal angle between the cameras is 140 degrees. The cameras acquire the instantaneous overlapping images continuously and transmits them to a base station via a communication link. The base station calibrates, corrects, correlates and stitches the non-overlapping corrected images to a composite one. The resultant final image covers 310 degrees field of view. The system is of low computational complexity compared with previously implemented systems. It is tested on a laptop and on a standalone embedded computing device. The processing speed for the panorama image stitching including the correction of the fisheye barrel distortion on the laptop computer and the embedded computer is 11 fps, and 6 fps, respectively.

Keywords Barrel distortion correction · Real-time image processing · Panorama image stitching · Wide-angle view · Fisheye camera

✉ Belal H. Sababha
bsababha@ieee.org

¹ Electrical Engineering Department, Princess Sumaya University for Technology, Amman 11941, Jordan

² Computer Engineering Department, Princess Sumaya University for Technology, Amman 11941, Jordan

1 Introduction

Robots and unmanned systems are increasingly rising in the new era of artificial intelligence and fourth industrial revolution [2, 3, 21, 24, 26–34]. The advancements in digital image processing improved the development and the intelligence of robots and made them more popular [2, 22]. Computer vision First Person View (FPV) systems is one of the applications that enables remotely control robots. FPV vision systems are useful for indoor and outdoor missions and provide different type of useful feedback. Conventional visual FPV feedback systems utilizing conventional cameras usually cover less than 100 degrees field of view. Cameras could be equipped with fisheye lenses to widen the field of view. The use of fisheye lenses induces a type of optical distortion to the captured images. This type of optical distortion is called barrel distortion and occurs due to the optical design of the lens.

In this paper, the design and implementation of a wide-angle stereo vision system based on two fisheye cameras with low computational cost is presented. The system is suitable for many real-time applications. It is capable of providing 11 fps (frames per second). During the work, several commercial systems and existing algorithms were studied and considered. The system proposed in this paper compared with existing systems in the literature is different. The proposed work presents a stereo vision FPV system that covers 310 degrees field of view and outputs 11 fps using a laptop and 6 fps using a stand-alone embedded computing device. Up to the authors' knowledge no reported similar systems in the literature that process the fisheye effect from two cameras and produces a stereo image achieves this rate. A comparison between various image processing and computer vision algorithms on a resource constraint embedded device was also conducted. Although some commercial similar systems may exist, these did not present scientific backgrounds or design details, hence, this work contributes to the body of knowledge through scientifically presenting and describing the proposed system.

The rest of the paper is organized as follows. In Section II, related literature is reviewed. Section III covers the design and implementation of the proposed system. Section IV presents experimental results and performance analysis. Finally, Section V concludes the paper and highlights future work.

2 Literature review

Utilizing two cameras to build binocular vision systems has several advantages. One main advantage is the wider field of view when compared to monocular systems. In this section similar existing systems and previous work done in this field are overviewed.

To obtain a wide field of view, some systems are implemented using a fixed wide-angle camera mounted on a moving object. The distorted images are then corrected using various kinds of algorithms to provide natural looking images. The obtained rectified images are passed to detection tools and processes to be applied for different kinds of purposes and applications such as in advanced driver assistance systems (ADAS) to detect the host vehicle or to track objects [1, 7, 14, 37, 42]. The wide field of view is achieved in other systems by using one conventional camera with a convex mirror. The output image is an annulus image. This image is then converted to a rectangular form to obtain a panorama image. The resultant image covers 360 degrees field of view [7].

Other systems are implemented using multiple conventional cameras mounted on a moving tripod. The purpose is to capture a series of successive non-instantaneous overlapping images. Then stitch them together using a state of the shelf stitchers to cover 360 degrees field of view [25, 36, 38].

The systems presented in [5, 11, 35, 39] are stereo vision systems implemented using conventional cameras. These systems are used for different kinds of applications such as 3D point reconstruction and object localization and mapping for autonomous mobile robots.

The work presented by Zhang *et al.* focuses on stereo matching of fisheye images without distortion correction. According to the authors, the matching algorithm the researchers proposed is expected to meet the need of stereo matching. In their work, where the optical axis of the two fisheye cameras are perpendicular, the researchers concentrate on finding the matching points between two fisheye spherical images for 3D applications. The researchers proposed an algorithm based on two algorithms called Maximally Stable Extremal Regions (MSER) and Affine Scale Invariant Feature Transform (ASIFT). In the reported results the researchers reported best time needed to find 37 matching points to be 3.566 s. This time is based on running the algorithm on a personal computer featuring a dual core processor running at 2.2 GHz and 5GB of memory [41]. The reported processing time did not include the cost of generating a stereo image out of the two captured fisheye images.

In [8, 19, 20] stereo vision systems are implemented to construct the position of vehicles. The systems used two cameras equipped with fisheye lenses. The distorted images are then calibrated and rectified. The stereo matching algorithm is used to compute the 3D points of the rectified images.

Wei *et al.* presented a robust scheme for fisheye video correction. The proposed method solved the time-varying problem which increases the distortion while the objects are moving. Six distinct but related correction criteria are used. The paper compared the corrected image with other methods. The proposed algorithm was tested on a PC with Intel 2.5 GHz Core 2 CPU and 2 GB memory. This approach supports interactive video processing with a runtime of around 0.42 s per frame (*i.e.* 2.3 fps) [40].

In [4] a real-time fisheye lens distortion correction system is designed and implemented using a FPGA-based camera. The work generated a complex image processing application and optimization on the source level on the original code to exploit the memory architecture in the FPGA. The performance comparison between the Core 2 Quad and the FPGA implementation was shown. In the Core 2 quad software the fps rate was 5.26 but in the FPGA it was 22 fps.

In [15] the system has been implemented using two back-to-back fisheye cameras to capture 360degrees FoV images. The cameras are exploited by a smartphone system. MediaTek's smartphone with 5 megapixels and 182 degrees FoV is used to acquire the forms. The acquired images are wrapped and blended into a 4 k panorama image. The presented paper has designed and implemented a wrapping and projection techniques to eliminate the fisheye lens distortion. Moreover, a memory efficient blending technique has been introduced to blend the two fisheye images together. The system achieved 7.8 Mpixel/s, *i.e.* 1 K resolution with 15 fps by using 2.5 GHz Octa-core CPU, and power VR GPU.

The work presented by Ho *et al.* has introduced a novel method to align images generated by a dual-fisheye camera by employing interpolation grids based on rigid Moving Least Squares (MLS) to produce seamless stitching panorama images. Besides, they have reduced the jitter in videos that are generated by image-based stitching algorithms by incorporating a new temporal-coherent algorithm to maintain smooth-to-smooth frame transition. The system has been implemented using C++ and Matlab software, and the Rigid MLS has accelerated by GPU. The presented method achieved 360×180 degrees FoV seamless stitching panorama images [13].

In [12] proposed a stereo system that provides a panorama image with 360×65 degree FoV in horizontal and vertical directions. Also, the presented work addressed some constraints in stereo systems like camera modeling, fisheye intrinsic calibration, stereo self-calibration, and depth estimation. A bundle adjustment based approach and markerless stereo self-calibration method are used to optimize and to reduce the number of calibration parameters. The system introduced a Region of Interest (ROI) extraction modules to obtain distortion-free pinhole images from partial regions in fisheye images. The system was implemented using two mvBlueFox-M1c202bG2 cameras with image resolutions 1280×960 , 235degree FoV and 1280×1024 , 245 degree FoV.

3 Proposed system

The proposed system takes into consideration noisy and low feature images. It is also a low computational complexity system. The proposed system is implementable on a laptop computer and on a resource constraint embedded computing device such as NI myRio-1900 [17]. The proposed system's performance suits many applications with real-time requirements. The block diagram in Fig. 1 illustrates a high-level presentation of the wide-angle stereo vision system presented in this work.

3.1 System hardware

The proposed system utilizes two wide-angle fisheye cameras. The cameras are low-cost car reverse backup rear view mini color Charge-Coupled Device (CCD) cameras. Each one covers 170 degrees field of view with resolution $756(H) \times 720(V)$ pixels.

