Dual-fisheye Lens Stitching for 360-degree Imaging & Video

Tuan Ho, PhD. Student
Electrical Engineering Dept.,
UT Arlington

UNIVERSITY OF TEXAS ARLINGTON | COLLEGE OF ENGINEERING

Introduction

- 360-degree imaging: the process of taking multiple photographs and stitching them together to create a 360x180-degree image.
- Imaging systems:
  - Catadioptric: using lens+mirror → no stitching seams but **limited** in field of view (cannot produce 360x180 images).
  - Polydioptric: using multiple wide-view lenses in a same rig.

Columbia U's catadioptric camera

GoPro's Odyssey 360 rig (16 cameras)

Facebook Surround 360 rig (17 cameras)

Motivation

- Polydioptric cameras become more and more popular in 360-degree imaging and video.
- But most professional polydioptric optical systems are
  - Bulky
  - Very expensive (ranging $15,000 ~ $30,000 for a complete system) → not for the masses
- Samsung Gear 360 camera:
  - Very compact
  - Affordable (around $300/camera)
  - Use two fisheye lenses

Gear 360:

Affordable optics and simpler hardware.

Complexity shifted to stitching algorithm.

Challenges

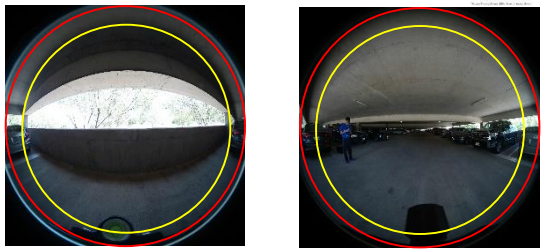
- Little overlap between two fisheye images taken by Gear 360.
- Mis-alignment between two lenses in the camera.
- Fisheye light drop-off: intensity decreases moving away from the center of the image.

Challenges (cont.)

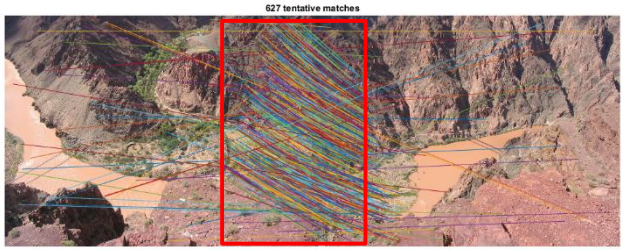
Image Alignment & Stitching using Conventional Methods

Good amount of overlap

Input

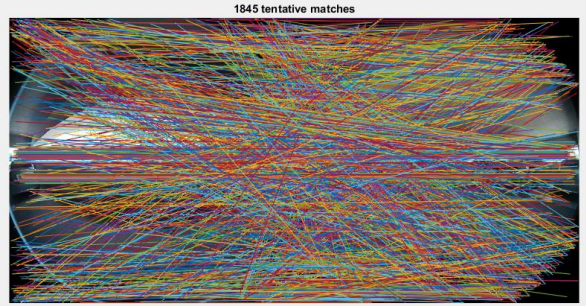


Very limited amount of overlap

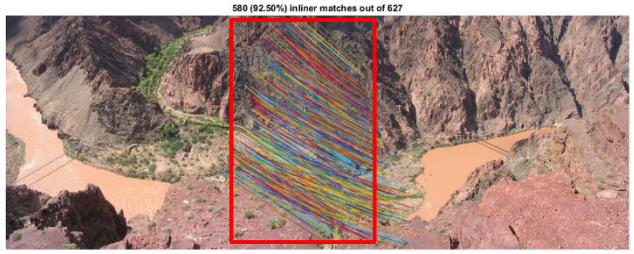


Inlier features outnumber outliers (thanks to the large overlap)

Feature extraction & matching (SIFT)

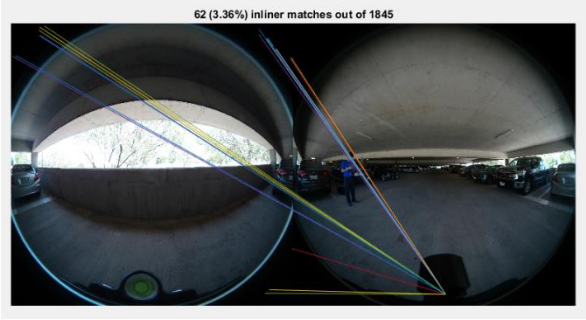


Overwhelming incorrect feature matches due to : small overlap area + fisheye extreme distortion at boundary



Inliers retained correctly (reliable)

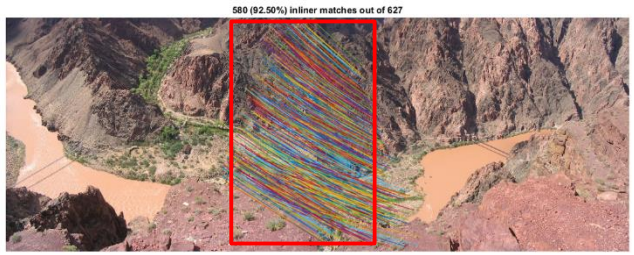
Outliers elimination (RANSAC)



Inliers, mistook as outliers, got removed (not reliable)

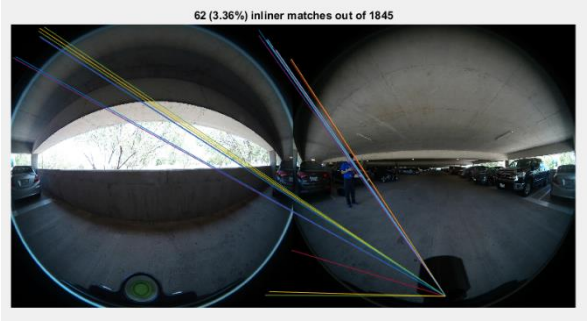
Challenges (cont.)

Image Alignment & Stitching using Conventional Methods



Inliers retained correctly (reliable)

Outliers elimination (RANSAC)



Inliers, mistook as outliers, got removed (not reliable)

Warp image to the homography estimated from the inliers.

Straighten the pano. Blend the overlap! Done!

Warp image

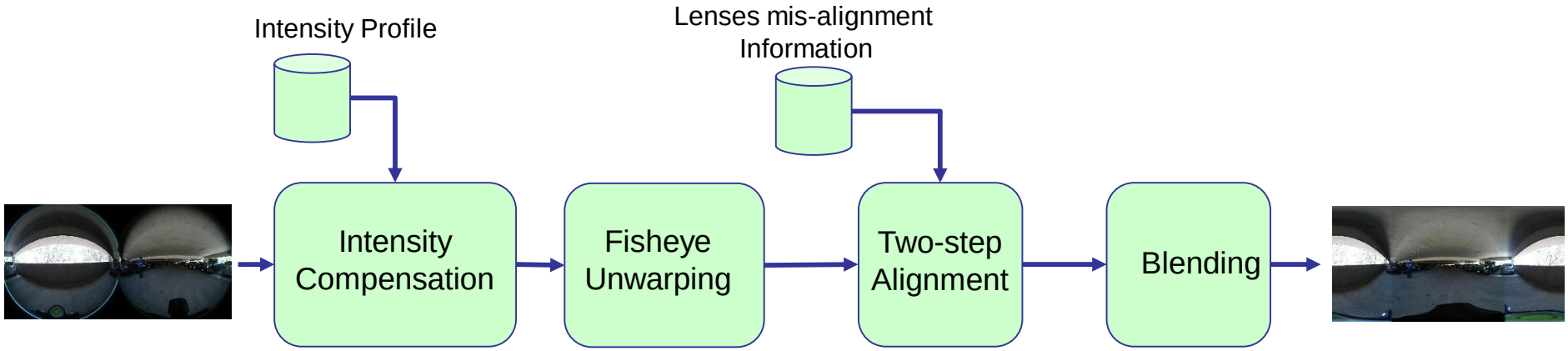
Not enough reliable inliers → cannot warp image

Gear360 Stitching Method

- Unconventional stitching method
- Specialized for  cameras
  - Individual lens characteristics
  - Relative difference between two lenses put on a same system
- Two-step alignment
  - Gear360-specific alignment, mostly computed offline, followed by
  - Heuristic (& adaptive) method for refined alignment, computed online

Our Approach

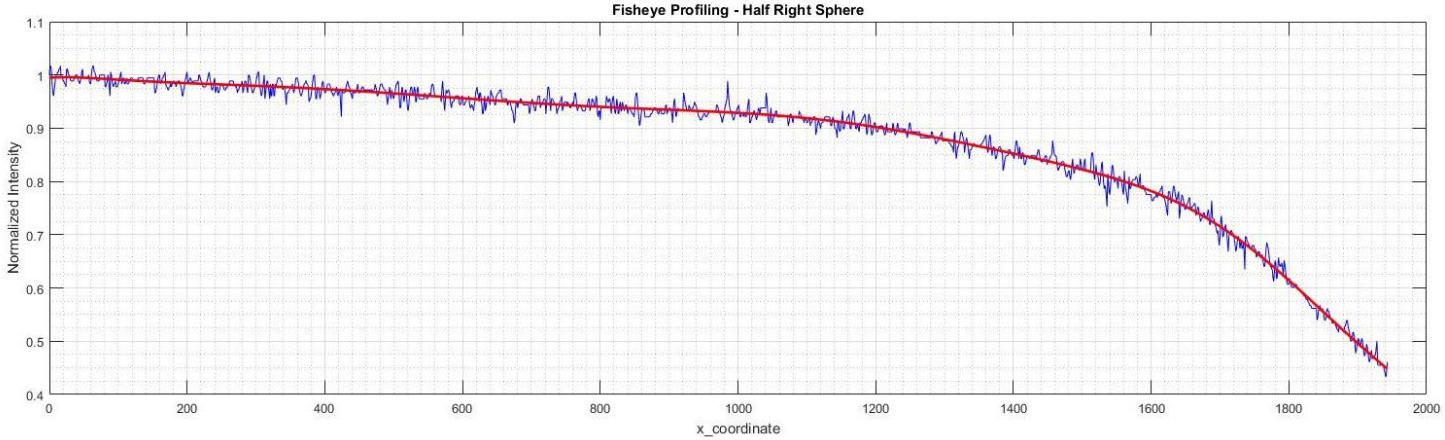
- Intensity compensation: for the light fall off away from the center
- Unwarping: Derive geometric transformation to unwarp the Gear360 fisheye images
- Alignment: Two-step alignment
- Blending: Ramp function blending (simple & fast)



Intensity Profiling

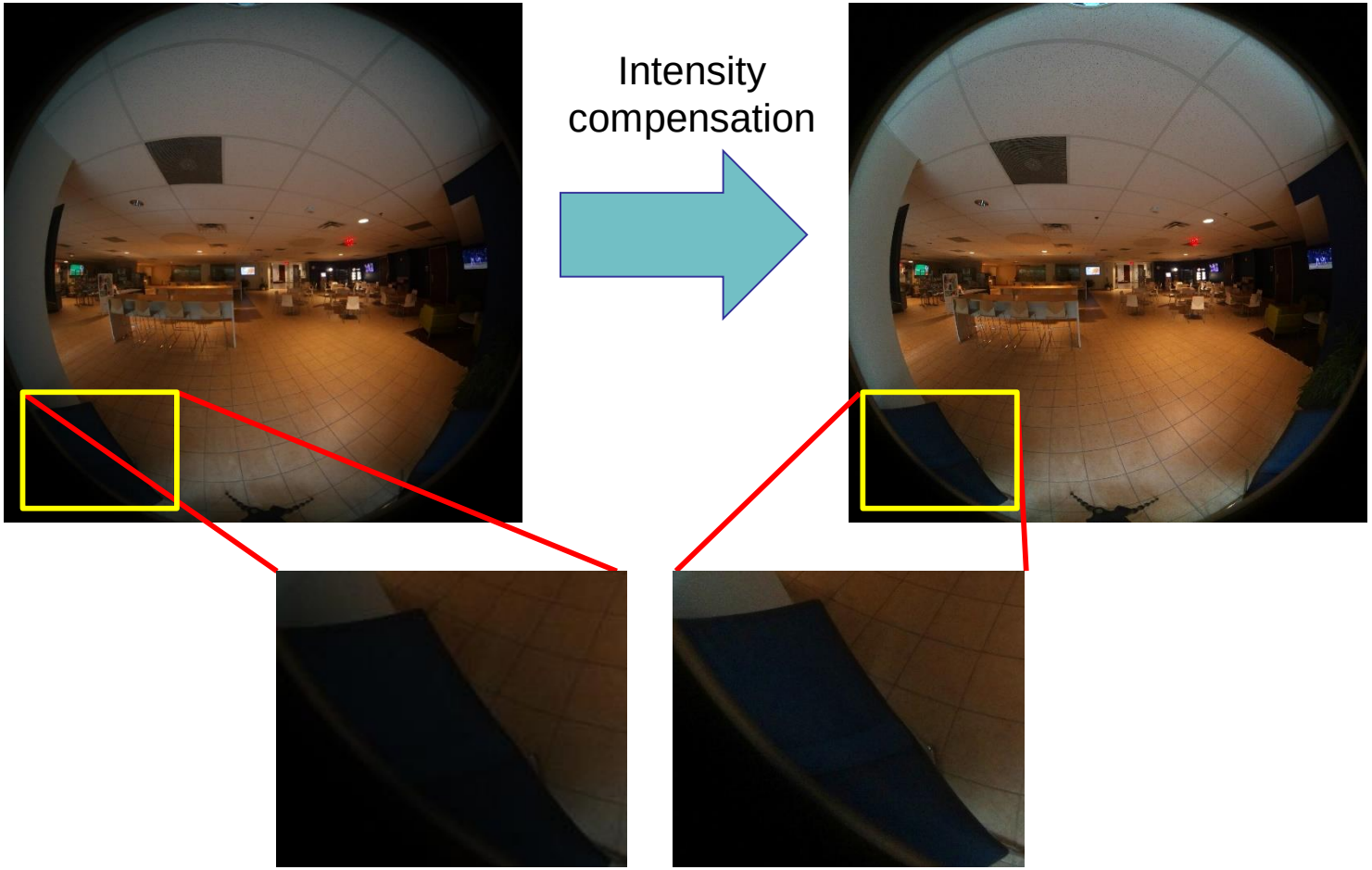
- The intensity as well as the incident lights get distorted toward the radius (fisheye lens's natural effect)
- Need to compensate for the light drop off → profiling experiment

