Copyright © by TUAN P. M. HO
All Rights Reserved
To my wife Nhung Nguyen.
ABSTRACT

DUAL-FISHEYE LENS STITCHING FOR 360-DEGREE IMAGING

TUAN P. M. HO, Ph.D.

The University of Texas at Arlington,

Supervising Professor: K. R. Rao

Dual-fisheye lens cameras have been increasingly used for 360-degree immersive imaging. However, the limited overlapping field of views and misalignment between the two lenses give rise to visible discontinuities in the stitching boundaries. This proposal introduces a novel method for dual-fisheye camera stitching that adaptively minimizes the discontinuities in the overlapping regions to generate full spherical 360-degree images. Results show that this approach can produce good quality stitched images for Samsung Gear 360, a dual-fisheye camera, even with hard-to-stitch objects in the stitching borders.
# TABLE OF CONTENTS

**ABSTRACT** ................................................................. iv

**LIST OF ILLUSTRATIONS** ............................................. vii

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Background</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Previous Works</td>
<td>2</td>
</tr>
<tr>
<td>2. Fisheye-lens Stitching Algorithm</td>
<td>5</td>
</tr>
<tr>
<td>2.1 Fisheye Lens Intensity Compensation</td>
<td>5</td>
</tr>
<tr>
<td>2.2 Fisheye Unwarping</td>
<td>6</td>
</tr>
<tr>
<td>2.3 Two-step Image Alignment</td>
<td>8</td>
</tr>
<tr>
<td>2.3.1 Control Point-based Alignment</td>
<td>10</td>
</tr>
<tr>
<td>2.3.2 Refined Alignment</td>
<td>12</td>
</tr>
<tr>
<td>2.4 Implementation and Results</td>
<td>13</td>
</tr>
<tr>
<td>2.5 Conclusions</td>
<td>16</td>
</tr>
<tr>
<td>2.6 Future Work</td>
<td>16</td>
</tr>
<tr>
<td>2.6.1 Improving Stitching Quality</td>
<td>16</td>
</tr>
<tr>
<td>2.6.2 360-degree Video Stitching</td>
<td>17</td>
</tr>
<tr>
<td>2.6.3 360-degree Video Compression</td>
<td>17</td>
</tr>
</tbody>
</table>

Appendix

A. MPEG Proposal: Lens Shading Parameters Metadata for Omnidirectional Video ......................................................... 18

**REFERENCES** .............................................................. 21
## LIST OF ILLUSTRATIONS

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>(a) Samsung Gear 360 Camera (left). (b) Gear 360 dual-fisheye output image (7776 x 3888 pixels) (right). Left half: image taken by the front lens. Right half: image taken by the rear lens.</td>
<td>2</td>
</tr>
<tr>
<td>1.2</td>
<td>Image stitching illustration. Left column: (a) Regular pictures with good overlaps. (b)(c) Features Matching using SIFT and outlier removal using RANSAC. (d) Image warping and panorama creation. Right column: (e) Fisheye images taken by Samsung Gear 360. (f)(g) Features Matching (using SIFT) and outlier removal (using RANSAC). Courtesy: VLFeat [1] toolbox.</td>
<td>3</td>
</tr>
<tr>
<td>2.1</td>
<td>Block Diagram of Our Fisheye Images Stitching [12].</td>
<td>5</td>
</tr>
<tr>
<td>2.2</td>
<td>(a) Fisheye profiling experiment (left). (b) Intensity fall-off curve (right). Coordinate in x-direction is in pixel unit.</td>
<td>6</td>
</tr>
<tr>
<td>2.3</td>
<td>Compensation result. Without intensity compensation (left); with intensity compensation (right).</td>
<td>7</td>
</tr>
<tr>
<td>2.4</td>
<td>Fisheye Unwarping.</td>
<td>8</td>
</tr>
<tr>
<td>2.5</td>
<td>Unwarping results.</td>
<td>9</td>
</tr>
<tr>
<td>2.6</td>
<td>Experimental Setup for Gear 360 Dual-lens Mis-alignment.</td>
<td>10</td>
</tr>
<tr>
<td>2.7</td>
<td>Control point selection on the overlapping area. The fisheye images are unwarped using the method presented in section 2.2.</td>
<td>11</td>
</tr>
<tr>
<td>2.8</td>
<td>Result of the control point-based alignment. (a) Without the first alignment. (b) With the first alignment.</td>
<td>11</td>
</tr>
</tbody>
</table>
2.9  (a) A person close to the camera and between the lens boundaries. (b) The blended overlaps with the proposed refined alignment (discontinuity minimized). (c) The blended overlaps without the proposed refined alignment (very visible discontinuity). The first alignment is already applied for both (b) and (c) to align the images vertically.