The dimensions of each camera is $18 \text{ mm} \times 18 \text{ mm} \times 22 \text{ mm}$ and powered by a 12 V DC / 200 mA power source. The horizontal angle between the two cameras is 140 degrees as illustrated in Fig. 2. The cameras capture the overlapping images and transmit them via the communication links to the computing system. In this paper the computing system calibrates, corrects and stitches the images to produce a live panorama image suitable for many real-time applications.

The computer used for system development is an Intel® Core™ i7-2670QM CPU @ 2.20 GHz, with 8.00 GB installed memory laptop and Intel® HD Graphics 3000 graphics card. Moreover, the system is tested on a National Instruments (NI) myRIO-1900 embedded

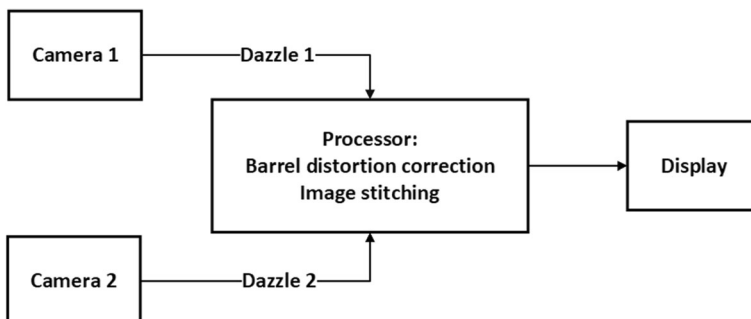


Fig. 1 The block diagram of the proposed system

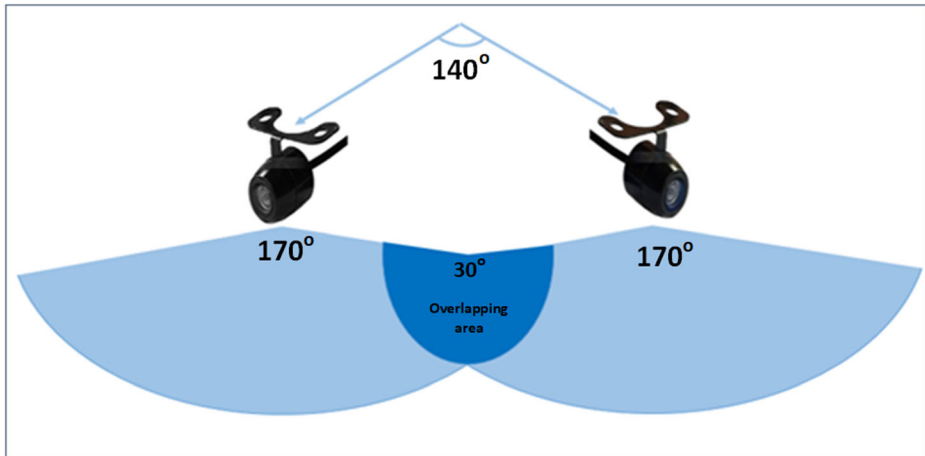


Fig. 2 The horizontal angle between the cameras

computing device. NI myRIO-1900 is a portable Reconfigurable I/O (RIO) device with dimensions of 136.6 mm × 86 mm and weighs 193 g, shown in Fig. 3.

The NI myRIO-1900 Processor is Xilinx Z-7010 operating at 667 MHz. The processor features 2 cores, 512 MB of Nonvolatile memory, 256 MB DDR3 memory with 533 MHz clock frequency and a 16-bit data bus. The used NI myRIO-1900 features two USB ports that will be used to connect the two cameras. Figure 4 illustrates the hardware connection using the laptop and NI myRIO-1900.

3.2 System software

In the proposed system, the horizontal angle between the two fisheye cameras is experimentally determined by trying to reduce the overlapping area as much as possible yet try to cover the widest field of view. Before starting the distortion correction process, a calibration process is carried out to determine the intrinsic and extrinsic parameters of the cameras. After the calibration process, a barrel distortion correction algorithm is implemented. The barrel distortion correction algorithm shall consider the execution time constraints to enable the system to

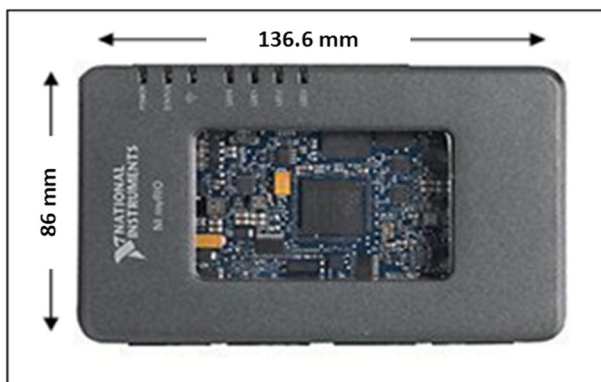


Fig. 3 NI myRIO-1900

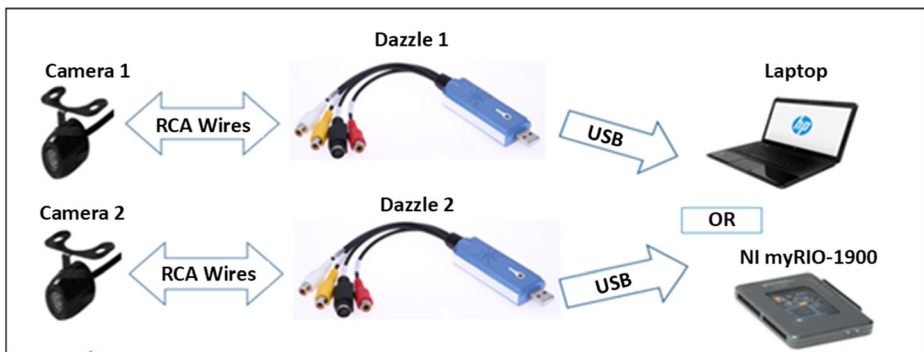


Fig. 4 System hardware connection

instantaneously handle the two images coming from the two fisheye cameras. The system undistorts every two instantaneous images and then combines them together to obtain a natural looking wide range view that covers 310 degrees field of view. Figure 5 illustrates the proposed system software design to generate a live panorama image from the images captured by the two fisheye cameras. The system starts by calibrating the fisheye lenses. Then the barrel distortion is removed from the acquired overlapped images. After that, the corrected overlapped images are correlated to measure the similarity between the images. Finally, the non-overlapped images are stitched to a composite image.