Measure the intensity along the radius
Polynomial curve fit the data

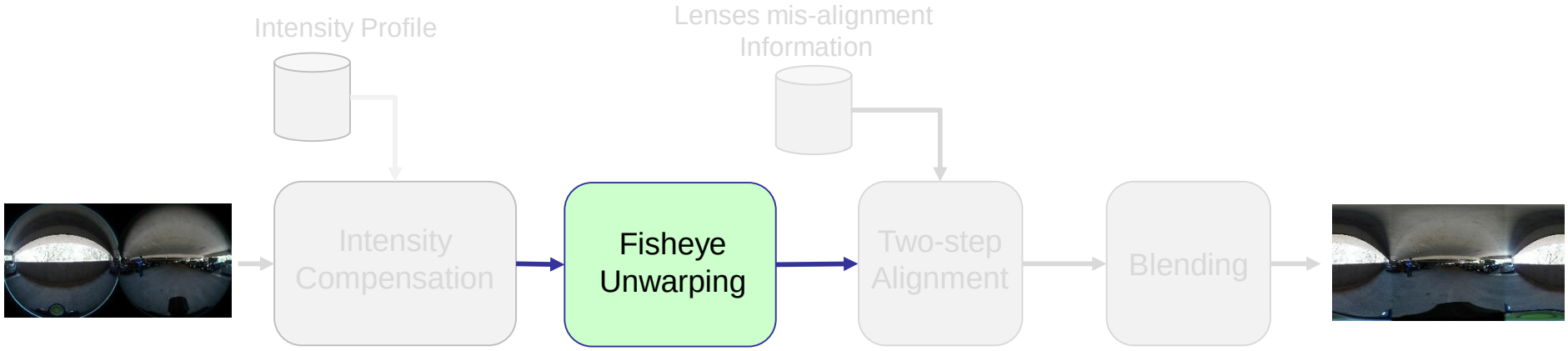


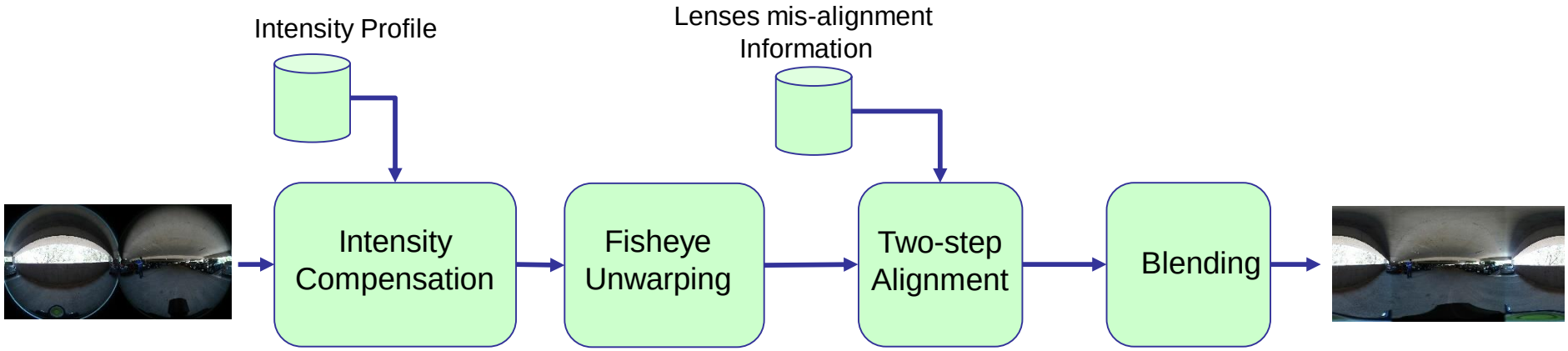
Intensity Profiling: Result

- Assume the optical symmetry of the fisheye lens
- Compensate the light fall off using the constructed intensity profile



Next


Intensity Profile
Lenses mis-alignment Information
Intensity Compensation
Fisheye Unwarping
Two-step Alignment
Blending


Intensity Profile
Lenses mis-alignment Information
Intensity Compensation
Fisheye Unwarping
Two-step Alignment
Blending

Unwarping: Fisheye Optical Model

- Fisheye lenses achieve larger than 180-degree field of view by bending the incident lights

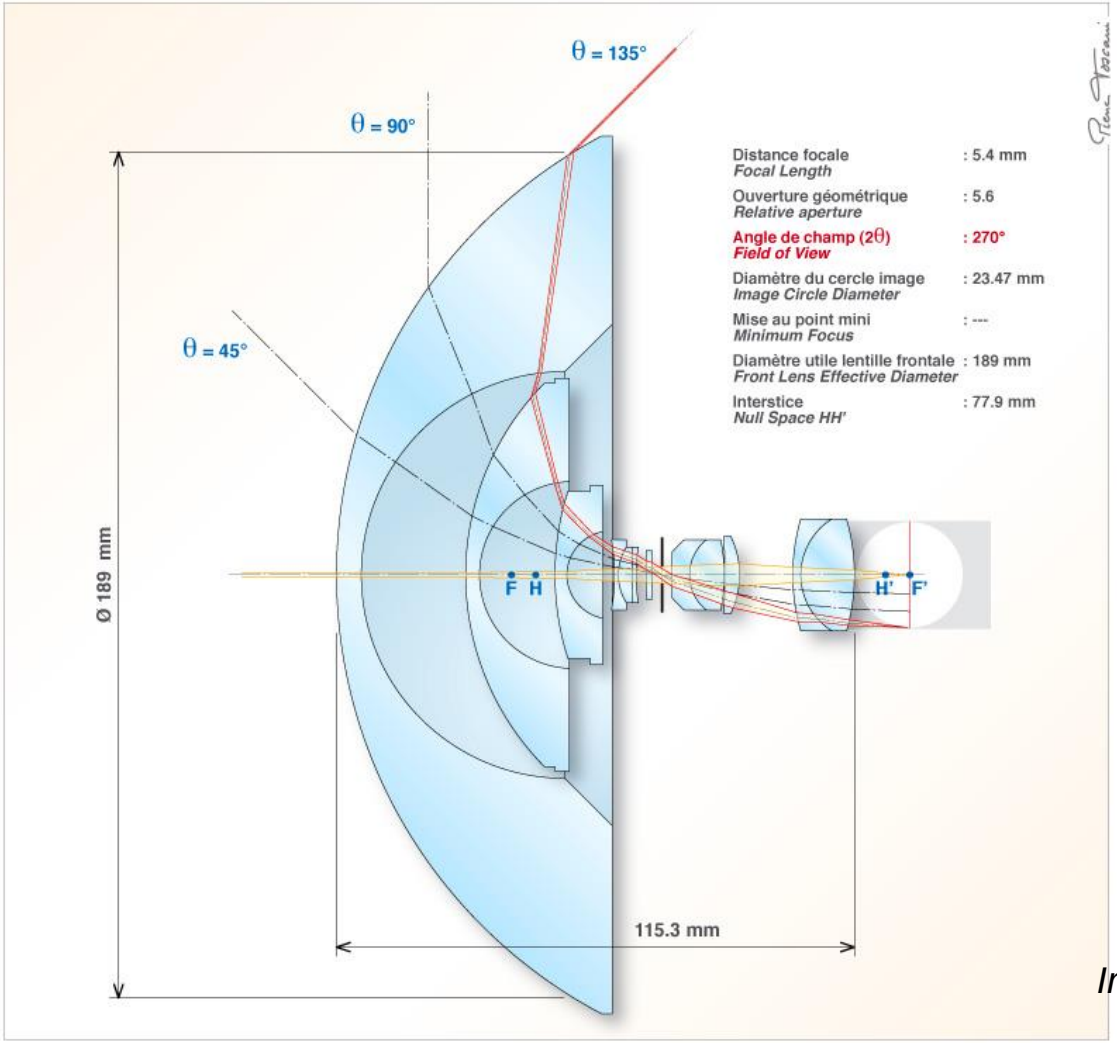


Image courtesy: Pierre Toscani

Unwarping: Fisheye Geometry Model

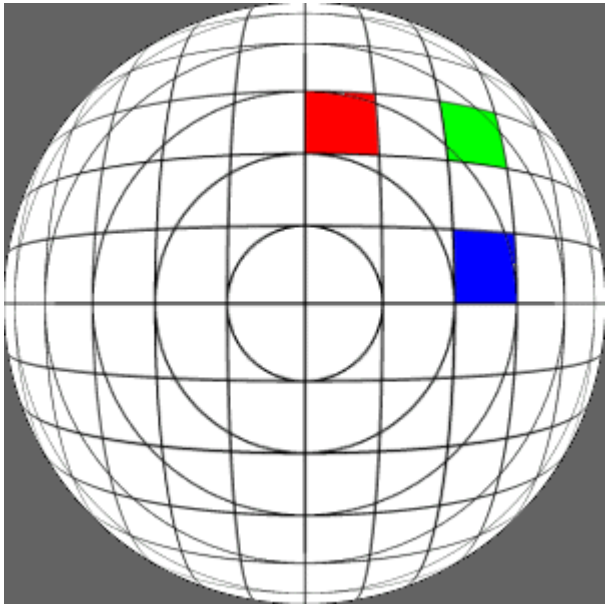
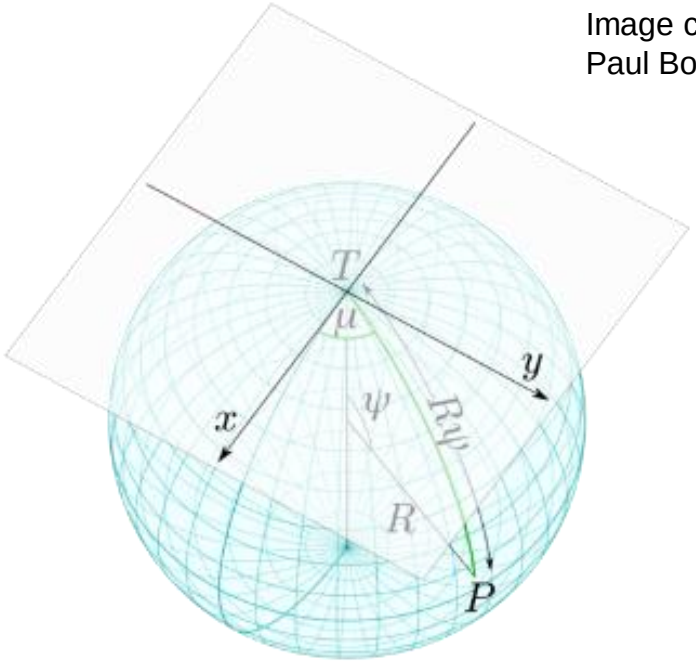


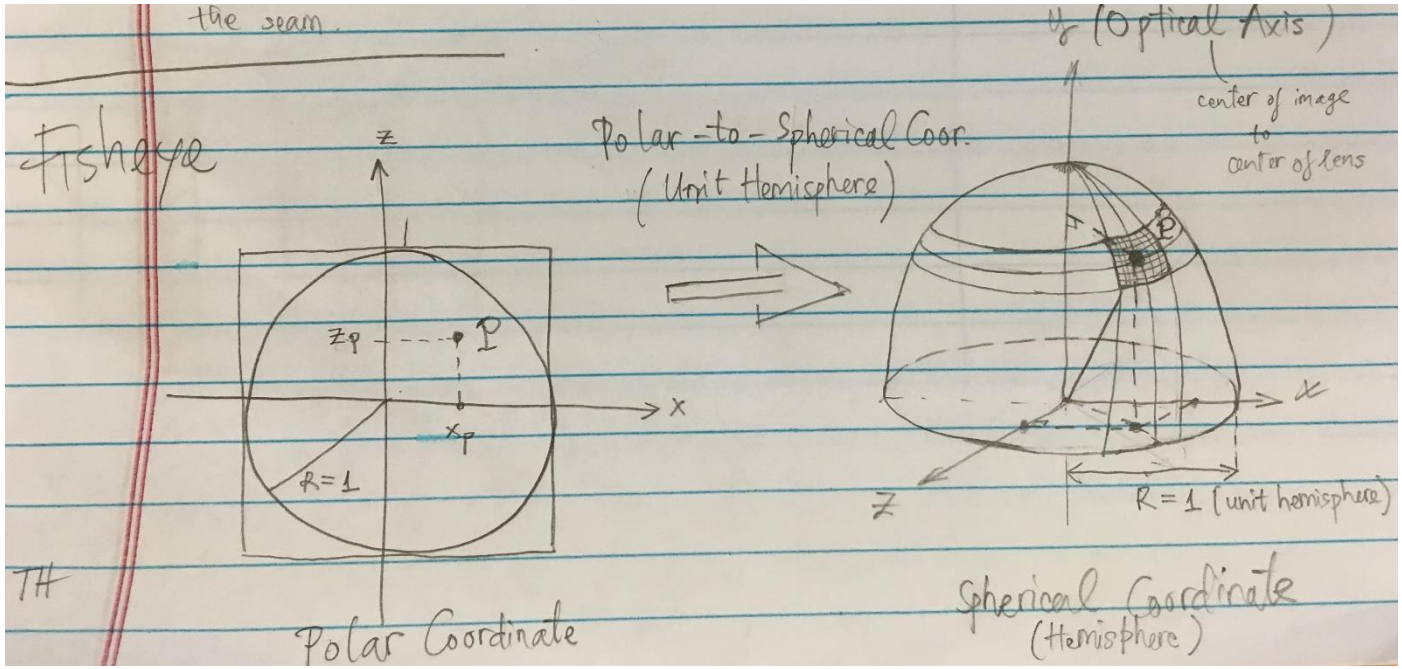
Image courtesy:
Paul Bourke



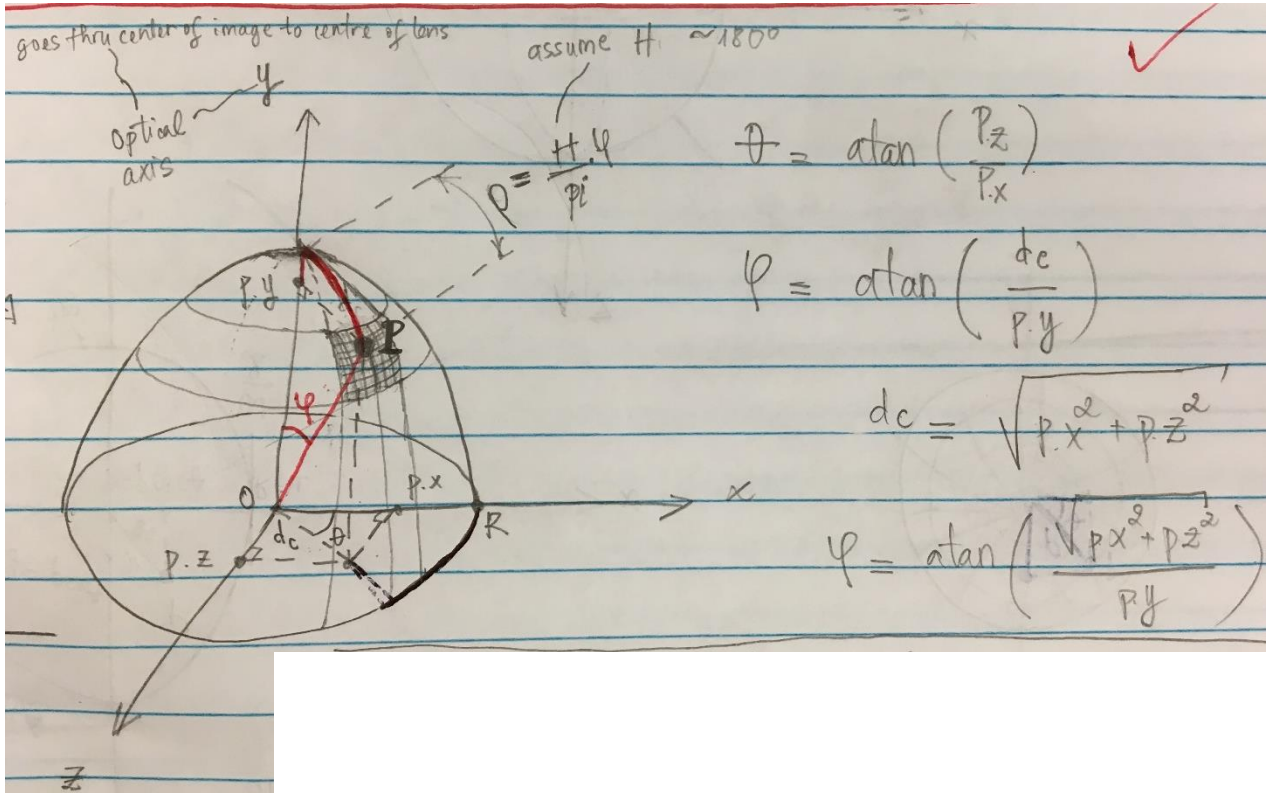
2D to 3D:
x-coordinate: yaw
y-coordinate: pitch