2.10 Unwarping results: ......................................................... 14

2.11 Unwarping results: ......................................................... 15

A.1 Screenshot of our proposal, page 1. ................................. 19

A.2 Screenshot of our proposal, page 2. ................................. 20
CHAPTER 1
INTRODUCTION

1.1 Background

360-degree videos and images have become very popular with the advent of easy-to-use 360-degree viewers such as Cardboard [2] and GearVR [3]. This has led to renewed interest in convenient cameras for 360-degree capturing. A 360-degree image captures all the viewing directions simultaneously and gives users the sense of immersion when viewed. Early 360-degree imaging systems used a catadioptric camera [4], which combines lens (dioptic) and mirror (catoptric), to record 360-degree contents. Although the lens plus mirror geometry is sophisticated and usually requires proper calibration, such as one in [5] [6], to generate good visual results, a catadioptric system can produce panoramas without seams. However, due to the inherent lens+mirror arrangement, the captured field of view is typically limited to less than 360x180 degrees, and some of the catadioptric systems are not compact.

An alternate method for 360-degree recording is using a polydioptric system which incorporates multiple wide-angle cameras with overlapping field of views. The images from the multiple cameras are stitched together to generate 360-degree pictures. However, due to camera parallax, stitching artifacts are typically observed at the stitching boundaries. Example 360-degree polydioptric cameras include Ozo [7], Odyssey [8], and Surround360 [9] by some of the major companies. The number of cameras used in these systems ranges from 8 to 17. These cameras typically deliver professional quality, high-resolution 360-degree videos.
On the downside, these high-end 360-degree cameras are bulky and extremely expensive, even with the decreasing cost of image sensors, and are out of reach for most of the regular users. To bring the immersive photography experience to the masses, Samsung has presented Gear 360 camera, shown in Figure 1.1(a). To make the camera compact, Gear 360 uses only two fisheye lenses whose field of view is close to 195 degrees each. The images generated by the two fisheye lenses (Figure 1.1(b)) have very limited overlapping field of views but can, however, be stitched together to produce a full spherical 360x180 panorama.

1.2 Previous Works

For stitching of images from the multiple cameras, a feature-based stitching algorithm [10][11] is typically used to extract the features of the images being stitched. These features are then matched together. An iterative method is carried out to eliminate the incorrect matches (outliers). The reliability of this process not only depends on the iterative method being used but also on the size of the overlapping regions. With sufficient overlap, more reliable matches (inliers) are retained while
Figure 1.2: Image stitching illustration. Left column: (a) Regular pictures with good overlaps. (b)(c) Features Matching using SIFT and outlier removal using RANSAC. (d) Image warping and panorama creation. Right column: (e) Fisheye images taken by Samsung Gear 360. (f)(g) Features Matching (using SIFT) and outlier removal (using RANSAC). Courtesy: VLFeat [1] toolbox.

outliers get removed. Using these inliers, a homography matrix is computed to warp and register the pictures together (assuming the camera matrix is already estimated) before stitching them.