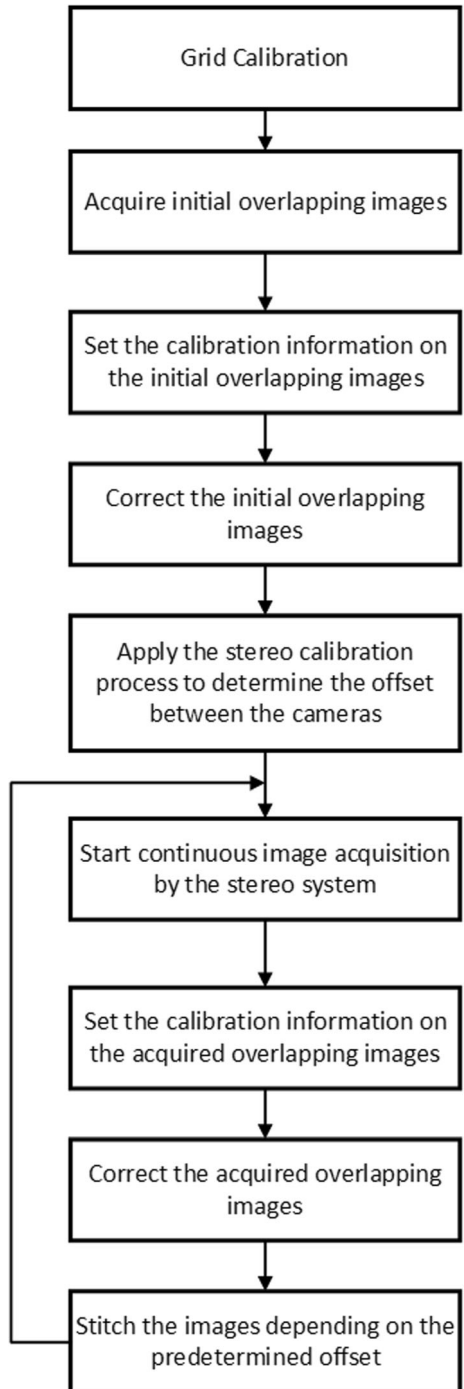
3.2.1 Fisheye lens calibration

The calibration process is carried out to obtain the intrinsic and extrinsic parameters of the camera [6, 23]. These parameters are used for image correction, distance determination, stereo matching and accurate measurements. Intrinsic calibration maps the camera and the image coordinates to get the camera properties such as focal length, principal point, and lens distortion. Figure 6 illustrates the intrinsic parameters of the camera. Intrinsic parameters determine the projection of a 3D object into a 2D image, therefore determines the relative position of the camera relative to the object's coordinates. Furthermore, extrinsic parameters define the location and the orientation of the camera recovered by rotation and translation matrices. In multi-camera systems, the extrinsic parameters describe the relative position and attitude of each camera to the other.

The fisheye lens calibration in this work is performed by acquiring a predetermined circular grid via the wide-angle fisheye camera. The camera is equipped with a 170-degree field of view fisheye lens. The same grid is used later as a reference template for the system. The grid consists of 9 equal circles distributed regularly on an A4 sheet. The radius of each circle equals 2.5 cm. The horizontal and the vertical distances (dx , dy) between two adjacent centers equals 10 cm and 7 cm respectively as shown in Fig. 7.

The grid is converted into an 8-bit greyscale image, then thresholding is conducted to produce a binary image. As shown in Fig. 8, the detected dark particles are the circles, boarders, and some other noise distributed in different positions in the image. To remove the unwanted boarders, a particles filter function is applied. This filter removes or keeps the particles depending on predetermined parameters such as particle area. As a result, the particle filter keeps the circles of the grid and removes the boarders and noise. The target point circles' area ranges from 100 to 10,000 square pixels. The filter is set to remove any particles outside this

Fig. 5 Software design of the proposed system



range and produce a clean grid image. Figure 9a and b illustrate the calibration grid after applying the particles filter and computing the target points in pixels and real-world coordinates.

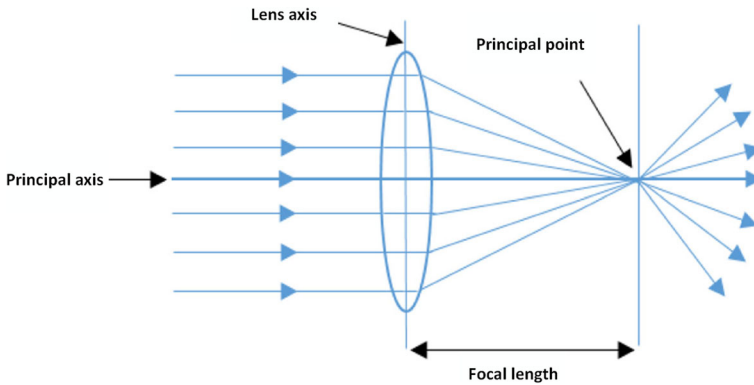


Fig. 6 The intrinsic parameters of the camera

The previously computed reference points are then passed to a distortion learning calibration function. This function learns the distortion model of the camera and the lens setup. Generally, learning the image makes a map between the pixels' coordinates and the real-world coordinates. After the learning process is complete, the function is applied on all acquired images by the cameras.

3.2.2 Barrel distortion correction

Barrel distortion follows the quadratic form of degree two. This effect appears as curved lines on the captured images. To fit this curve and the radius to a linear shape, different interpolation methods may be used. However, the interpolation uses known data to estimate the value of an unknown point. Hence, to correct the distorted image pixels, the pixel coordinates and their new intensity are required to be projected on the destination image. Pixel coordinates are obtained from the calibration process and the new value of these distorted pixels are obtained from the interpolation method. Some of the widely used interpolation methods that are used to correct barrel distortion are the nearest linear neighbor interpolation and bilinear interpolation [16]. Nearest neighbor interpolation method estimates the value of the interpolated point depending on the closest pixel value. In the bilinear interpolation method, the interpolated point is estimated by the average weight of 2×2 nearest neighborhood pixels. However, in the

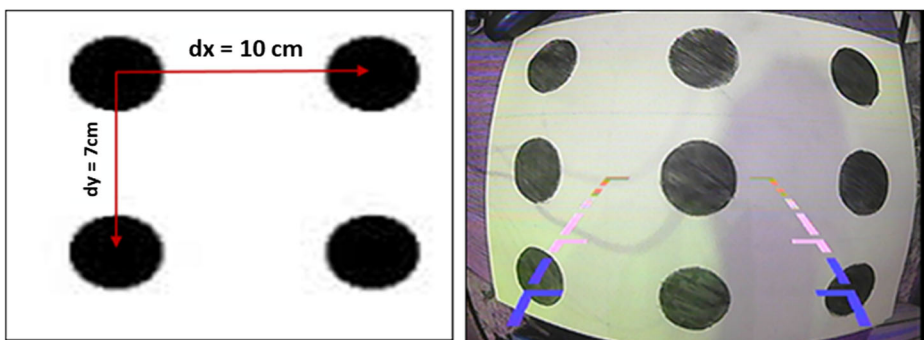


Fig. 7 Calibration grid parameters (a) Grid parameter description and the proposed calibration grid taken by 170 degrees cameras

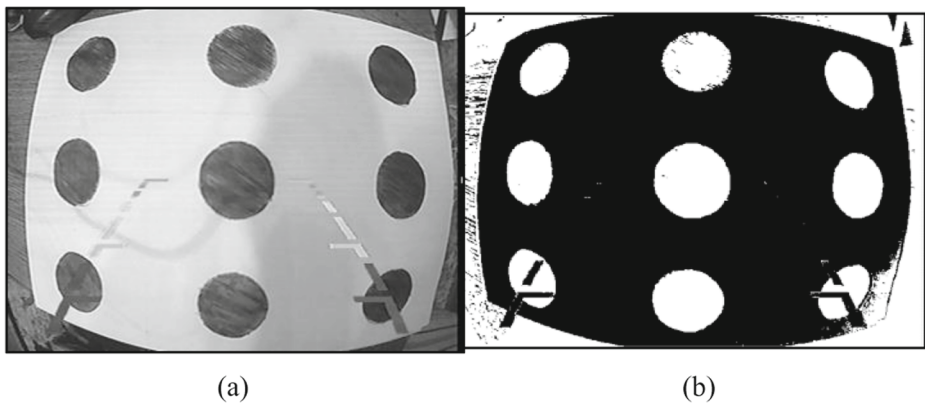


Fig. 8 Illustration for the calibration grid before and after applying thresholding technique **a** A greyscale threshold calibration grid and **b** The calibration grid after applying thresholding technique

bilinear interpolation method the resultant image appears smoother than the image that is obtained from the nearest neighbor method. But the execution time of the bilinear method is longer than that of the nearest neighbor method. Figure 10a and b show a simple representation of the nearest neighbor interpolation and bilinear interpolation methods, respectively. Because the nearest neighbor interpolation method is faster than the other methods, it is adapted in this work. Figure 11a, b, c, and d show the overlapping images before and after barrel distortion correction, respectively, by using the nearest neighbor interpolation method.

3.2.3 System calibration

In this step, the intersection and similarities between the two captured fisheye images are measured. This process consists of two main parts: (a) finding the overlapping particles in the two corrected images. (b) the found overlapping particles are then processed to figure out the overlapping area and the relative position of the first camera to the second camera.