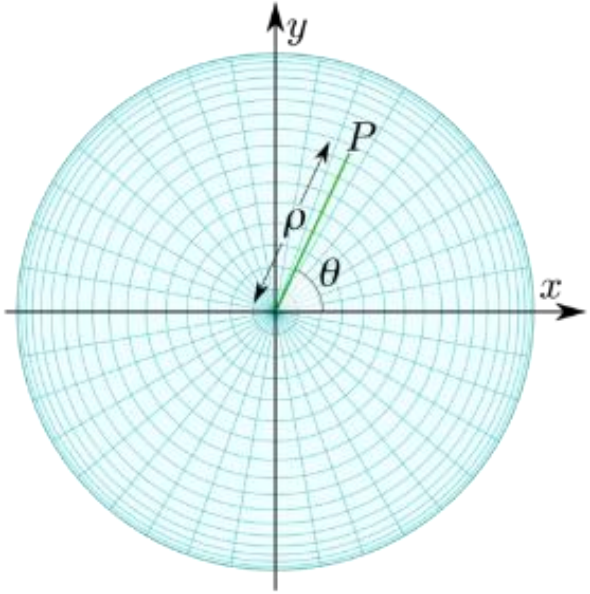
Unwarping: derivation

- Treat the original fisheye image (2-D) in a unit polar coordinate
- Reconstruct the 3-D surface structures in a unit sphere
- Map the 3-D into 2-D: preserves distant (straight lines → straight lines), and compatible with 360 viewer



Unwarping: derivation





$$x\_prj = \rho * cos(theta)$$
$$y\_prj = \rho * sin(theta)$$

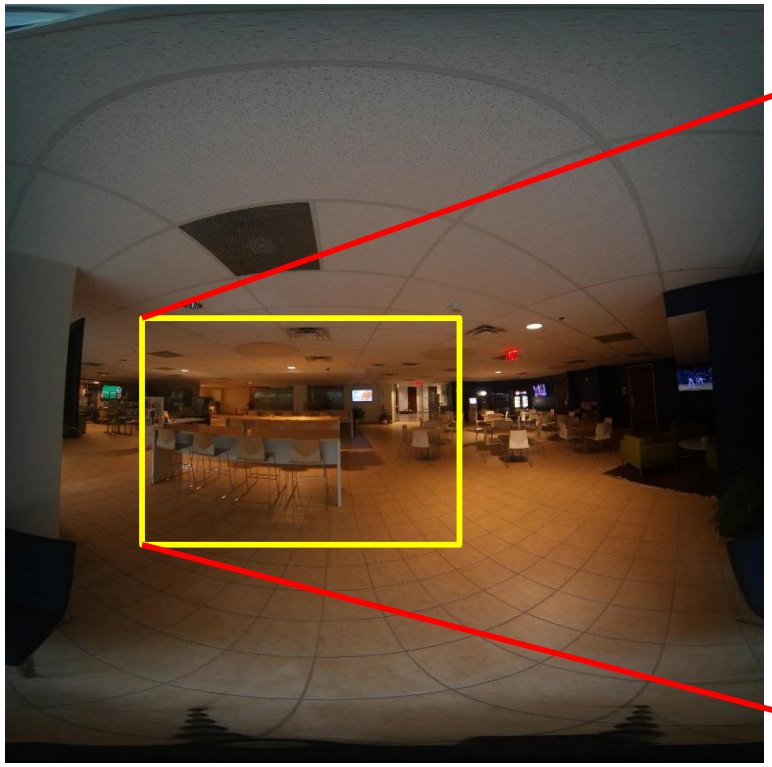
Unwarping: Result

Unwarp

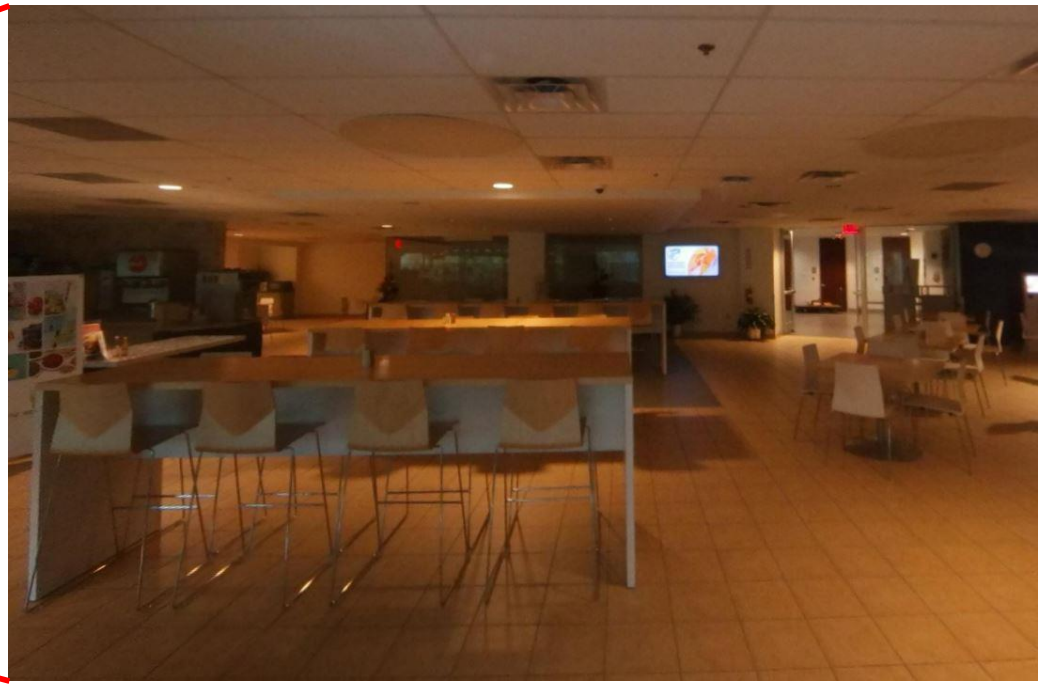
View Port on PTGui 360 Viewer

View Port on PTGui 360 Viewer

Unwarping: Result



Display on non-360 viewer



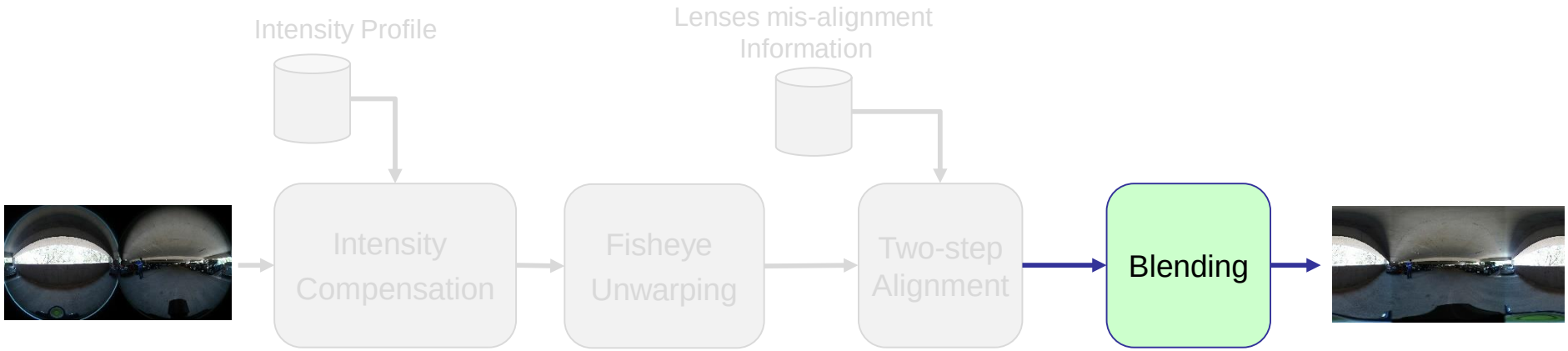
View Port on PTGui 360 Viewer

Unwarp the light-compensated images

Original (output of Gear360)

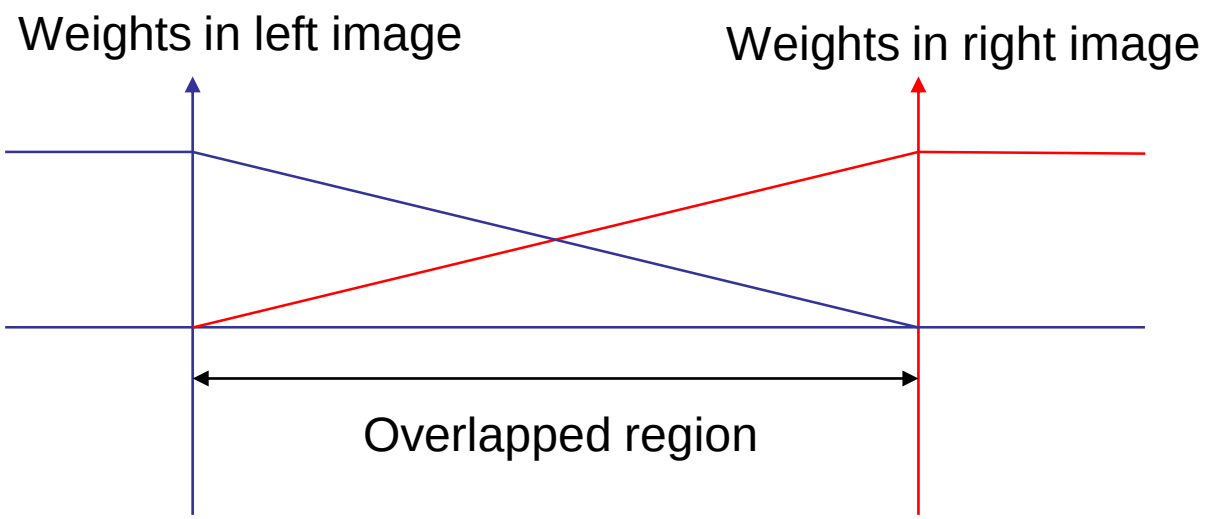
Light compensated & Unwarped

Next


Intensity Profile
Lenses mis-alignment Information
Intensity Compensation
Fisheye Unwarping
Two-step Alignment
Blending

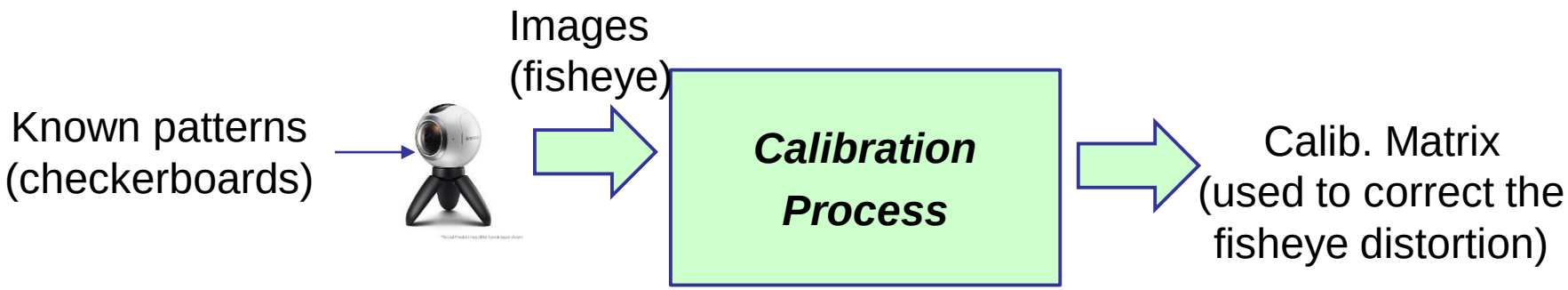
Blending

- First blending version: ramp blending (fast)



Fisheye Lens Calibration

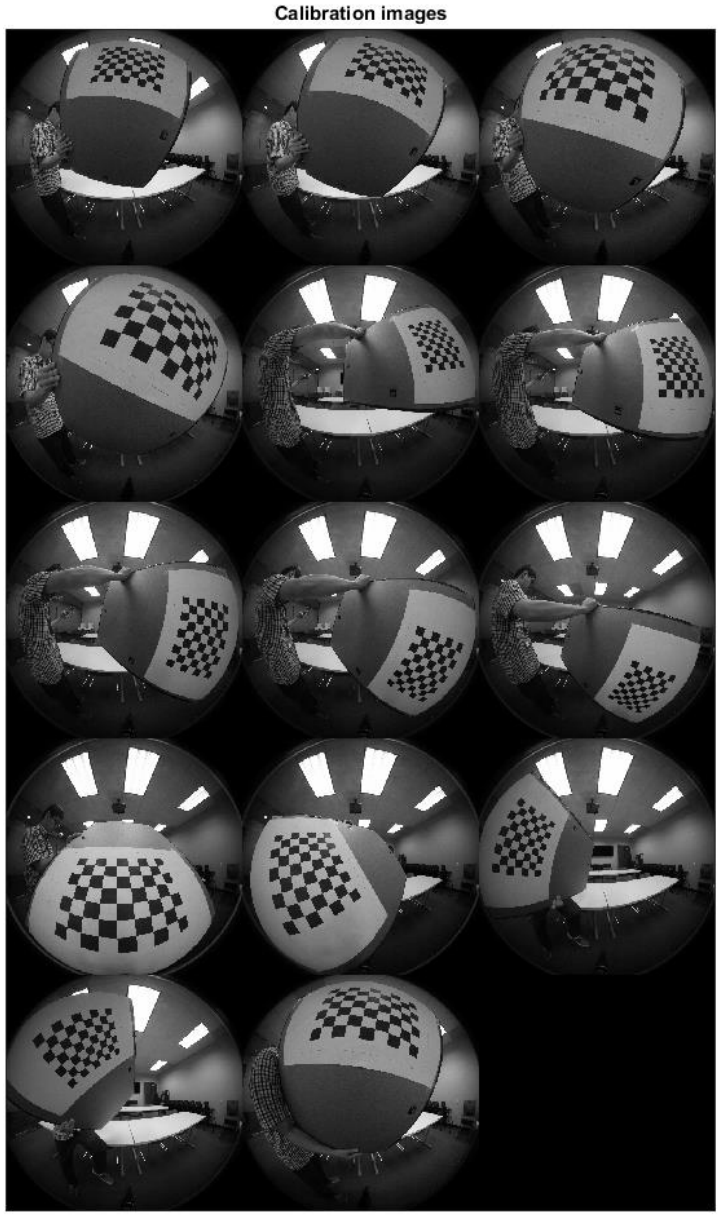
- Study the lens' optic distortion
- Goal: make sure if there is any distortion, we can compensate to make straight lines straight, mid points mid, etc.