However, this conventional stitching method does not work well on Gear 360-produced pictures since there is very limited overlap between the two fisheye images. Figure 1.2 shows the stitching processes for the photos taken by the regular rectilinear lens and the ones taken by Samsung Gear 360. The pictures on the left column, from
[1], in Figure 1.2 have a good overlap and can be aligned and stitched well. In contrast, Gear 360 has limited overlap leading to a small number of inlier matches only on the outer ring of the fisheye images. This results in a homography matrix that is invalid for the interior of the fisheye images. Hence, a conventional stitching process cannot be directly used for stitching fisheye images from two-lens systems such as Gear 360.
CHAPTER 2

Fisheye-lens Stitching Algorithm

We have introduced a novel stitching method [12] that adaptively minimizes the discontinuities in the overlapping regions of Gear 360 images to align and stitch them together. The proposed algorithm has four steps. The first step describes how to measure and compensate for the intensity fall off of the cameras fisheye lenses. The second phase explains the geometry transformation to unwarp the fisheye images to a spherical 2-Dimensional (equirectangular projection [13]) image. The next stage introduces our proposed two-step alignment to register the fisheye unwarped images. Finally, the aligned images are blended to create a full spherical 360x180-degree panorama. Figure 2.1 shows the block diagram of our fisheye stitching framework.

2.1 Fisheye Lens Intensity Compensation

Before stitching the two fisheye images, one needs to compensate for the light fall-off of the two fisheye images.
Vignetting is an optical phenomenon in which the intensity of the image reduces at the periphery compared to the center. To compensate for this light fall-off, we captured an image of a large white blank paper using Gear 360 and measured the variation of pixel intensity along the radius of the fisheye image toward its periphery in Figure 2.2. The intensity is normalized to one at the center of the picture. We used a polynomial function $p(x)$ to fit the light fall-off data.

$$p(x) = p_1 x^n + p_2 x^{n-1} + \ldots + p_n x + p_{n+1}$$  \hspace{1cm} (2.1)

whereas $x$ is the radius from the center of the image. Figure 2.3 shows the result of this process.

2.2 Fisheye Unwarping

Fisheye lenses can produce ultra-wide field of views by bending the incident lights. As a result, the image looks severely distorted, particularly in the periphery.

Figure 2.2: (a) Fisheye profiling experiment (left). (b) Intensity fall-off curve (right). Coordinate in x-direction is in pixel unit.
Therefore, a fisheye unwarping—a geometric transformation is necessary to generate a natural appearance for the Gear 360 fisheye-produced picture. Instead of rectifying the fisheye-distorted image, we use a method that unwarps the image and returns a 2-D spherical projected picture for 360-degree purposes.

This method involves two steps, shown in Figure 2.4. First, each point $P'(x', y')$ in the input fisheye image is projected to a 3-D point $P(\cos \theta_s \sin \varphi_s, \cos \theta_s, \sin \theta_s)$ in the unit sphere. $\varphi_s$ and $\theta_s$ can be derived by considering the coordinates of the fisheye image directly as pitch and yaw. Therefore, $\theta_s = (f \ast x)/W - 0.5$, and $\varphi_s = (f \ast y)/H - 0.5$, where $f$ is the lens field of view (in degree), $W$ and $H$ are image width and height respectively. The second step derives the distance between the projected center and the 3-D point $P(x, y, z)$: $\rho = H/f \ast \tan^{-1}(\sqrt{x^2 + z^2})/y$.
whereas \( x = \cos \varphi s \sin \theta s \), \( y = \cos \varphi s \cos \theta s \), \( z = \sin \varphi s \). Then the 2-D spherical (equirectangular) projected point \( P''(x'', y'') \) is constructed as \( x'' = 0.5W + \rho \cos \theta \), \( y'' = 0.5H + \rho \sin \theta \), and \( \theta = \tan^{-1} \frac{z}{x} \). In this equirectangular projection, \( x'' \) and \( y'' \) are pitch and yaw respectively. The unwarped image can be viewed on a 360-degree player. Figure 2.5 shows the result of the unwarping process. Figure 2.5 (a) illustrates the natural look of the unwarped image when viewed on a 360-degree viewer compared to the distorted appearance of the original fisheye image.