Particle detection and analysis The angle between the two cameras is experimentally determined to cover the largest possible field of view and at the same time to maintain the mid area from

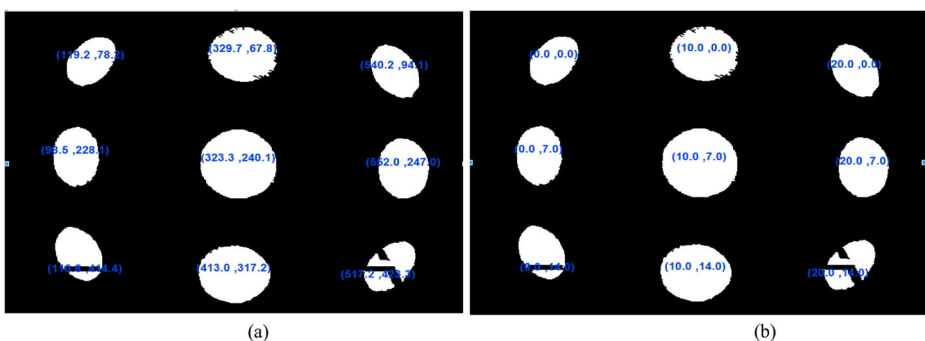


Fig. 9 The detected target points after applying the particles filter function and converting circular dots to target points **a** The target center of mass (x, y) of the circles in pixels and **b** The target center of mass (x, y) of the circles in real-world coordinates

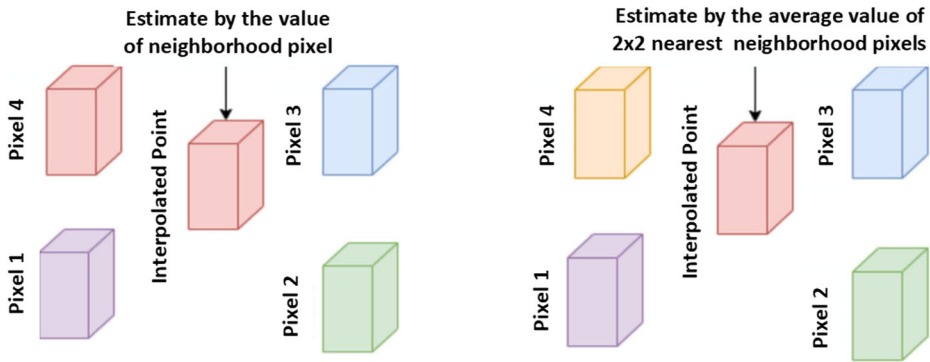


Fig. 10 Representation of interpolation methods **a** Nearest neighbor interpolation method and **b** Bilinear interpolation method

getting lost or seriously affected by losing details. Because the barrel distortion correction may produce loss in parts of the image near the borders of each image, in this work, the best optimal angle was experimentally found to be 140 degrees horizontally. After obtaining the undistorted images from the cameras, the resulting images are processed as follows:

- **Auto Threshold:** Thresholding an image is usually implemented at the first step of the machine vision application depending on the image type. This function separates the image into a particles' region and a background region. It converts the grey level image into a binary image depending on the objects that the function is looking for. It uses different kinds of techniques to separate the pixels' intensities such as clustering, entropy, inter variance, metric, and moment techniques. In this work, the clustering threshold technique is used because it can threshold the image into multi-classes. It sorts the histogram of the image into a discrete number of classes to obtain the center of mass of each class. The histogram represents the number of pixels with the same color in a fixed

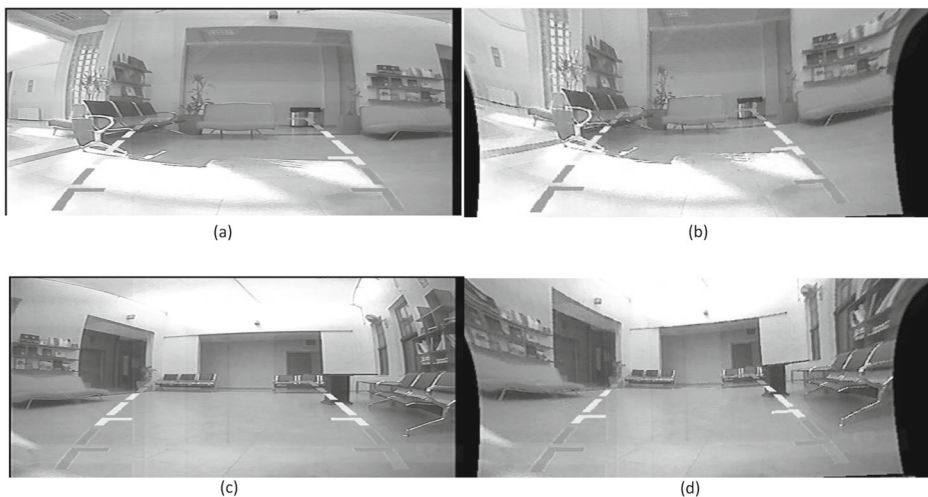


Fig. 11 **a** Left overlapping distorted image **b** Left overlapping corrected image **c** Right overlapping distorted image and **d** Right overlapping corrected image

list of color range. Figure 12, represents the cropped corrected overlapping images before and after applying the clustering technique.

- **Basic Morphology:** This function isolates the objects in the image into non-overlapping regions based on the topography surface of the image. It depends on the relative ordering of pixel values not on their numerical values. Furthermore, the auto threshold technique separates the particles from the background. But basic morphology makes the particles in the image clearer and unattached. It separates the adjacent particles in the image depending on a small template that probes the image at all possible locations. This template is called the structuring element. It is a matrix of zeros and ones that fits or hits the input image. If all elements in the matrix are ones then it fits the corresponding pixels under the structuring element. If the matrix elements are zeros and ones, it hits the pixels under the structuring elements. Figure 13 describes the operation of the structuring elements on the image. Where the grey and white blocks represent '1' and '0' pixel values respectively. When probing structuring element #1 on locations A, B and C, it fits just location A and hits location A, B and C. And when probing structuring element #2 on the same locations, it fits and hits A and B. various techniques are used to separate the objects. The reason of using this function is to increase the gap between the particles and remove the thin edges. This makes the particle detection process easier and more accurate. The open objects technique with a 3×3 structuring element is used in this work. Figure 14 shows the image before and after applying the basic morphology on the half overlapping corrected images.
- **Particle removal:** After isolating the objects in non-overlapping areas, some small particles persist in the image. These particles may cause inaccurate correlation measurements, and

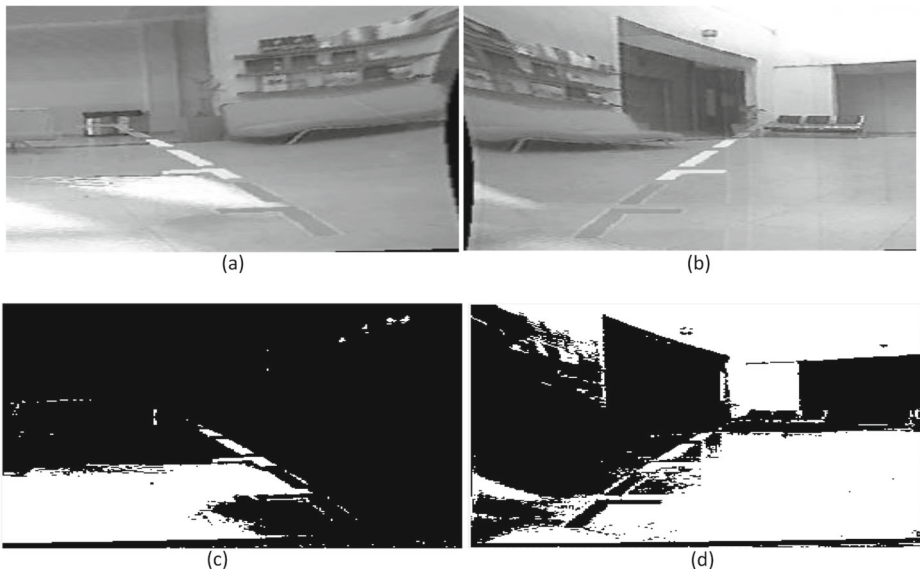


Fig. 12 The corrected images before and after applying thresholding technique **a** Half left corrected image **b** Half right corrected image **c** Half left corrected image after applying threshold and **d** Half right corrected image after applying thresholding technique