- Calibration: using OCamCalib_v3.0 for fisheye lens with FOV > 180 degree. (OpenCV Calib. may not work with fisheye lens whose FOV > 180 deg)

Calibration Experiments

- Checkerboard patterns with predefined size.
- Cover all rotational angles of the fisheye lens.
- Approach the lens as much as possible (without creating visible shadow)
- Get all checkboard corners detected for all images → fail: take different set of images & start over.
- The image set on RHS has all corners detected correctly.



Calibration Results

- Affine matrix shows that the lens is accurate (affine parameters at 5th, 6th decimal place after zero.

$$A = \begin{bmatrix} 1.000028 & , 0.000062 & , 0 \\ -0.000006 & , 1 & , 0 \\ 1957.061640, & 1943.391170, & 1 \end{bmatrix}$$

- Translation from center: deviated from center:
  - $1957.061640 - (3888/2) = 13.0616$ pixels horizontally
  - $1943.391170 - (3888/2) =$ -0.6088 pixel vertically

- Thus, no need for individual lens compensation

Affine Transformation

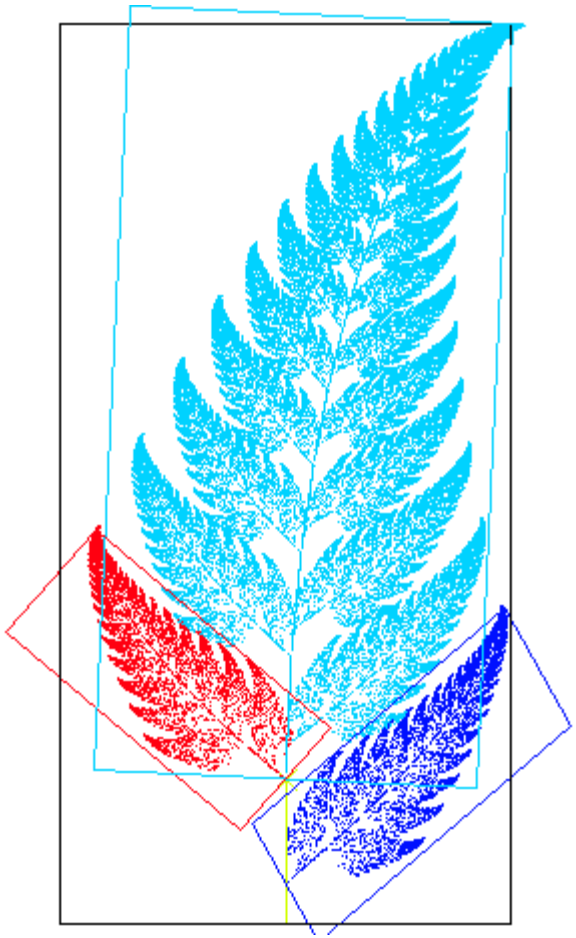


Image courtesy: wiki

Geometric transformation:

- Change coordinate of each image pixel (to map them to new places)
- Image intensity remains intact

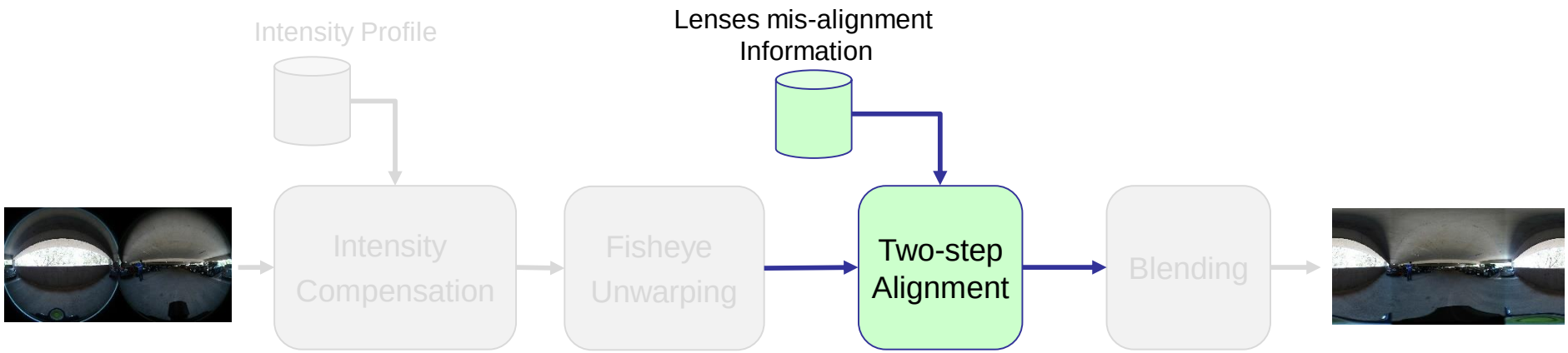
Nice features of Affine Transformation:

- Preserve lines, points & planes → preserve shapes
- Fast to derive: need two sets of correspondent points

Affine Transform	Example	Transformation Matrix	
Translation		$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix}$	t_x specifies the displacement along the *x* axis t_y specifies the displacement along the *y* axis.
Scale		$\begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$	s_x specifies the scale factor along the *x* axis s_y specifies the scale factor along the *y* axis.
Shear		$\begin{bmatrix} 1 & sh_y & 0 \\ sh_x & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	sh_x specifies the shear factor along the *x* axis sh_y specifies the shear factor along the *y* axis.
Rotation		$\begin{bmatrix} \cos(q) & \sin(q) & 0 \\ -\sin(q) & \cos(q) & 0 \\ 0 & 0 & 1 \end{bmatrix}$	q specifies the angle of rotation.

Courtesy: Mathworks

Next

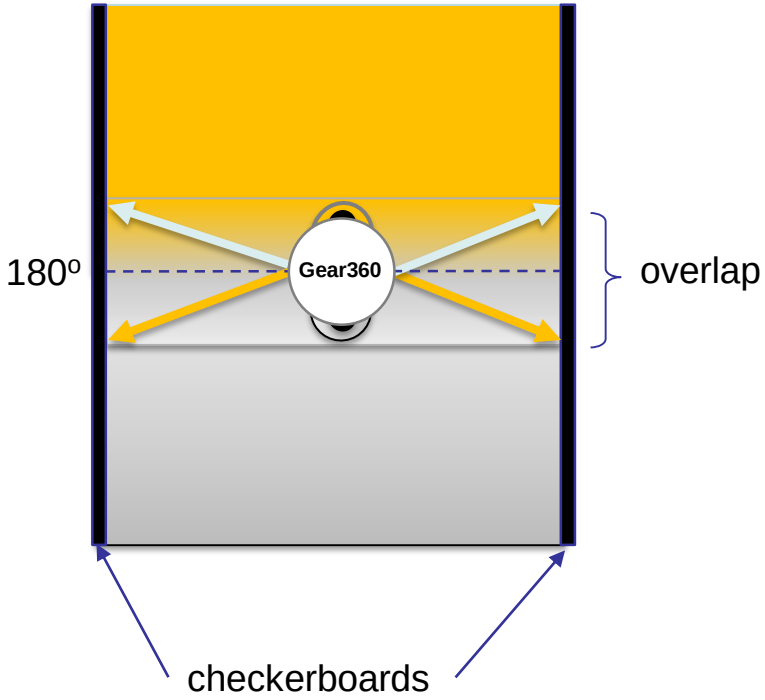

Intensity Profile
Lenses mis-alignment Information
Intensity Compensation
Fisheye Unwarping
Two-step Alignment
Blending

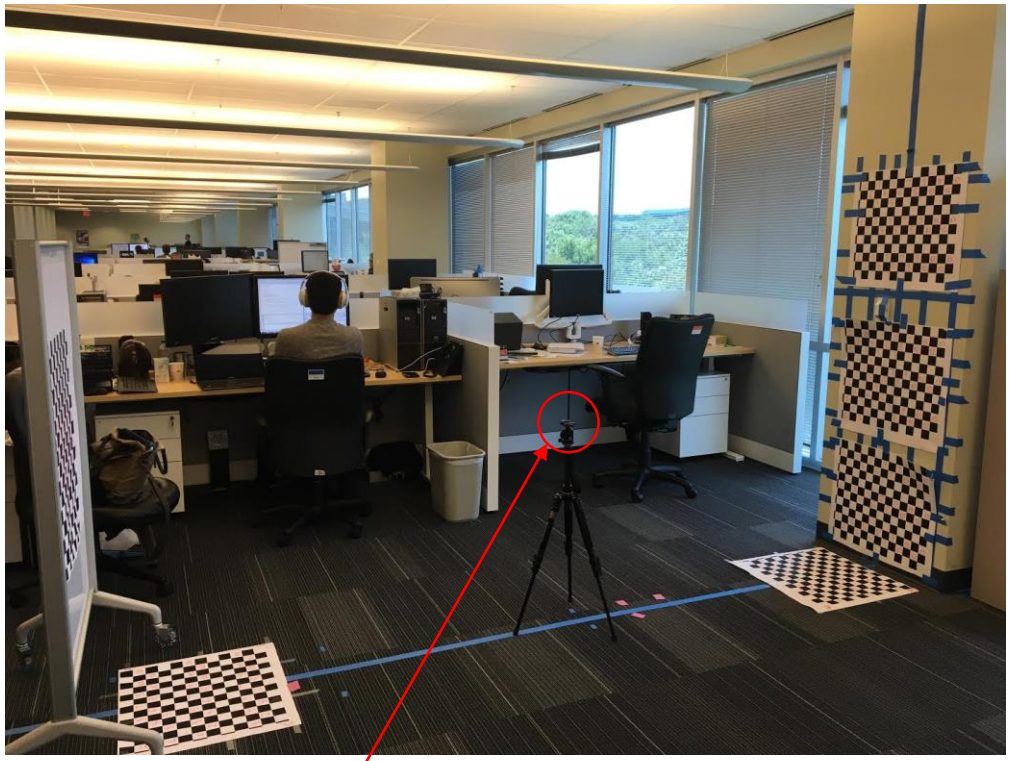
Two-Step Alignment

- Compensate for the relative mis-alignment between two lenses
  - Estimate points correspondence for a possible alignment [manually] [offline]
- Adaptively align images to minimize any small & remnant discontinuity in the overlapping regions after the first alignment [automatically] [online]

Lens Alignment

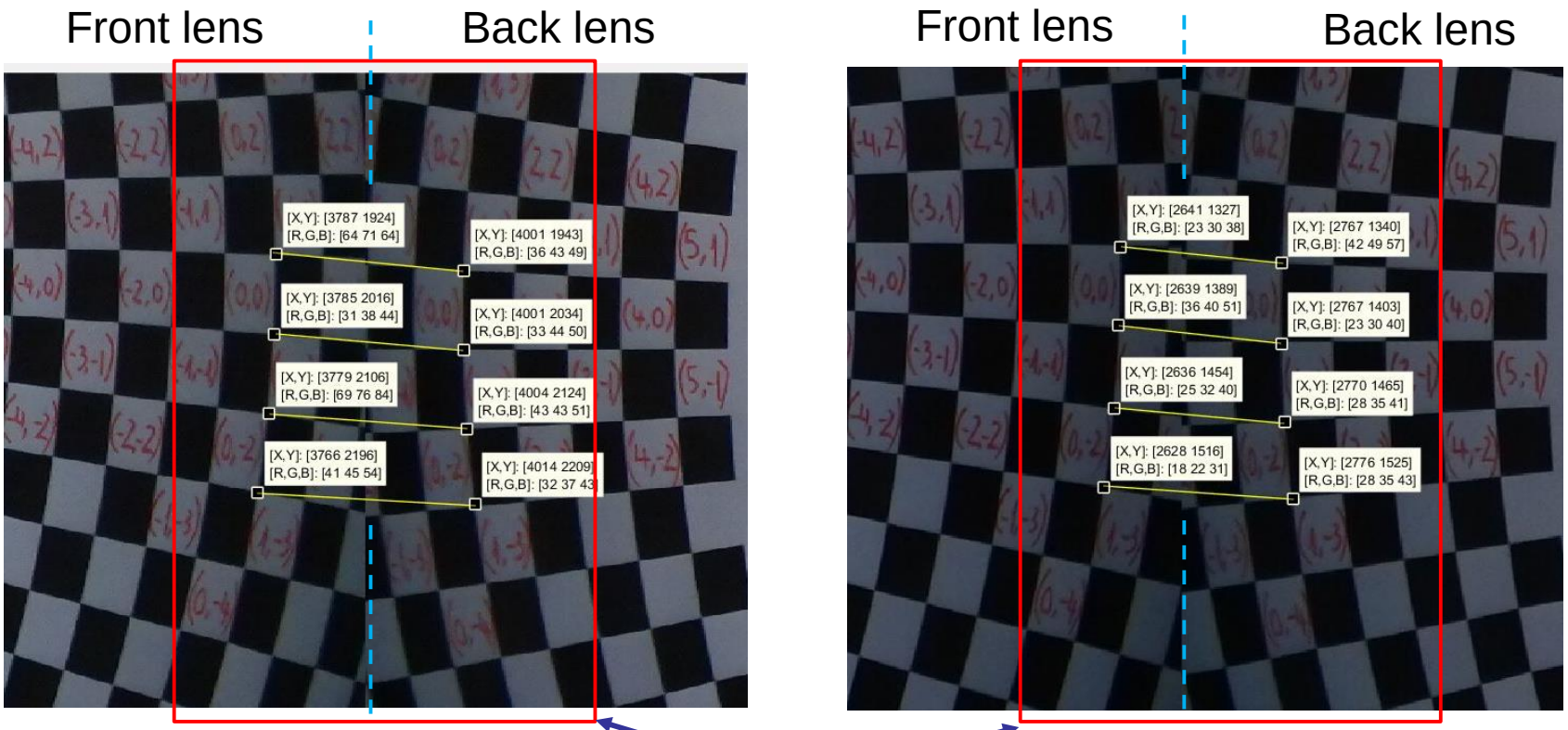
- How much the images produced by two lenses differentiate geometrically?
- Use checkboards to find the mis-alignment patterns at overlapping regions