Unwarping is a necessary process in fisheye image stitching. However, result in Figure 2.5 (b) shows that the unwarped images, when put together in a 360x180 plane, are not aligned with each other. Therefore, further steps have to be taken to register the pictures together.

2.3 Two-step Image Alignment

After unwarping, the two images are not aligned with each other. The between-lenses misalignment patterns are similar for different Gear 360 cameras of the same model. To minimize this misalignment we have used a control-point-based approach followed by a refined alignment as follows.
Figure 2.5: Unwarping results: (a) Fisheye-unwarped image whose view port displayed on a 360-degree viewer; (b) Two unwarped images arranged in a 360x180 image.
2.3.1 Control Point-based Alignment

In the setup in Figure 2.6, we position the Gear 360 so that both the lenses see the checkerboards on their sides. Therefore, they have the same view of the overlapping regions. Also, the distance between the camera and the checkerboards is around 2m, which is about the maximum reach that the checkerboard corners are still clearly visible for control point selection. The images taken by the Gear 360 left and right lenses are unwarped using the method introduced in section 2.2, and arranged in 360x180-degree planes shown in Figure 2.7. About 200 pairs of control points are then manually selected from the overlapping regions between the unwarped pictures, and are used to estimate a 6-parameter affine matrix $A$, which warps a point $B(x_2, y_2)$ to $T(x_1, y_1)$ as follows:

$$
[x_1, y_1, 1] = [x_2, y_2, 1] A, \quad \text{whereas } A = \begin{bmatrix}
a & b & 0 \\
c & d & 0 \\
t_x t_y & 1
\end{bmatrix}
$$ (2.2)
Figure 2.7: Control point selection on the overlapping area. The fisheye images are unwarped using the method presented in section 2.2.

Figure 2.8: Result of the control point-based alignment. (a) Without the first alignment. (b) With the first alignment.
2.3.2 Refined Alignment

The first registration helps align the images, as shown in Figure 2.8. However, when the objects in the boundaries move closer or further away from the camera, the horizontal discontinuities become visible as shown in Figure 2.9(c).

To minimize the discontinuity in the overlapping regions, we choose to maximize the similarity in these areas. To this end, a novel adaptive alignment that involves a fast template matching for objects in the overlapping region and utilizes the matching displacement to derive a refined affine matrix to align the images further is proposed.

The matching is a normalized cross-correlation operation. The cross-correlation of two signals maximizes at a point when the two signals match each other. In addition, since there are always some level of exposure differences in the overlapping regions, the template and reference images to be matched should be normalized. This proposal employs a fast normalized cross-correlation algorithm [14]:

$$\gamma(u, v) = \frac{\sum_{x,y}[f(x,y) - \bar{f}(u,v)][t(x-u, y-v) - \bar{t}]}{[\sum_{x,y}[f(x,y) - \bar{f}(u,v)]^2 \sum_{x,y}[t(x-u, y-v) - \bar{t}]^2]^{0.5}}$$

(2.3)

where $\gamma$ is the normalized cross-correlation, $f$ is the reference image, $\bar{t}$ is mean of the template image, $\bar{f}(u, v)$ is the mean of $f(x, y)$ in the region under the template. The template and reference are taken from the top and bottom unwarped images respectively, as shown in Figure 2.9(a).

The maximum value of the normalized cross-correlation returns the displacement of where the best match occurs. This shift indicates how much the template – a rectangular window has to move to match the reference. The proposed method then estimates an affine matrix from vertices of the matching windows (four in each overlapping region) and warp the bottom image to align it further with the top one.
Figure 2.9: (a) A person close to the camera and between the lens boundaries. (b) The blended overlaps with the proposed refined alignment (discontinuity minimized). (c) The blended overlaps without the proposed refined alignment (very visible discontinuity). The first alignment is already applied for both (b) and (c) to align the images vertically.