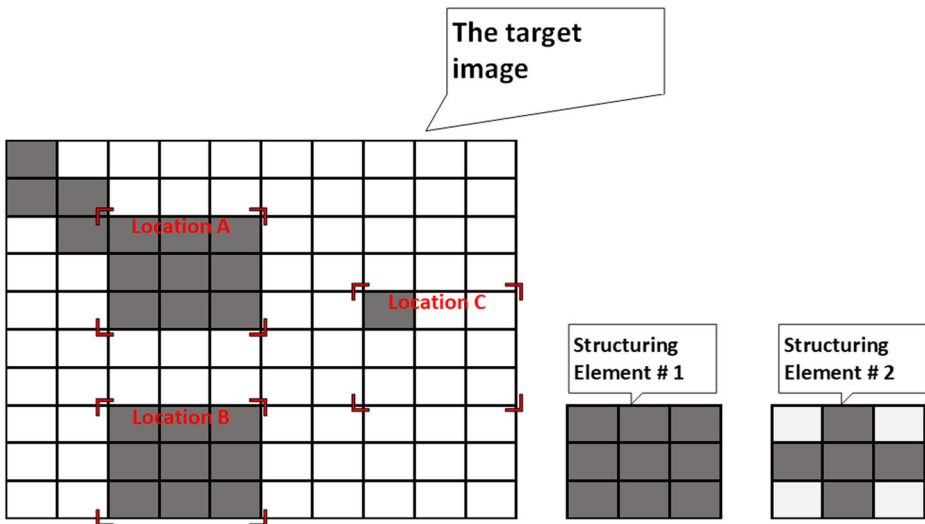


Fig. 13 The operation of the structuring elements on an image

as a result may lead to inaccurate stereo calibration. Therefore, wrong overlapping areas are determined. The function is a filter that removes the large or the small particles in the image. The number of iterations determines the number of erosions to be applied in the image. In the here presented work, the small particles are removed, the number of erosions equal to 9 is experimentally determined. Figure 15 illustrates the corrected overlapping image before and after removing the small objects.

- **Particle analysis:** this is the last step before measuring the correlation between the two images. This step provides around eighty measurements for every particle found, such as the center of mass, the bounding rectangle, the object's area, etc. To measure the correlation between two images the most suitable ten properties are taken, the center of mass pixels (x,y), first pixel (x,y), the bounding rectangle (Top, Left, Right, and Bottom), the area, and the area of the particle over the image area. Figure 16 represents the detected particles of the found objects bounded by rectangles for the left and right corrected overlapping images.



Fig. 14 The images before and after applying the basic morphology on the half overlapping corrected images **a** The half left processed image and **b** The half right processed image

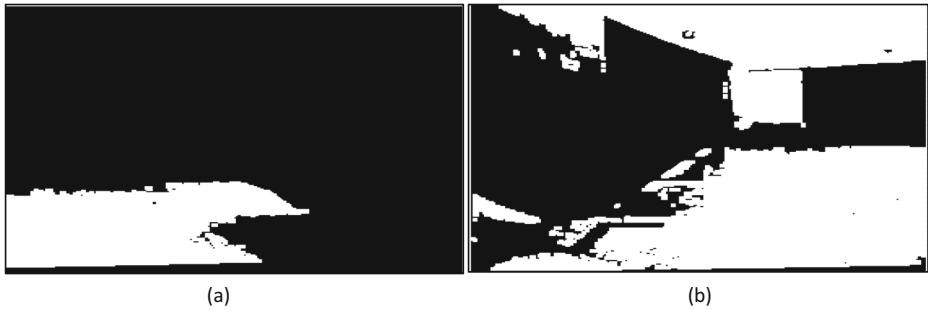


Fig. 15 The corrected overlapping image before and after removing the small objects **a** The half left processed image and **b** The half right processed image

Correlation and offset measurements The cross correlation measurement is a known concept used to find the similarity between two objects. It finds the average difference between two different objects. In this work, the regional correlation is done by taking the measurements found in the last step of (a), namely: particle analysis. The properties are the center of mass (x,y) , first pixel (x,y) , surrounding rectangle, the area of particles, and the area of the particle over the image area. These ten values (A) are taken for each particle found from the two images. The first particle found from the left image is correlated with all particles found from the right image, and the second particle found in the left image is correlated with all particles in the right image and so on. Assuming the number of particles found in the left image are five particles and in the right image are five as well, the correlation loop will repeat twenty five times to correlate each particle in the left image with all particles in the right image. Next, the correlation values finding process between the images is described.

According to the following pseudo code, $MCVn$ is computed.

```

1 set  $n = 0$ 
2 set  $m = 0$ 
3 Compute Array  $X_{diff} = |PnL - PmR|$ 
4 Compute  $X_{sum} = \text{Add All elements of } X_{diff}$ 
5 Compute  $X_{avg} = X_{sum} / A$ 
6 Increment  $m$ 
7 Repeat 3–6 until  $m = \text{max}$ 
8  $MCVn = \min(X_{avg}, n)$ 
9 Increment  $n$ 
10 Repeat 2–8 until  $n = \text{max}$ 

```

Where,

$MCVn$ is the minimum correlation value number n .

n is the number of particles in the left image.

m is the number particles in the right image.

A is the number of used properties.

X_{diff} is a one dimensional array of size (A) containing the resultant values of subtracting two one-dimensional arrays PnL and PmR .

PnL is a one dimensional array of size (A) containing the properties mentioned above for particle number n in the Left image.

PmR is a one dimensional array of size (A) containing the properties mentioned above for particle number m in the Right image.

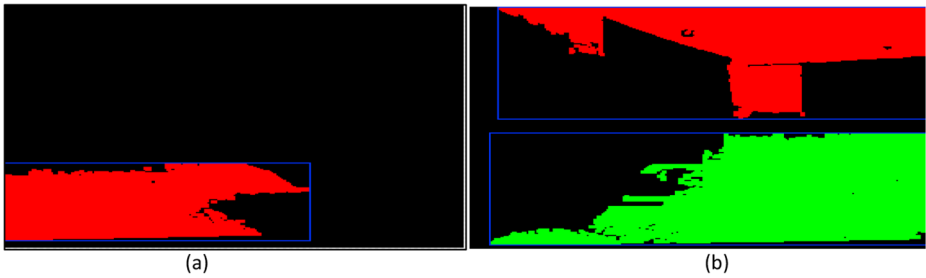


Fig. 16 The detected particles of the found objects bounded by rectangles for the left and right corrected overlapping images **a** The half left image and **b** The half right image

$Xsum$ is the summation of all elements of $Xdiff$.

$Xavg$ is the average of $Xsum$ over A the number of used properties.

The predetermined ten properties (A) of particle 1 in the left image are subtracted from each particle in the right image then the average summation of each one is computed. The properties values of the particles found in the left and right images are illustrated in Fig. 17. These values are used in the previous pseudo code. The correlation values of the particles in the left image with the others in the right image are 736.703 and 359.525 pixels respectively. The minimum correlation value indicates the maximum particles similarity. According to the values that are obtained from the pseudo code, the left particle (Red) with the right particle (Green) have the maximum similarity. The center of mass (x) of these particles are taken to find the average between them $((106.07+ 148.41)/2 = 127.24$ pixels). This average value is used in the next step to measure the image locations.