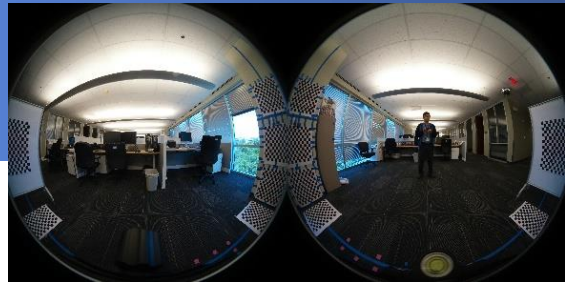
Each of all test  cameras is put here

Mis-alignment Pattern in Overlapping Regions

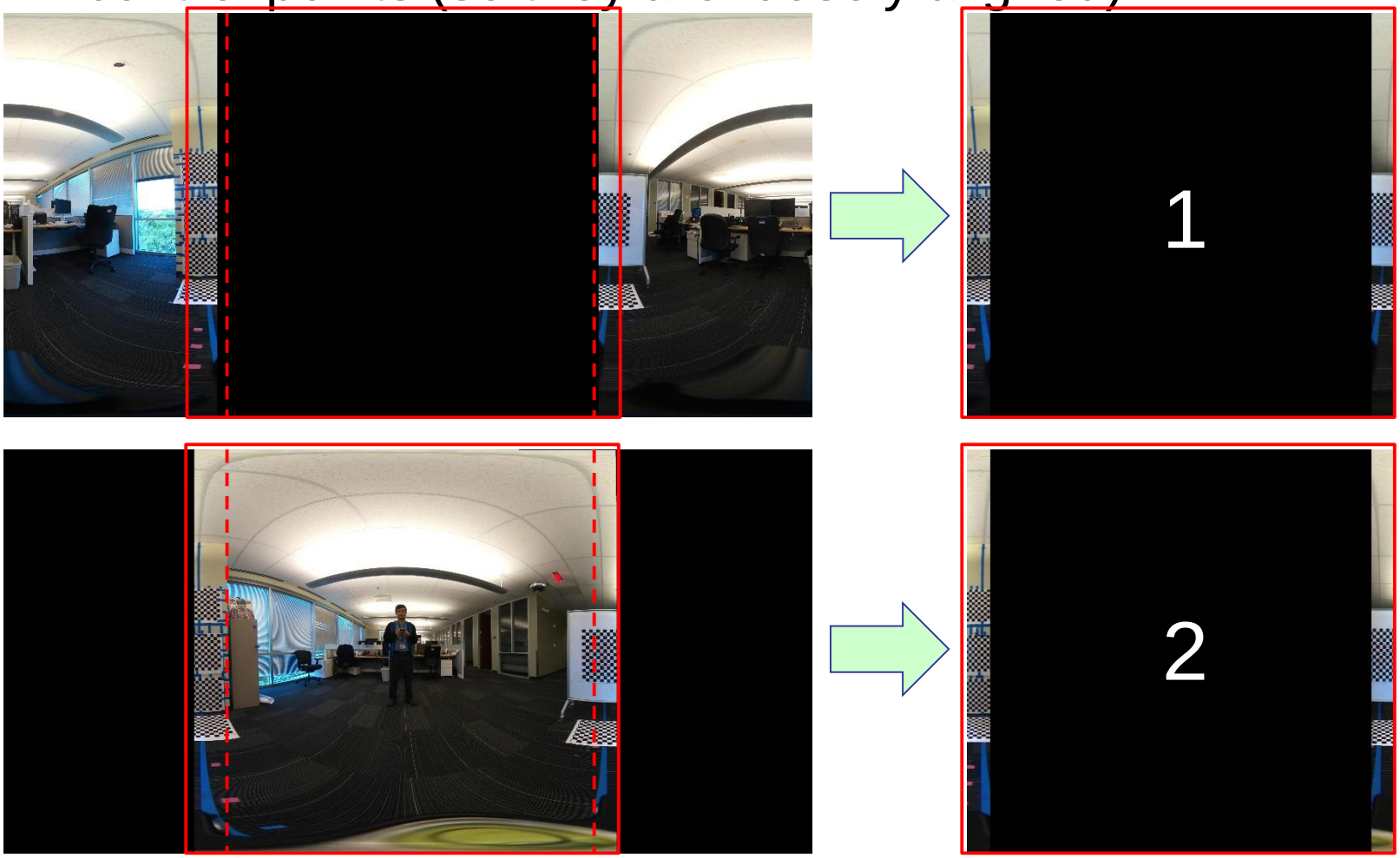


- Absolute coordinate of pixel coordinates (e.g. checkerboard square) can vary from camera to camera.
- Relative position of same checkerboard squares in the overlapping regions remains similar/same among different cameras.

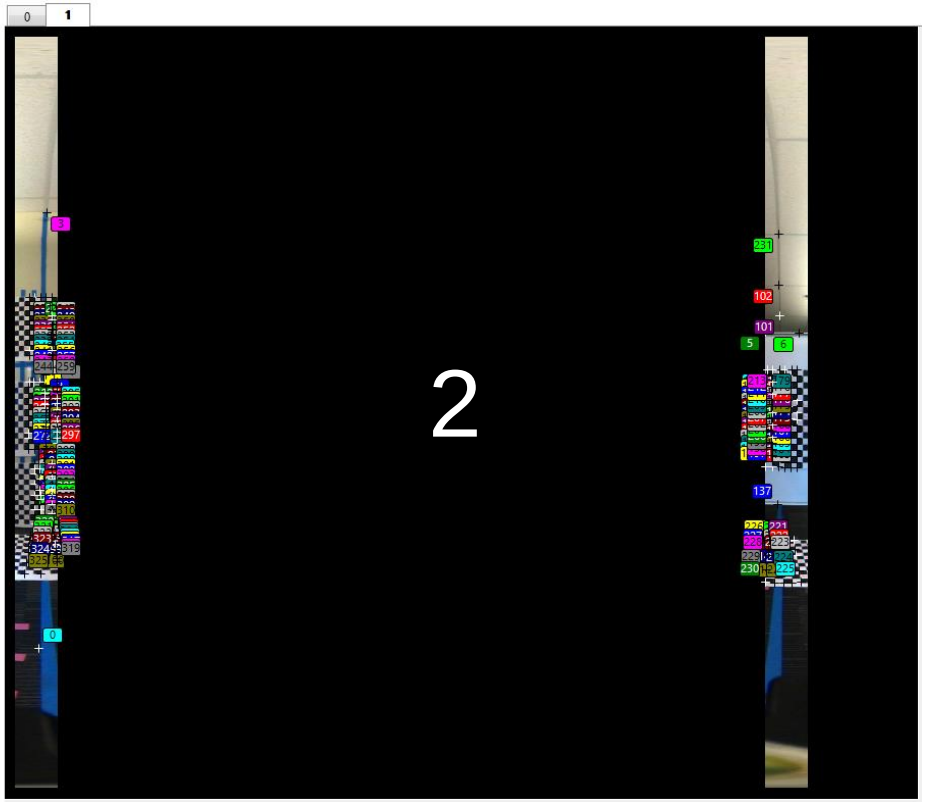
Lens Alignment



- Unwarp the fisheyes
- Arrange the images in a 360x180 pano before extract control points (so they are loosely aligned)



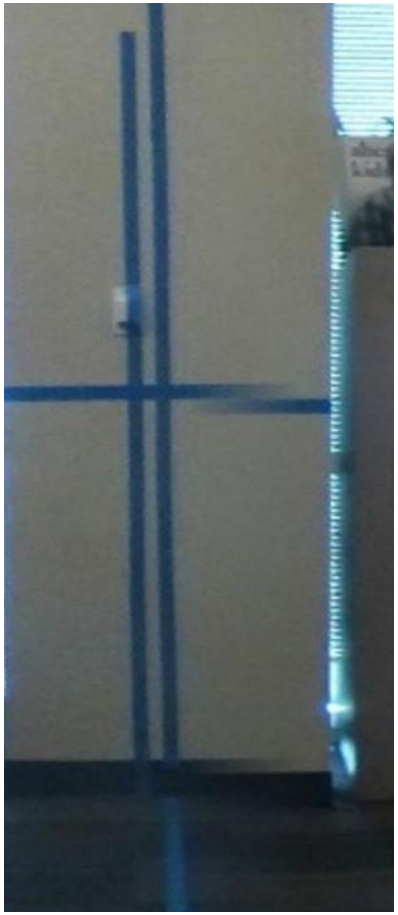
Lens Alignment



- Control points selection → set of correspondent pairs (~200 pairs)
- Solve for an affine matrix [computed offline]
- Align: warp (apply affine transformation on) one image to another

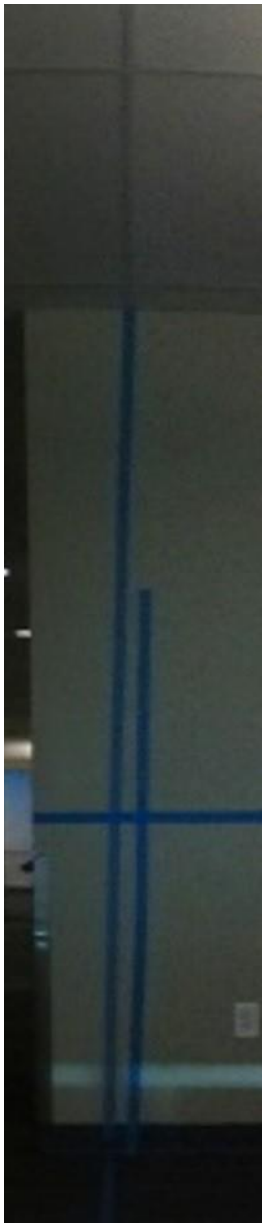
Lens Alignment

- WITHOUT Alignment



Lens Alignment

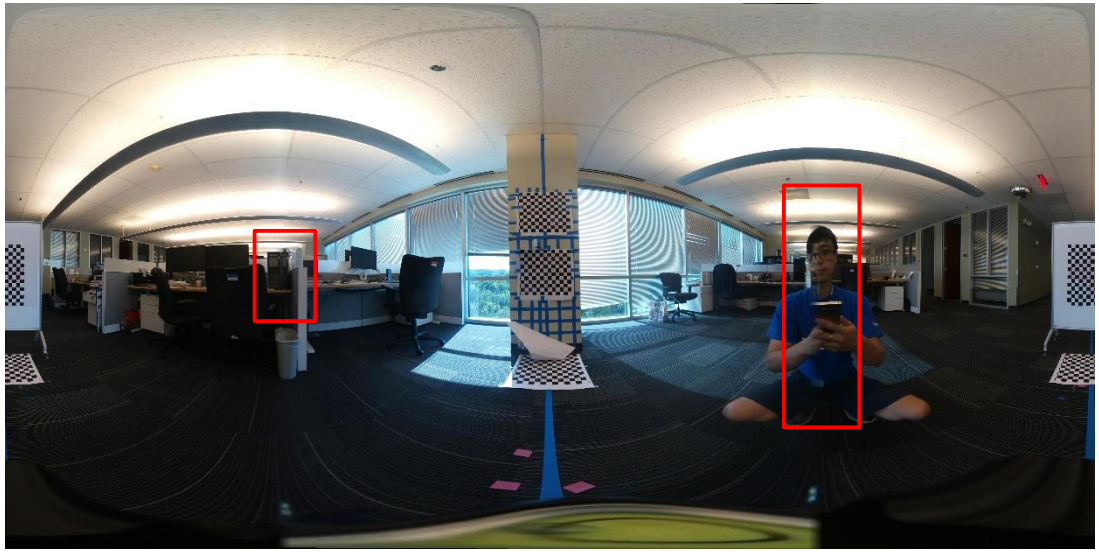
- With the proposed Alignment



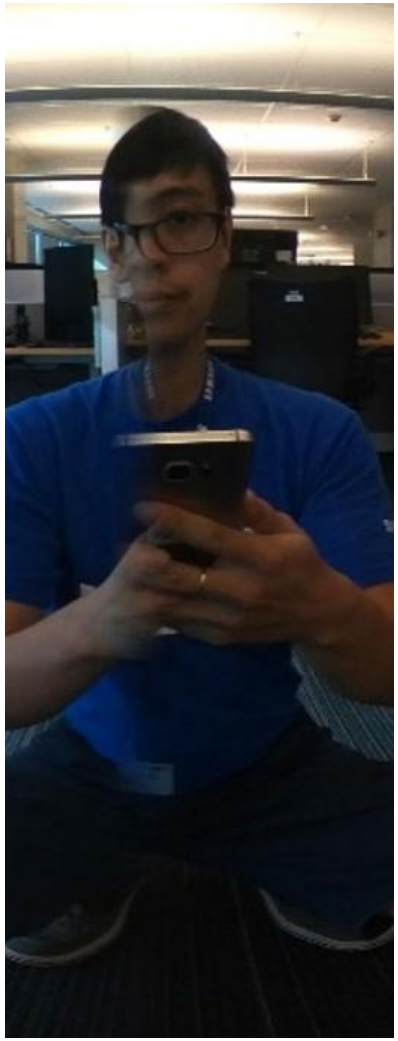
- The estimated affine matrix makes both images align vertically

Refined Alignment

- Objects/persons too close to camera
- Need more than the first alignment method



The person sitting ~ 1m to the camera



The first alignment is already applied.