Figure 2.9 shows that the refined alignment helps align the images by maximizing the similarity in the overlapping region. The person, close and in the lens boundary, appears as a complete one (i.e. no visible duplicate or missing any body parts) in the stitched 360x180-degree picture.

2.4 Implementation and Results

We have implemented the proposed approach in C++ with OpenCV library and Matlab. The affine matrix in the first alignment is pre-computed off-line and included as part of the fisheye unwarping process. The refined alignment, however, is computed on-line, adaptively to the scene. The polynomial coefficients in section 2.1 are: 
\[ p_1 = -7.5625 \times 10^{-17}, \quad p_2 = 1.9589 \times 10^{-13}, \quad p_3 = -1.8547 \times 10^{-10}, \quad p_4 = 6.1997 \times 10^{-8}, \quad p_5 = -6.9432 \times 10^{-5}, \quad p_6 = 0.9976. \] 
We found that the field of view of 193 degrees, which is very close to the documented 195-degree, gives the best results as shown in Figures 2.10 and 2.11. Our approach can also accurately stitch images taken by different Gear 360 cameras of same model thanks to the proposed refined
alignment that operates adaptively and can compensate for the geometric mismatch between Gear 360 lenses.

Figure 2.10: Samsung Gear360 360x180-degree panoramas stitched by the proposed method: (a) A person not close to the lenses overlapping boundaries; (b) A person close to the lenses overlapping boundaries.
Figure 2.11: Samsung Gear360 360x180-degree panoramas stitched by the proposed method: (a) Garage; (b) Building with patterned ground.
2.5 Conclusions

This proposal introduces a new stitching method for 360-degree cameras with dual-fisheye lens. It uses a novel alignment algorithm that adaptively maximizes the similarities in the boundary regions of the images from the two fisheye lenses for accurate registration and stitching. In summary, the proposed approach compensates for fisheye lens intensity fall-off, unwarps the fisheye images, then registers them together using the proposed adaptive alignment, and applies blending on the registered images to create a 360x180-degree panorama that is viewable on 360-degree players. Results show that not only this method can stitch Gear 360 images that have limited overlap, but it can also produce well-stitched pictures even if there are objects that are at an arbitrary distance to the camera and stand in the lenses boundaries.

2.6 Future Work

Although this work has provided a complete framework for stitching the fisheye-lens images, there are still many rooms for improvement.

2.6.1 Improving Stitching Quality

As shown in Figure 2.11(b), there are visible discontinuities in the stitching boundary of pictures with special pattern on the background. This stems from the fact that our current method solves an over-determined system for a warping matrix (subsection 2.3.1). This results in a least-squares approximated solution which could not transform all the source control points to the desired target positions precisely.

Therefore, instead of estimating a warping matrix in a least-squares sense, we generate interpolation grids to deform the image by a weighted least-squares approach. In particular, we assign higher weight to the grid points inside the boundary and let the weight gradually decline when the points’ position moving away from the stitching boundary.
borders. In other words, we estimate the grids by solving the moving least-squares system. This method can potentially out-perform the current one in term of stitching quality.

2.6.2 360-degree Video Stitching

We also would like to do more experiments with 360-degree videos after finishing the still image stitching. Each frame in the video sequence is comprised of multiple pictures from different lenses (two for dual-fisheye lens camera). Therefore, it is crucial to maintain the same level of image warping among frames in a video sequence. Otherwise, it would introduce visible glitches in the video.

2.6.3 360-degree Video Compression

Another aspect of improvement would be to develop a framework to compress the 360-degree videos which content far more data compared to the non-360-degree ones.
APPENDIX A

MPEG Proposal: Lens Shading Parameters Metadata for Omnidirectional Video
In this appendix, we propose to MPEG our method to compensate for the light fall of when fisheye lenses are used to create 360-degree contents.