3.2.4 Panorama image stitching

The corrected overlapped images are cropped to focus on the overlapping area. The size of the images before and after cropping are 526×225 and 263×225 pixels respectively. The (x) average component is found in the system calibration process to be added to the remaining part of the left cropped image. The size of the remaining cropped left image is equal to 263×225 pixels. The left image location and the right image location are computed to be (0, 0) and (390,

	Center (x)	Center (y)	First Pixel (x)	First Pixel (y)	Rect (Left)	Rect (Top)	Rect (Right)	Rect (Bottom)	Particle area	% P. Area / Image area
Left Image Particle (Red):	106.07	170.12	79.00	115.00	0.00	115.00	230.00	217.00	18974.00	32.06

	Center (x)	Center (y)	First Pixel (x)	First Pixel (y)	Rect (Left)	Rect (Top)	Rect (Right)	Rect (Bottom)	Particle area	% P. Area / Image area
Right Image Particle (Red):	160.59	31.95	4.00	0.00	4.00	0.00	263.00	111.00	12259.00	20.72
Left Image Particle (Green):	148.41	172.44	258.00	114.00	16.00	114.00	263.00	221.00	22285.00	37.66

Fig. 17 The resultant properties values for the particles found in left and right image

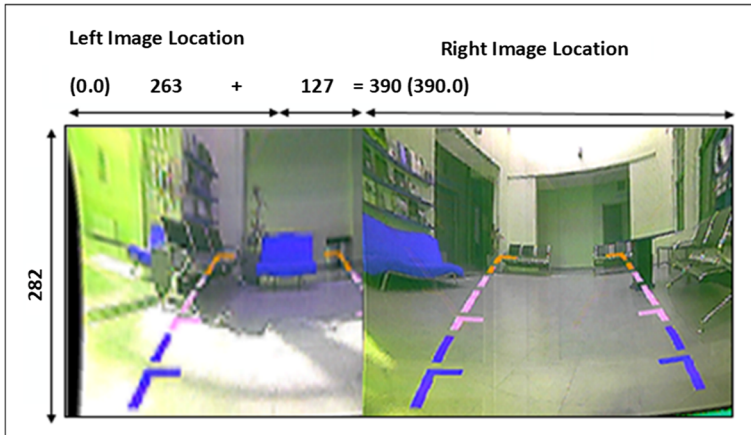


Fig. 18 The resultant panorama image

0) respectively in the destination image. This offset is the right image’s location relative to the left image, or the amount in pixels to shift the right image from the left image as illustrated in Fig. 18. If there is no similarity between the images, then the system will stitch the whole images together without removing any part of them because no overlapping area is found. This correlation is done one time at the beginning of the process, for the first frame from each camera to determine the overlapping area between the cameras.

4 Experimental setup and results

In this work, National Instruments (NI) LabVIEW [18] software is used to implement the proposed algorithm. The proposed system and currently available algorithms in the literature are tested using videos captured by wide-angle fisheye cameras.

The system is implemented as shown in the Fig. 1. Each block in the system is designed and tested separately to make sure that each sub-system works properly and gives the expected results. Afterwards, the overall system is integrated and tested over different parameters and setups.

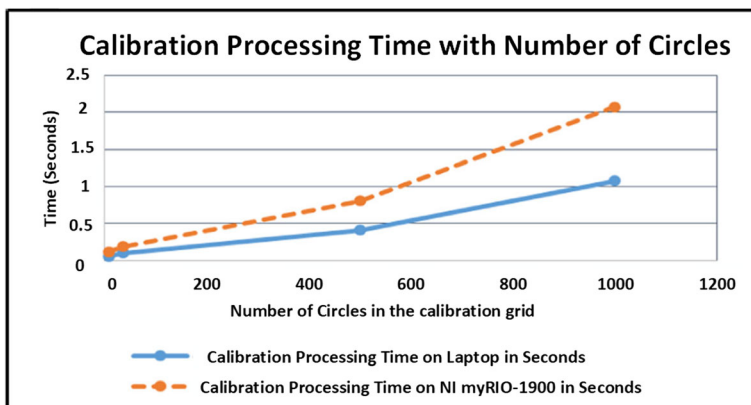


Fig. 19 The calibration processing time for the different grids using the laptop and the NI myRio-1900

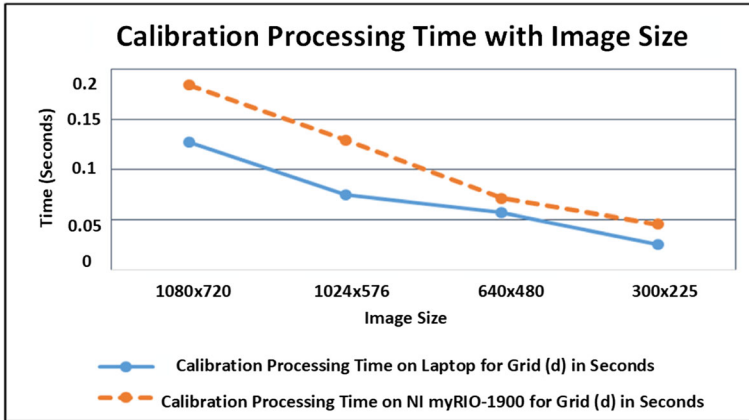


Fig. 20 The calibration processing time using different image sizes on the laptop and the NI myRio-1900

Different templates of grids are used to calibrate the cameras. The circles' coordinates in the sheet are very important. Hence, the overall points should cover the whole sheet in different positions to make a proper calibration. Increasing the number of circles in the grid increases the accuracy of the calibration, because the calibration process detects these circles and determines the center of mass of each one. Increasing the number of circles, increases the detected coordinates in the grid. As a result, it increases the accuracy of the calibration but at the expense of the processing time, because detecting and calibrating each point in the grid takes a considerable amount of time. Therefore, reducing the processing speed of the overall system. The accuracy of calibration appears if the system detects the dots properly and determines the center of mass (x,y) of each one (pixels and their real-world coordinates). In this work, different image sizes are calibrated and the time of each one is computed. The most suitable calibration grid is taken to test the images. Figure 19 shows the calibration processing time for the different grids using the laptop and the NI myRio-1900. Figure 20 illustrates the calibration processing time using different image sizes on the laptop and on the NI myRio-1900.

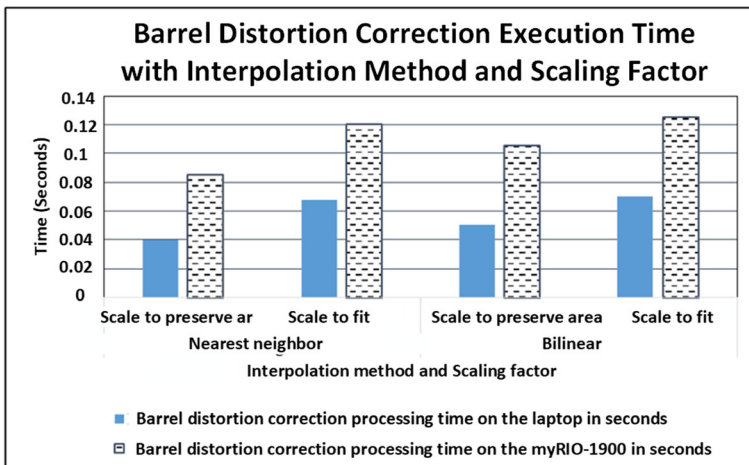


Fig. 21 The barrel distortion correction time with the interpolation method and the scaling factor

Table 1 The processed frames per seconds for the proposed work using the nearest neighbor interpolation method and scale to preserve area factor

Number of cameras	Processed fps on the laptop	Processed fps on the NI myRio-1900
Single Camera	15 fps	9 fps
Two Cameras	11 fps	6 fps

In this work it is required to generate a natural looking image with the shortest possible processing time. As mentioned earlier, the fastest interpolation method has been chosen (nearest interpolation). Another evaluation parameter that must be considered is the scaling factor of the corrected image. This evaluation parameter has two factors, the first one is “scale to preserve the image area” and the second one is “scale to fit”. Scale to preserve area displays the image such that the features of the input image have the same size of the corrected image, but the image size is smaller than the input image. Scale to fit displays the image such that the corrected image size has the same input image size. The processing speed of the scale to preserve image area factor is faster than scale to fit factor. Figure 21 illustrates the barrel distortion correction processing time for both scaling factors applied on the chosen calibration grid on the laptop and on the NI myRio-1900. Table 1 illustrates the processed frames per second using one and two cameras. The calibration grid with a 640×480 image size, scale to preserve image area factor, and using nearest interpolation method type is used.