Refined Alignment

- There is possible mis-matches after the first alignment
  - This mis-alignment builds up when objects move closer to camera
  - Use normalized cross-correlation to find the best match adaptively to the scene & objects, thus minimized any mis-alignment caused by object's distance to camera

- Correlation review
  - Measures the similarity of two signals
  - Two functions matched when their cross-correlation maximized
- Variation in exposure & lenses at the overlapped regions → normalized cross-correlation

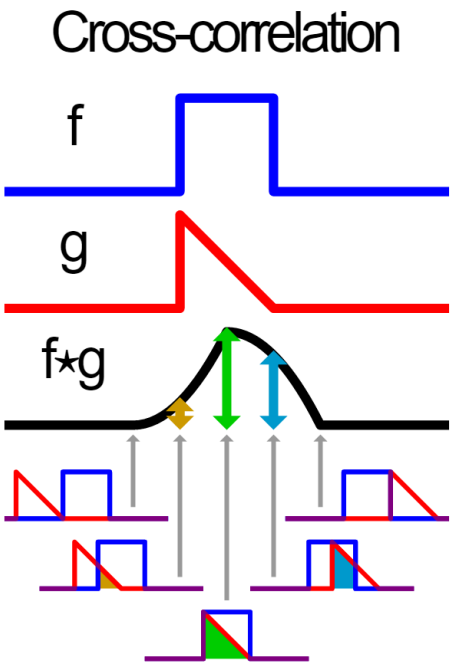


Image courtesy: wiki

Refined Alignment

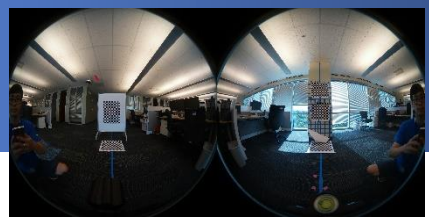
- Normalized cross-correlation [J. P. Lewis95] (old but fast & works well):

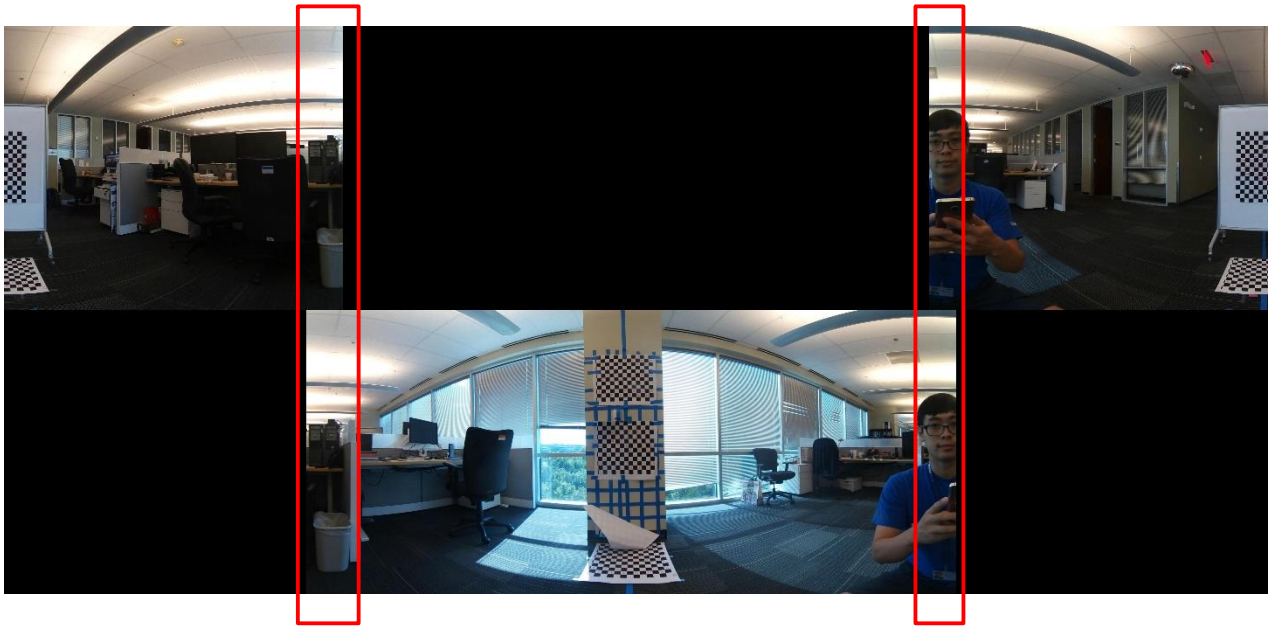
$$\gamma(u,v) = \frac{\sum_{x,y}\left[f(x,y)-\bar{f}_{u,v}\right]\left[t(x-u,y-v)-\bar{t}\right]}{\left\{\sum_{x,y}\left[f(x,y)-\bar{f}_{u,v}\right]^2\sum_{x,y}\left[t(x-u,y-v)-\bar{t}\right]^2\right\}^{0.5}}$$

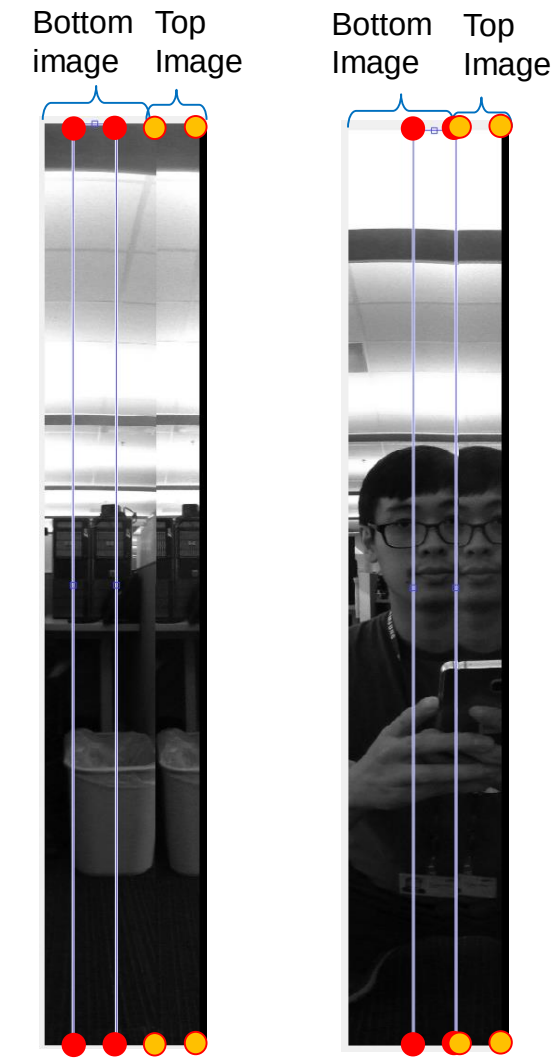
where

- f is the image.
- $\bar{t}$ is the mean of the template
- $\bar{f}_{u,v}$ is the mean of $f(x,y)$ in the region under the template.

Refined Alignment







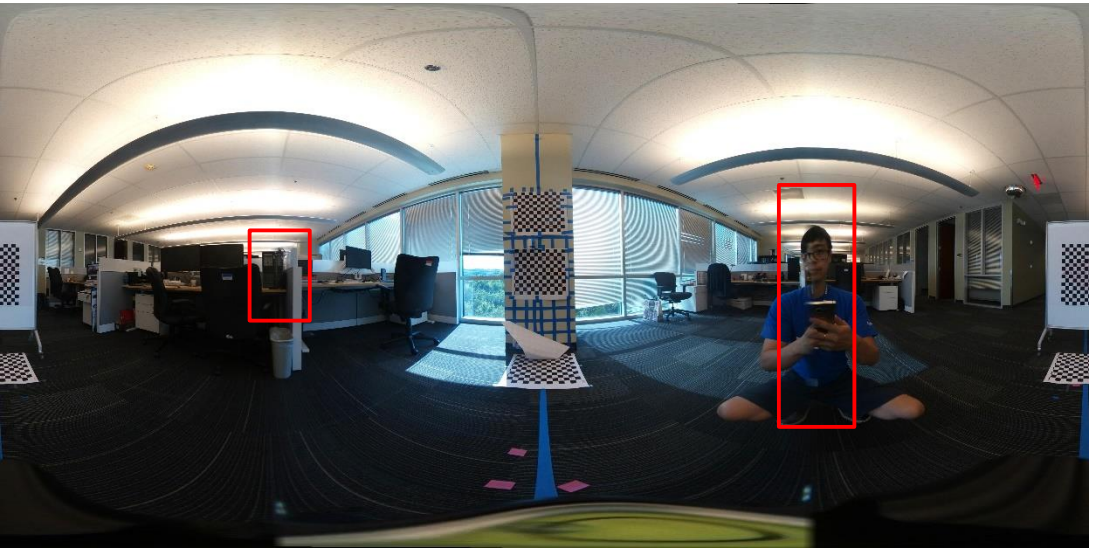
Match found!

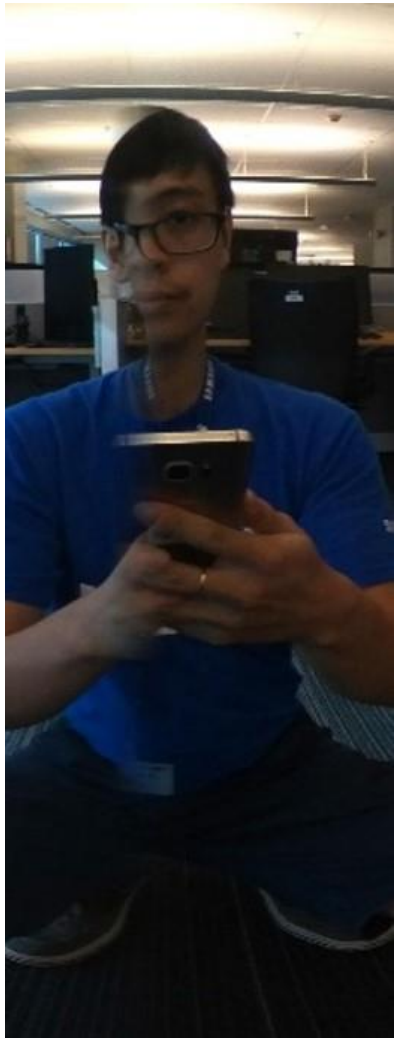
Match found!

- Find a best match of the templates of the top image in the references of the bottom one.
- Create 8 pairs of correspondent points → solve for an affine matrix [computed online]
- Warp image accordingly using this matrix.

Refined Alignment

- WITHOUT Refined Alignment *

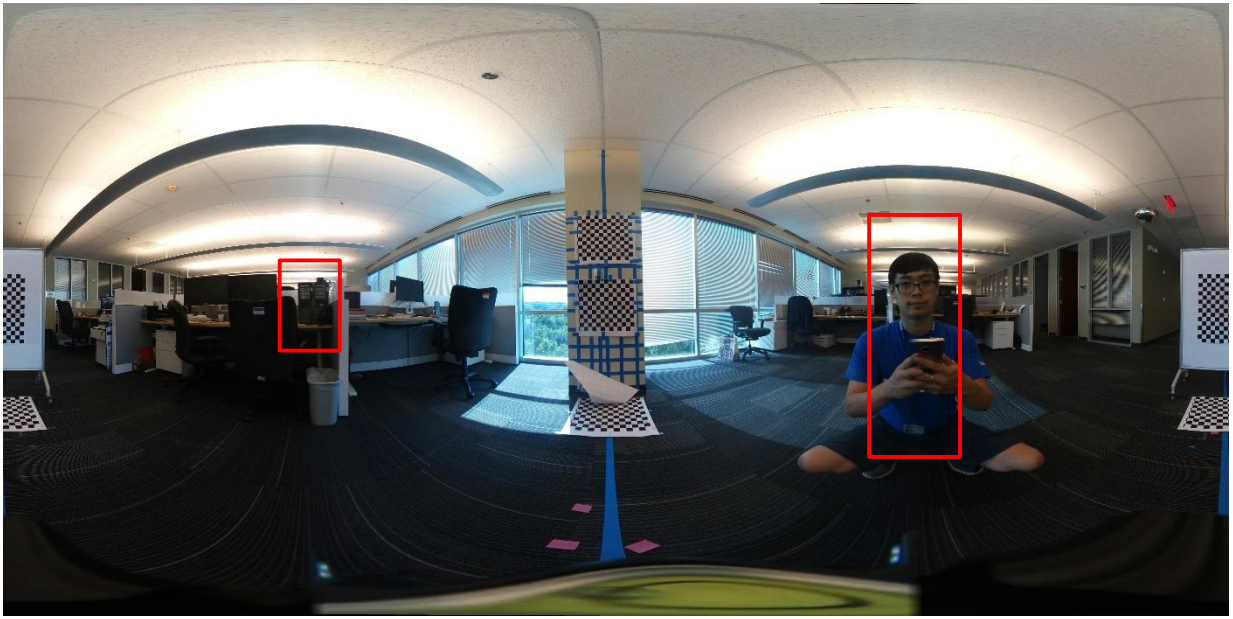


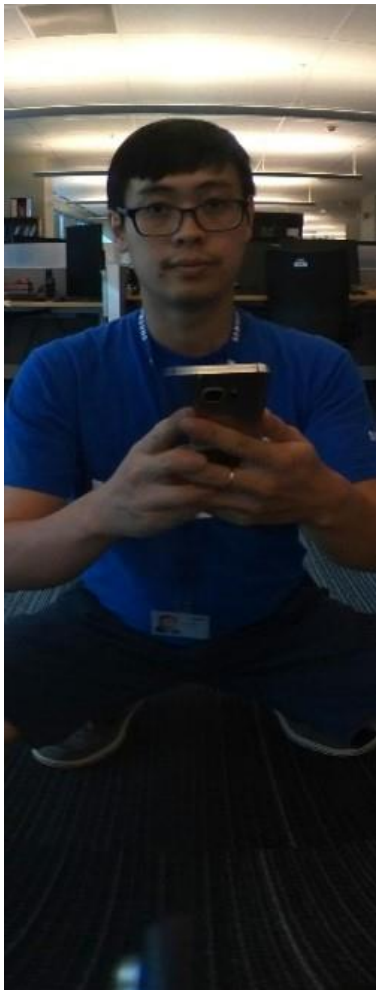


The first alignment is already applied.

Refined Alignment

- With the Proposed Refined Alignment *





On top of the first alignment.

Results

- Evaluate the Samsung Note-5's stitching algorithm vs. the proposed method.