INTERNATIONAL ORGANISATION FOR STANDARDISATION
ORGANISATION INTERNATIONALE DE NORMALISATION
ISO/IEC JTC1/SC29/WG11
CODING OF MOVING PICTURES AND AUDIO
ISO/IEC JTC1/SC29/WG11 MPEG2016/ m39469
October 2016, Chengdu, China

<table>
<thead>
<tr>
<th>Source</th>
<th>Samsung Electronics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>Input contribution</td>
</tr>
<tr>
<td>Title</td>
<td>OMAF: Lens Shading Parameters Metadata for Omnidirectional Video</td>
</tr>
<tr>
<td>Author</td>
<td>Tuan Ho, Madhukar Budgavli</td>
</tr>
</tbody>
</table>

Proposal

360-degree videos and images have become very popular with the advent of easy-to-use 360-degree viewers such as Cardboard [1] and GearVR [2]. This has led to renewed interest in convenient cameras for 360-degree image and video capture. Early 360-degree imaging systems used a catadioptric camera [3], which combines lens (dioptic) and mirror (catoptic), to record 360-degree contents. Due to the inherent lens+mirror arrangement, the captured field of view is typically limited to less than 360x180 degrees, and some of the catadioptric systems are not compact. An alternate method for 360-degree recording is using a polydioptic system which incorporates multiple wide-angle cameras with overlapping field of views. The images from the multiple cameras are stitched together to generate 360-degree pictures. The number of cameras used in these systems ranges from 8 to 18. These cameras typically deliver professional quality, high-resolution 360-degree videos but are bulky and expensive to use. To bring the immersive photography experience to the masses, several companies have created compact 360 degrees camera using only two fisheye lenses. Figure 1 shows one such example.

Figure 1: Samsung Gear360. Compact 360 camera with two fisheye lens.

M38938 [4] proposes that the images captured by such dual fisheye lens camera be directly encoded and transmitted. At the receiver side, the videos are then directly rendered according to user’s viewpoint. The advantage of this method is that it reduces end-to-end delay due to the absence of an intermediate stitching/projection step. Since an intermediate stitching/projection step is not used there is also one less resampling when compared to traditional 360 degrees systems which typically use an intermediate stitching/projection step to equirectangular or cubemap format. This reduction in the number of resamplings helps improve video quality.

Figure A.1: Screenshot of our proposal, page 1.
M38938 proposes metadata for several of the processing steps involved on the receiver side. Lens shading compensation is one such step that needs to be carried out on the receiver side. This is because of the vignetting optical phenomenon in which the amount of light reaching the focal plane/image sensor falls off as we move away from the center of the lens. Figure 2 below shows the lens shading curve—the blue curve is the measured lens shading curve obtained by capturing an image of a large white blank paper.

![Figure 2: Typical lens shading curve.](image)

M38938 proposes transmission of the lens shading curve samples to help with lens shading compensation at the receiver side. This contribution proposes instead that a polynomial approximation of the lens shading curve be transmitted to reduce the amount of metadata. The red curve in Figure 2 shows the polynomial approximation of the lens shading curve. The polynomial is of the form \( p(x) = p_1 x^0 + p_2 x^1 + \ldots + p_n x^n \) where \( x \) is the radius from the center of the image. A polynomial of order around 4 to 6 is sufficient to approximate the lens shading curve.

### Syntax

```c
unsigned int[16] num_polynomial_coefficients_lsc; for (j=0; j < num_polynomial_coefficients_lsc; j++) {
    unsigned int[32] polynomial_coefficient_K_lsc_R;
    unsigned int[32] polynomial_coefficient_K_lsc_G;
    unsigned int[32] polynomial_coefficient_K_lsc_B;
}
```

### Semantics

Figure A.2: Screenshot of our proposal, page 2.
REFERENCES


<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIFT</td>
<td>Scale-invariant feature transform</td>
</tr>
<tr>
<td>RANSAC</td>
<td>Random sample consensus</td>
</tr>
<tr>
<td>MPEG</td>
<td>Moving Picture Experts Group</td>
</tr>
<tr>
<td>JCT-VC</td>
<td>Joint Collaborative Team on Video Coding - ITU</td>
</tr>
</tbody>
</table>