Figure 22 shows a sequence of screenshots taken from a live video of the generated panorama image using the proposed system. The sequence is from left to right, up to down. A red rectangle is drawn around an object, an apple iPhone in this case, moving back and forth between the images of the two fisheye cameras in the constructed panorama image after going through all the steps described above. The rectangle is drawn in the first row only to enable the reader to better see the object in the other four rows. Tables 2 and 3 illustrate a comparison of the proposed work with other research.



Fig. 22 A sequence of screenshots from a live video with an object moving back and forth between the right and left fisheye images after removing the barrel distortion and generating the panorama image

Table 2 Comparison table between the proposed work and other works – Part I

	Proposed Work implemented on a laptop (calibrates, corrects and stitches two images)	Proposed Work implemented on an embedded device (calibrates, corrects and stitches two images)	[40] (Calibrates and corrects one image)	[4] implemented on a laptop (Calibrates and corrects one image)	[4] implemented on an embedded device (Calibrates and corrects one image)	[7] (Corrects an annular image)	[36] (Stitches twelve undistorted overlapping images)	[38] (Stitches five or nine undistorted overlapping images)
Processing device	2.2 GHz Core i7 and 8 GB	Dual core 667 MHz 512 MB	2.5 GHz Core 2 2 GB	1 GHz, Core2 2 GB	FPGA 668.8 MHz	800 MHz, Core i7 1 GB	_____	_____
Processed fps	11 fps	6 fps	2.3	5.26 fps	22 fps	_____	_____	_____
Field of view (degrees)	310	310	160	_____	_____	360	_____	360
Instantaneous (No overlapping time between the adjacent images)	Yes	Yes	Yes	_____	_____	Yes	_____	Yes
Ability to apply the method on time varying applications	Yes	Yes	A Portion of the resultant image can be applied	Yes	Yes	A Portion of the resultant image can be applied	No	_____
Barel Distortion	Nearest neighbor Interpolation	Nearest neighbor Interpolation	Linear Interpolation	Bi-cubic Interpolation	Bi-cubic Interpolation	A conversion program converts the annular image into a cylindrical	No distortions in the images	Bi-cubic Interpolation
Correction method	_____	_____	_____	_____	_____	_____	_____	_____
Number of cameras	2	2	1	1	1	1	1	5 or 9
Software	LabVIEW	LabVIEW	_____	C code	C code	Apple Quick Time VR	_____	PanoFactory 2.1 and Matlab

Table 3 Comparison table between the proposed work and other works – Part II

	Proposed Work implemented on a laptop (calibrates, corrects and stitches two images)	Proposed Work implemented on an embedded device (calibrates, corrects and stitches two images)	[15] (Calibrates and unwarp two fisheye images)	[13] (calibrates and unwarp two fisheye images)	[12] (calibrates and corrects two fisheye images)
Processing device	2.2 GHz Core i7	Dual core 667 MHz	2.5 GHz Octa-core CPU, and power VR GPU.	—	—
CPU power	8 GB	512 MB	1 GB DDR3	—	—
Memory	11 fps	6 fps	15	—	—
Processed fps	310	310	90	360×180	360×65
Field of view (degrees)	Yes	Yes	Yes	Yes	Yes
Instantaneous (No overlapping time between the adjacent images)	Yes	Yes	Yes	Yes	Yes
Ability to apply the method on time varying applications	Yes	Yes	Yes	Yes	Yes
Barrel Distortion Correction method	Nearest neighbor Interpolation	Nearest neighbor Interpolation	No rectification, just unwarping the fisheye images.	No rectification, just unwarping the fisheye images.	Not mentioned
Number of cameras	2	2	2	2	2
Software	LabVIEW	LabVIEW	—	C++ and Matlab software, and the Rigid MLS has accelerated by GPU	—

5 Conclusion and future work

In this work, a wide-angle stereo vision system based on two fisheye cameras is designed and implemented. The two barrel distorted images are calibrated and corrected instantaneously. The resultant images are correlated to measure the similarity between them. Finally, the system stitches the images into a composite one to generate a panoramic view. The proposed system processes the input fisheye images instantaneously and produces the resultant panoramic images as a live video that suites many applications with strict time constraints. The number of panoramic frames produced by the proposed system using the laptop was 11 fps. The frames per second processing speed of the stereo system using the NI myRio-1900 embedded computing system was 6 fps. The system is able to generate panorama images that cover 310 degrees field of view horizontally. The proposed system achieves a frame rate that fits many real-time applications.

For future work, the proposed system could be enhanced by implementing it using Field Programmable Gate Arrays (FPGA). Moreover, using the stereoscopic system gives the ability to produce 3D views of the scenes providing depth perception. Furthermore, extra work could be done on the quality of the produced panorama image and use of evaluation measures to help improve the output stitched image [9, 10]. Another future research direction is to increase the number of cameras to produce a 360 degree field of view in all directions and in real-time.

References

1. Abughalieh KM, Sababha BH, Rawashdeh NA (2018) A video-based object detection and tracking system for weight sensitive UAVs. *Multimed Tools Appl*:1–19. <https://doi.org/10.1007/s11042-018-6508-1>
2. Al-Harasis R, Al-Zmaily E, Al-Bishawi H, Abu Shash J, Shreim M, Sababha BH (2015) Design and Implementation of an Autonomous UGV for the Twenty Second Intelligent Ground Vehicle Competition. *The International Conference on Software Engineering, Mobile Computing and Media Informatics (SEMCM I 2015)*, Kuala Lumpur
3. Bechar A, Vigneault C (2017) Agricultural robots for field operations. Part 2: Operations and systems. *Biosyst Eng* 153:110–128
4. Bellas N, Chai SM, Dwyer M, Linzmeier D (2009) Real-time fisheye lens distortion correction using automatically generated streaming accelerators. In: *Field Programmable Custom Computing Machines, 2009. FCCM'09. 17th IEEE Symposium on, Napa*
5. Ben-Tzvi P, Xu X (2010) An embedded feature-based stereo vision system for autonomous mobile robots. In: *Robotic and Sensors Environments (ROSE), 2010 IEEE International Workshop on, Phoenix*
6. Chira IM, Chibulcutean A, Danescu RG (2010) Real-time detection of road markings for driving assistance applications. In: *Computer Engineering and Systems (ICCES), 2010 International Conference on, Cairo*
7. Driscoll Jr E, Morrow H, Steinhauer AJ, Lomax WC (2002) Method and apparatus for a panoramic camera to capture a 360 degree image. U.S. Patent Patent 6,459,451
8. Drulea M, Szakats I, Vatavu A, Nedevschi S (2014) Omnidirectional stereo vision using fisheye lenses. In: *Intelligent Computer Communication and Processing (ICCP), 2014 IEEE International Conference on, Cluj Napoca, Romania*
9. Fan D-P, Cheng M-M, Liu Y, Li T, Borji A (2017) Structure-measure: A new way to evaluate foreground maps. In: *Proceedings of the IEEE International Conference on Computer Vision, Venice*
10. Fan D-P, Gong C, Cao Y, Ren B, Cheng M-M, Borji A (2018) Enhanced-alignment Measure for Binary Foreground Map Evaluation. In: *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18), Stockholm*
11. Gao L-F, Gai Y-X, Fu S (2007) Simultaneous localization and mapping for autonomous mobile robots using binocular stereo vision system. In: *Mechatronics and Automation, 2007. ICMA 2007. International Conference on*
12. Gao W, Shen S (2017) Dual-Fisheye Omnidirectional Stereo. *IEEE/ RSJ International Conference on Intelligent Robots and system (IROS)*, Vancouver
13. Ho T, Schizas ID, Rao KR, Budagavi M (2017) 360-Degree Video Stitching for Dual-Fisheye Lens Cameras Based on Rigid Moving Least Squares. [arXiv:1708.05922v1](https://arxiv.org/abs/1708.05922v1)