Results (Phone vs. Proposed)

Stitched by Samsung Note-5

Results (Phone vs. Proposed)

Stitched by the Proposed Method

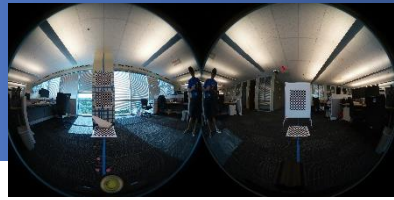
Results (Phone vs. Proposed)

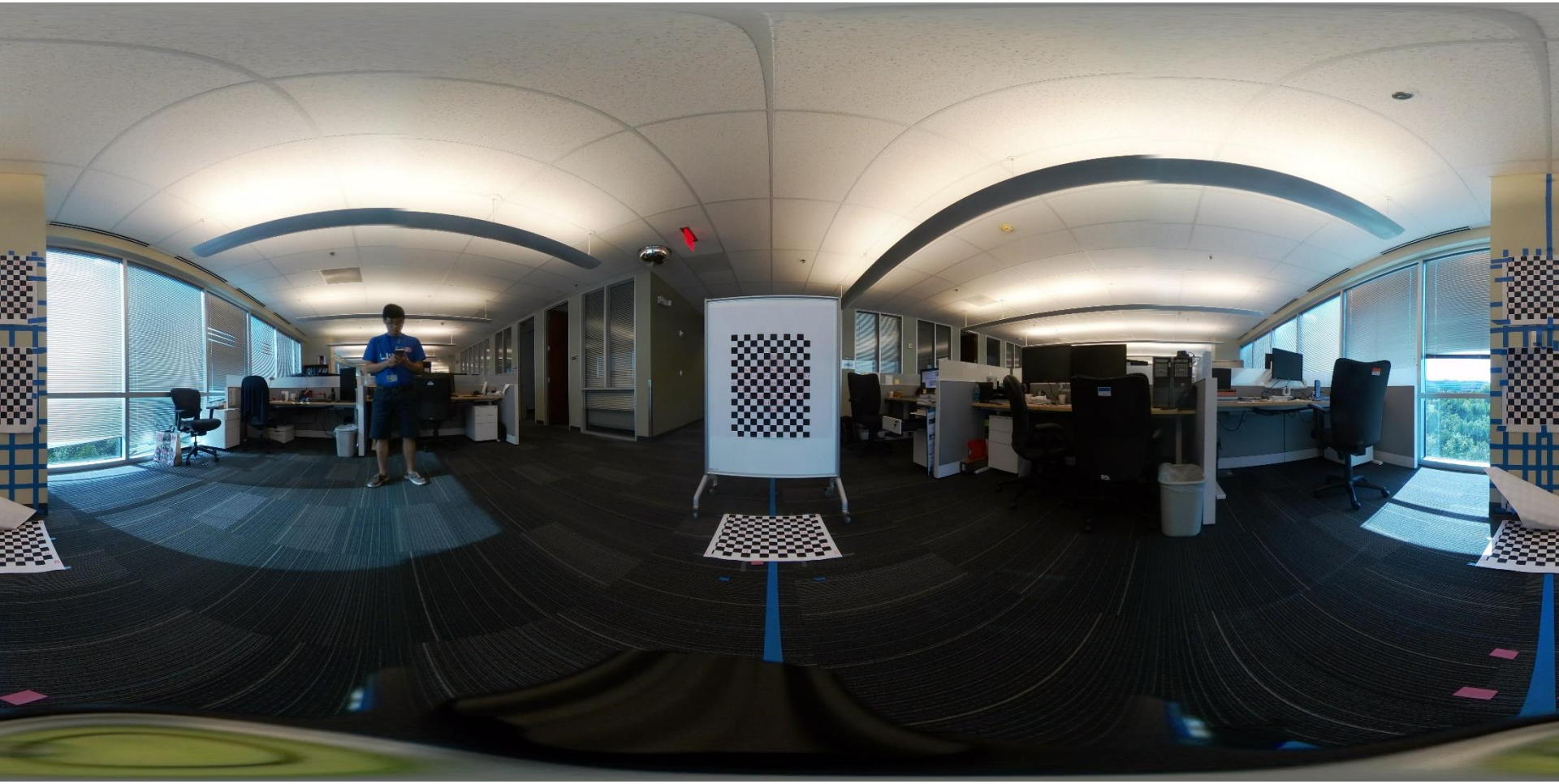
Stitched by Samsung Note-5

Results (Phone vs. Proposed)

Stitched by the Proposed Method

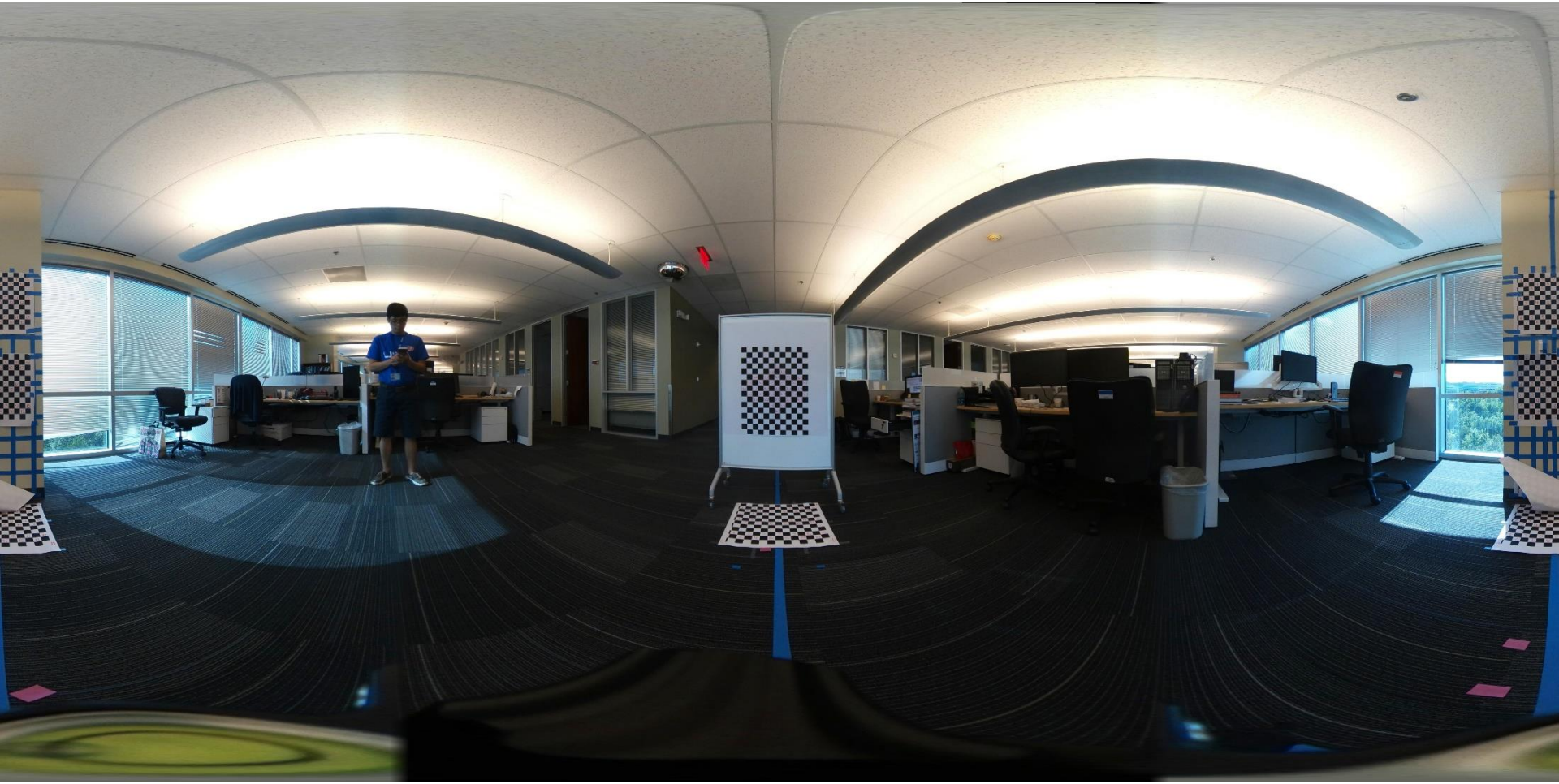
Results (Phone vs. Proposed)





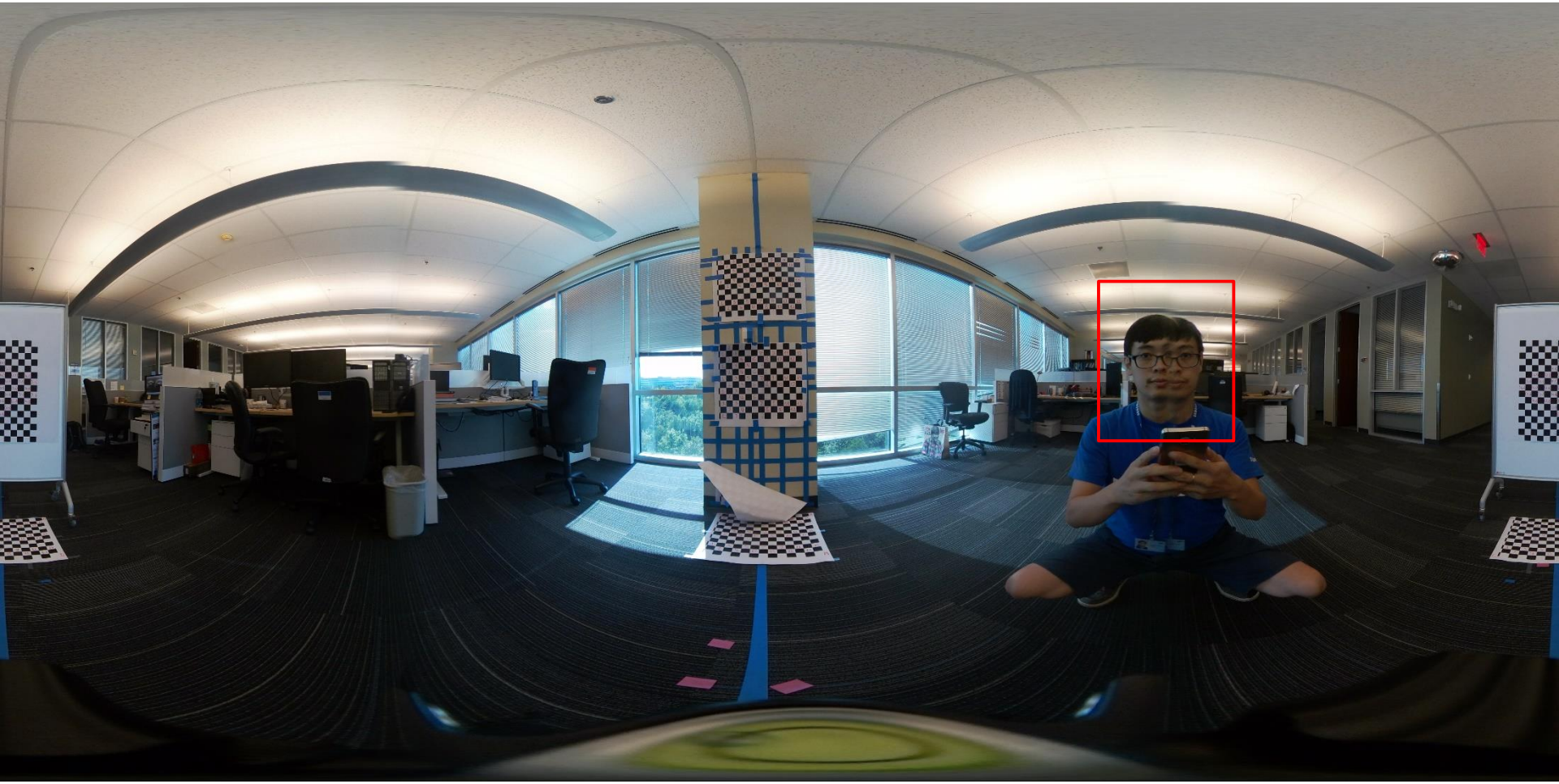
Stitched by Samsung Note-5

Results (Phone vs. Proposed)



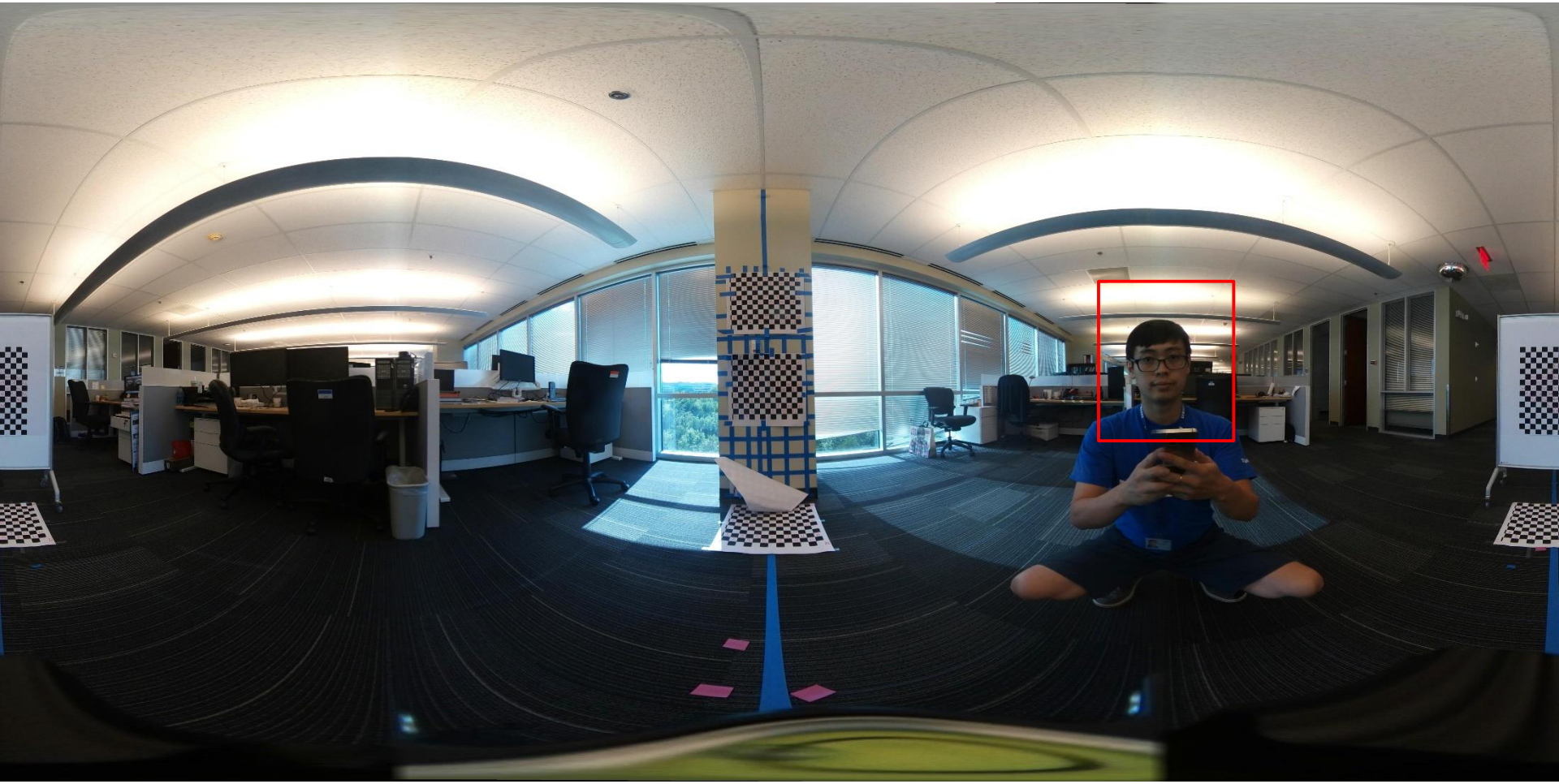
Stitched by the Proposed Method

Results (Phone vs. Proposed)