14. Kim H, Chae E, Jo G, Paik J (2015) Fisheye lens-based surveillance camera for wide field-of-view monitoring. In: Consumer Electronics (ICCE), 2015 IEEE International Conference on, Las Vegas
15. Liu T-M, Ju C-C, Huang Y-H, Chang T-S, Yang K-M, Lin Y-T (2017) A 360-degree 4Kx2K Panorama Video Processing Over Smart-phones. IEEE International Conference on Consumer Electronics (ICCE)
16. Mathworks (2009) Interpolation-methods. Available: <https://www.mathworks.com/help/vision/ug/interpolation-methods.html>. [Accessed 6 July 2017]
17. National Instruments (2015) myRIO Student Embedded Device. Available: <http://www.ni.com/en-lb/shop/select/myrio-student-embedded-device>
18. National Instruments. LabVIEW. Available: <http://www.ni.com/en-lb/shop/labview.html>
19. T. Nishimoto and J. Yamaguchi (2007) Three dimensional measurement using fisheye stereo vision. In: SICE Annual Conference 2007, Takamatsu
20. Ohashi A, Tanaka Y, Masuyama G, Umeda K, Fukuda D, Ogata T, Narita T, Kaneko S, Uchida Y, Irie K (2016) Fisheye stereo camera using equirectangular images. In: Mechatronics (MECATRONICS)/17th International Conference on Research and Education in Mechatronics (REM), 2016 11th France-Japan & 9th Europe-Asia Congress on, Compiègne
21. Polydoros AS, Nalpantidis L (2017) Survey of model-based reinforcement learning: Applications on robotics. *J Intell Robot Syst* 86(2):153–173
22. Rawashdeh NA, Rawashdeh OA, Sababha BH (2017) Vision-based sensing of UAV attitude and altitude from downward in-flight images. *J Vib Control* 23(5):827–841
23. Rawashdeh O, Sababha B (2011) An Image-Processing-Based Gimbal System Using Fisheye Video. *Computer Technology and Application Journal* 2(2):85–93
24. Rawashdeh OA, Yang HC, AbouSleiman RD, Sababha BH (2009) Microraptor: A low-cost autonomous quadrotor system. In: ASME 2009 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference
25. Ray LA, Schaufelle CN, VanSprewenburg B (2000) Method and apparatus for capturing panoramic images with range data. U.S. Patent Patent 6,023,588
26. Sababha BH, Al Zu'bi HM, Rawashdeh OA (2015) A rotor-tilt-free tricopter UAV: design, modelling, and stability control. *International Journal of Mechatronics and Automation* 5(2–3):107–113
27. Sawalmeh HK, Bjanthala HE, Al-Lahham MM, Sababha BH (2015) A Surveillance 3D Hand-Tracking-Based Tele-Operated UGV. In: The 6th International Conference on Information and Communication Systems (ICICS2015), Amman
28. Semwal VB, Bhushan A, Nandi GC (2013) Study of humanoid Push recovery based on experiments. In: Control, Automation, Robotics and Embedded Systems (CARE), 2013 International Conference on
29. Semwal VB, Chakraborty P, Nandi GC (2015) Less computationally intensive fuzzy logic (type-1)-based controller for humanoid push recovery. *Robot Auton Syst* 63:122–135, 2015
30. Semwal VB, Gaud N, Nandi G (2017) Human Gait State Prediction Using Cellular Automata and Classification Using ELM. In: MISIP-2017
31. Semwal VB, Katiyar SA, Chakraborty R, Nandi GC (2015) Biologically-inspired push recovery capable bipedal locomotion modeling through hybrid automata. *Robot Auton Syst* 70:181–190, 2015
32. Semwal VB, Mondal K, Nandi GC (2017) Robust and accurate feature selection for humanoid push recovery and classification: deep learning approach. *Neural Comput & Applic* 28(3):565–574
33. Semwal VB, Nandi GC (2016) Generation of joint trajectories using hybrid automate-based model: a rocking block-based approach. *IEEE Sensors J* 16(12):5805–5816
34. Semwal VB, Raj M, Nandi GC (2015) Biometric gait identification based on a multilayer perceptron. *Robot Auton Syst* 65:65–75, 2016
35. Silva LC, Petraglia MR, Petraglia A (2004) A robust method for camera calibration and 3-D reconstruction for stereo vision systems. In: Proceedings of the 12th European Signal Processing Conference, Vienna
36. Sinclair MJ, Simmons LE (2003) Panoramic Digital Camera System and Method. US Patent Patent 6,545, 701 B2
37. Takano T, Ono S, Matsushita Y, Kawasaki H, Ikeuchi K (2015) Super resolution of fisheye images captured by on-vehicle camera for visibility support. In: Vehicular Electronics and Safety (ICVES), 2015 IEEE International Conference on, Yokohama
38. Wachtel RA, Keable J, Paulson R (2007) Digital imaging system using overlapping images to formulate a seamless composite image and implemented using either a digital imaging sensor array. U.S. Patent Patent 7,215,364
39. Wang W, Yan J, Xu N, Wang Y, Hsu F-H (2015) Real-time high-quality stereo vision system in FPGA. *IEEE Transactions on Circuits and Systems for Video Technology* 25(10):1696–1708
40. Wei J, Li C-F, Hu S-M, Martin RR, Tai C-L (2012) Fisheye video correction. *IEEE Trans Vis Comput Graph* 18(10):1771–1783

41. Zhang B, Liu N, Jiao Y, Li Y, Zhu J (2015) Research on image matching technology for the spherical stereo vision. In: Mechatronics and Automation (ICMA), 2015 IEEE International Conference on, Beijing
42. Zhao S, Zhang B, Li L, Cao Z (2010) Fisheye lens camera system calibration and localization error analysis. In: Computer Design and Applications (ICCD), 2010 International Conference on, Qinhuangdao

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Raghad O. Al-Harasis, received her Bachelor and Master degrees in Telecommunications and Electrical Engineering from Princess Sumaya University for Technology (PSUT), Jordan in 2015 and 2017, respectively. She worked as teaching and research assistant at PSUT. Her research concentration areas are in UGV development and machine Vision.



Belal H. Sababha, Ph.D. is an Associate Professor of Electrical and Computer Engineering at the King Abdullah II School of Engineering, Princess Sumaya University for Technology, Amman, Jordan. Currently spending a leave at HCT, UAE. While in Academia, Dr. Sababha has published more than 35 peer reviewed scientific publications. He is a US patent holder and IEEE Conference Chair. He is a member of various accreditation advisory boards, international conference scientific committees, technical program committees and organization committees. He is a reviewer for several high impact journals. Prior moving to academia, Dr. Sababha has worked in the Automotive Industry. He worked as a Senior Controls Engineer in the Powertrain Controls

department at Chrysler Group LLC, Michigan USA. Belal received his PhD degree in Electrical and Computer Engineering – Embedded Systems from Oakland University, MI, USA in 2011. He has taught electrical and computer engineering undergraduate and graduate courses at various universities in the US and Jordan. He has many years of experience in curricula design, pedagogy in higher education and lead several teams at various universities in implementing curricular continuous improvement and attain national and international accreditation. Dr. Sababha has extensive experience in embedded systems design, control algorithm design and software development with applications related to Gasoline Engine Controls and Unmanned Aerial Vehicles (UAVs). He is a consultant in the fields of Embedded Systems and UAV design and control for various governmental and commercial firms. His research concentration areas are UAV development and control, Biomedical instrumentation, embedded sensors, embedded RTOS and CAN networks, distributed embedded systems, graceful degradation in embedded systems, rapid prototyping, and machine vision. Dr. Sababha has served in several senior leadership positions as a Dean, Associate Executive Dean, Director and Chairman. He is a Senior Member of IEEE and a member of several national and international professional organizations.