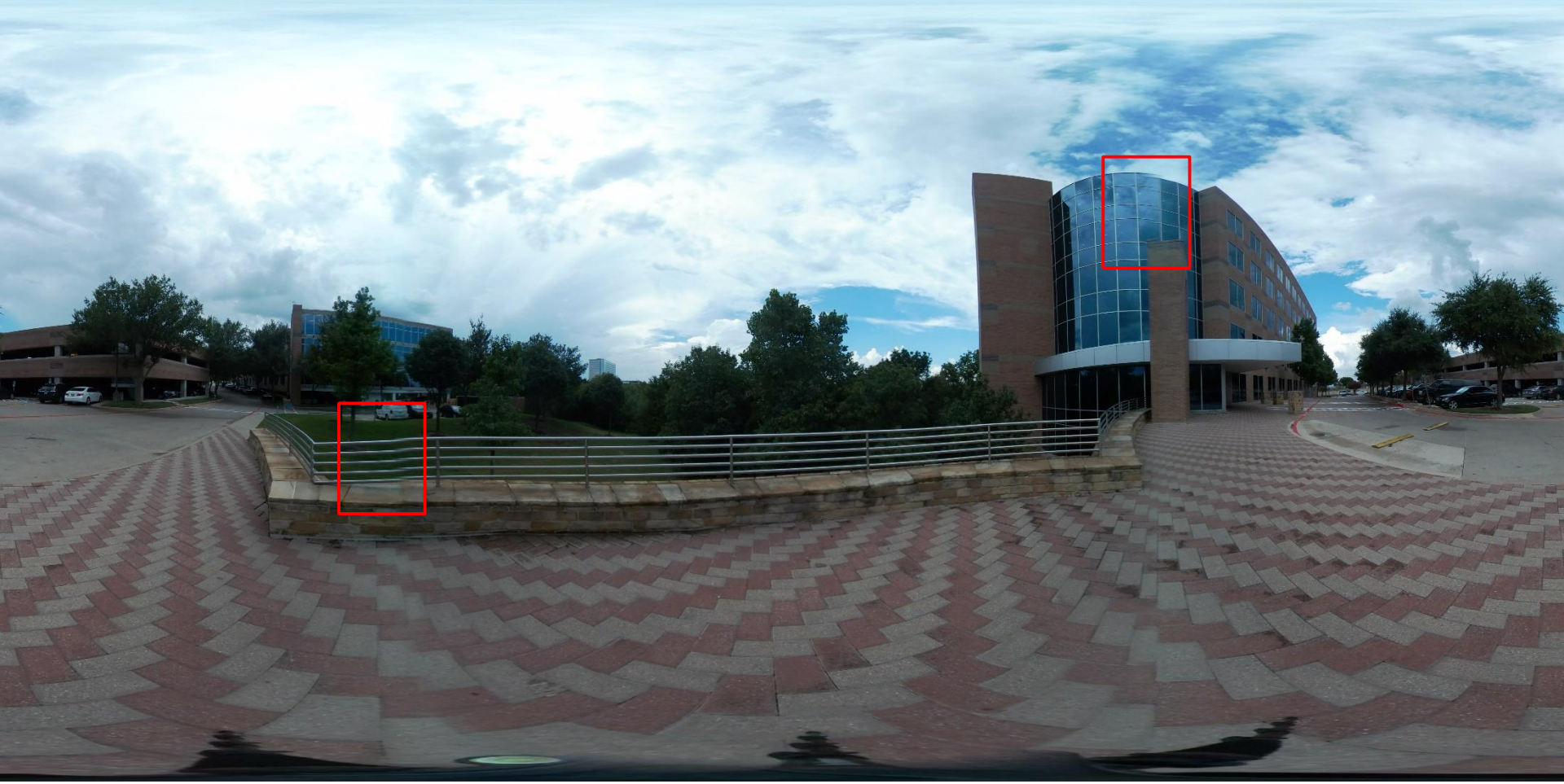
Stitched by Samsung Note-5

Results (Phone vs. Proposed)



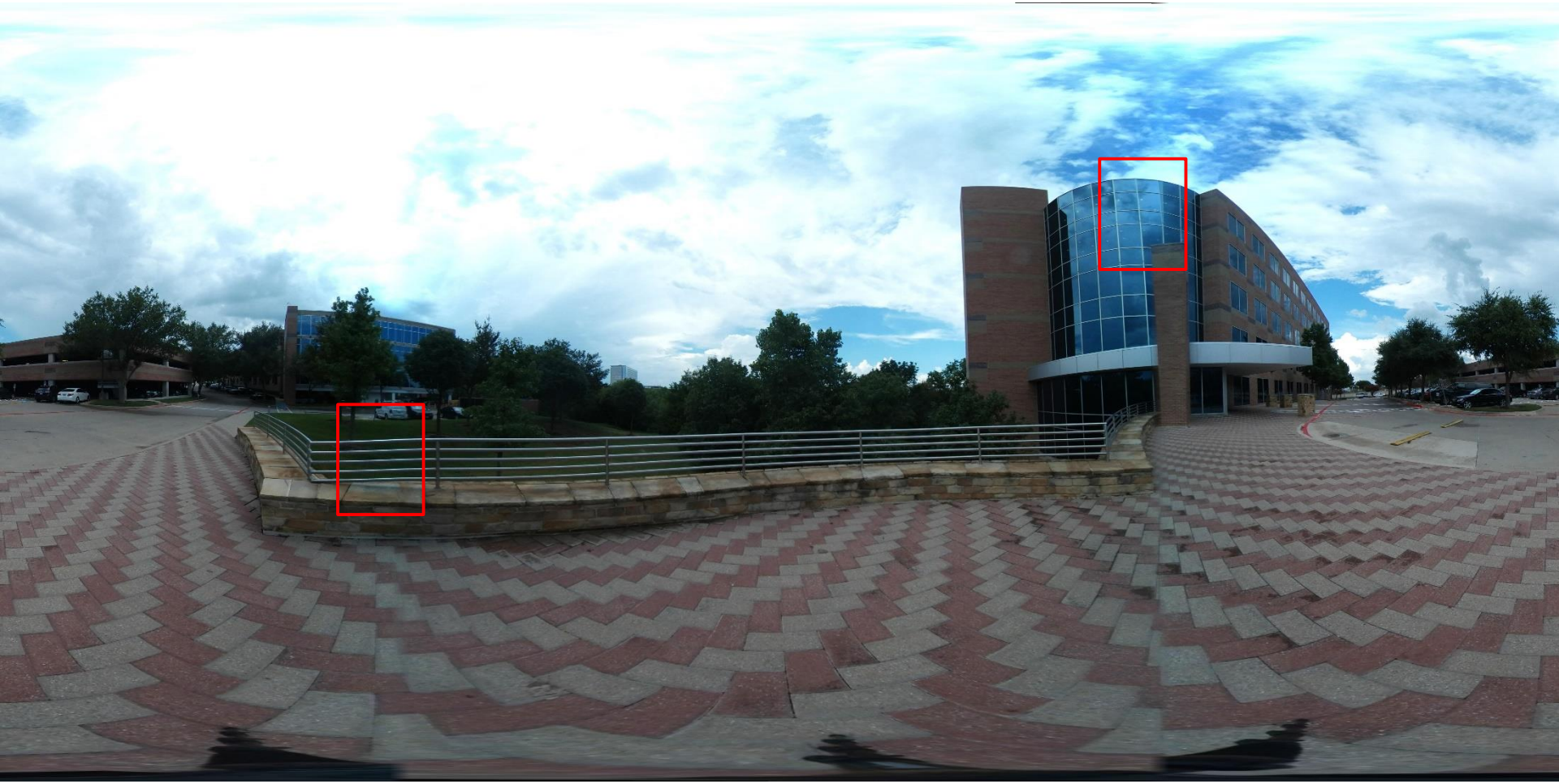
Stitched by the Proposed Method

Results (Phone vs. Proposed)



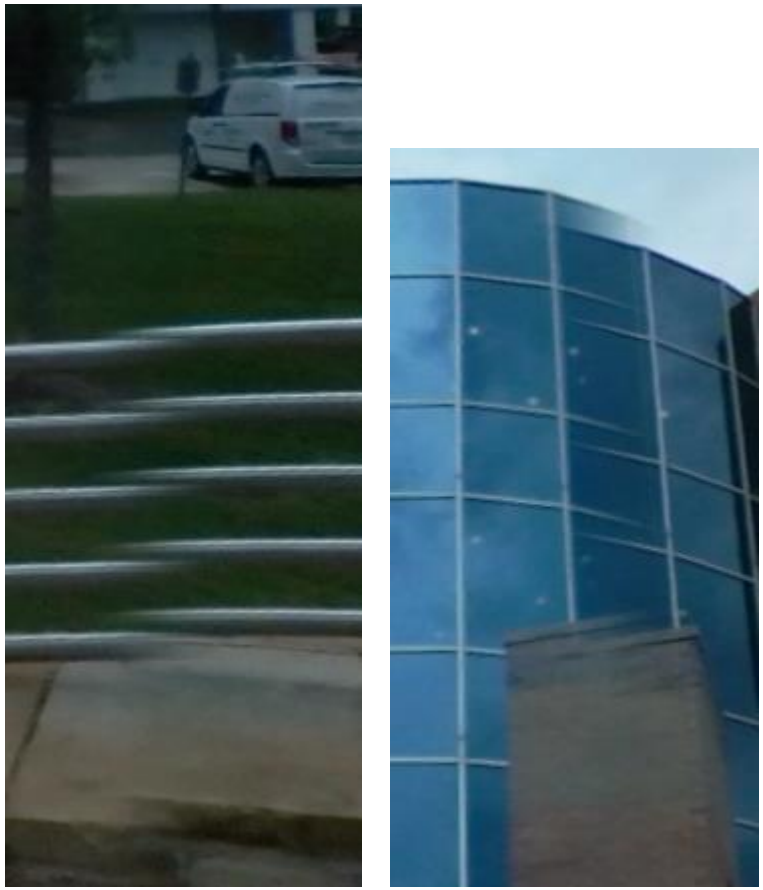
Stitched by Samsung Note-5

Results (Phone vs. Proposed)



Stitched by the Proposed Method

Results (Phone vs. Proposed): in 360-deg viewer



Stitched by Samsung Note-5

Stitched by the Proposed Method

Conclusion

- A new method for dual-fisheye (Gear360) stitching is presented. A system is built in OpenCV (C++) & Matlab.
- The proposed method is comprised of four steps:
  - Fisheye intensity compensation
  - Geometric derivation & fisheye unwarping
  - Two-step image registration that aligns one unwarped image to another
  - Blending using a ramp function
- Results show that the proposed method produces similar stitched image quality to the one by Samsung Note-5 for most of the cases and outperforms for some other cases (e.g. a person posing close to the lenses boundary).

Conclusion (cont.)

- One paper accepted for publication in ICASSP 2017.
- One provisional U.S. patent filled.
- One proposal to MPEG standardization for fisheye light fall-off at MPEG meeting Chengdu 2016*.

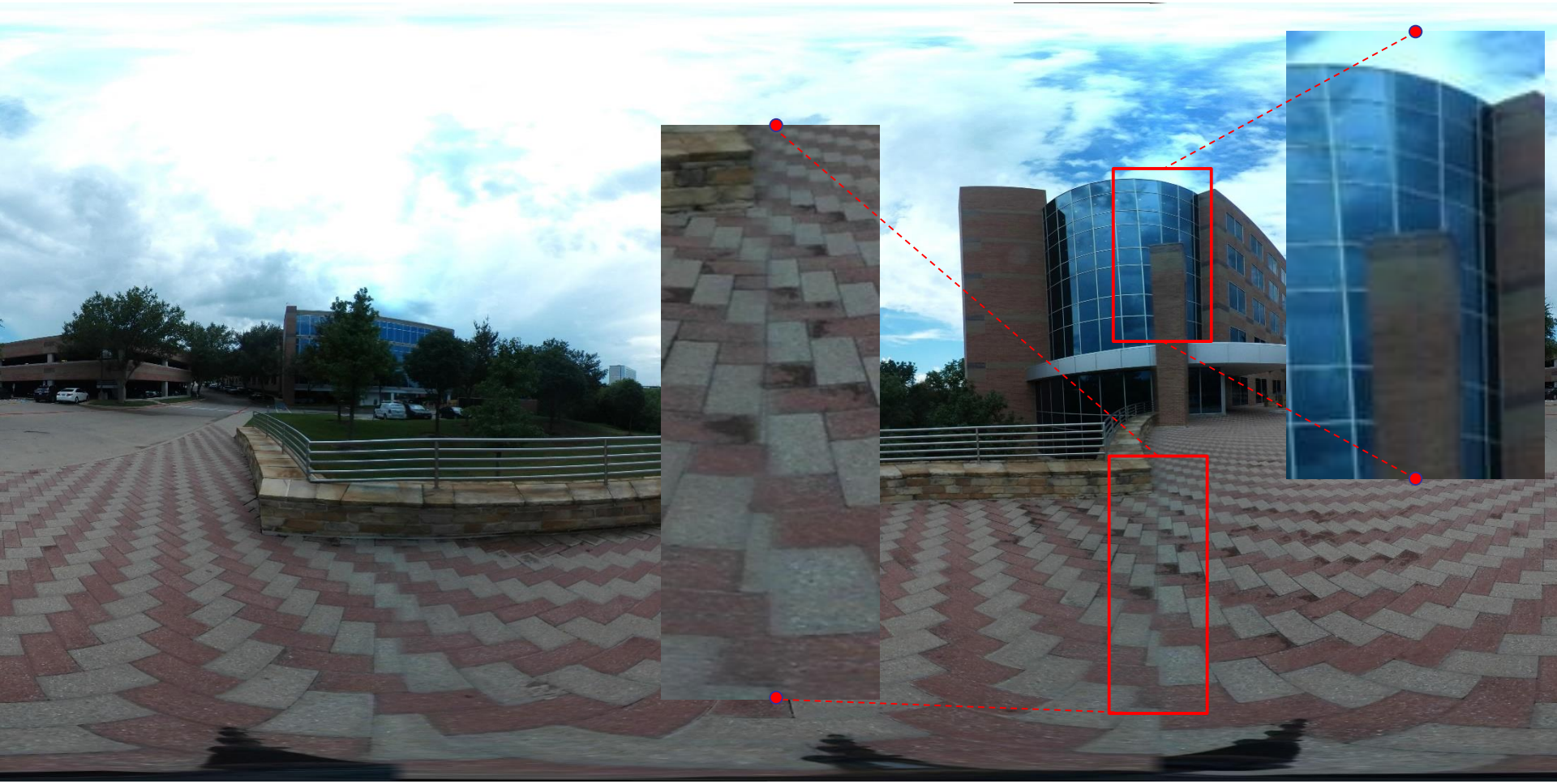
*Lens Shading Parameters Metadata for Omnidirectional Video, ISO/IEC JTC1/SC29/WG11 MPEG2016/m39469. Presented by Dr. Budagavi, Samsung Electronics.

Future Work

- Improve stitching quality for still image [under development].
- Develop 360-degree video stitching framework [under development].
- 360-degree video compression
- 360-degree video stabilization

Improve Stitching Quality

- Further align the patterned parts caused by parallax.

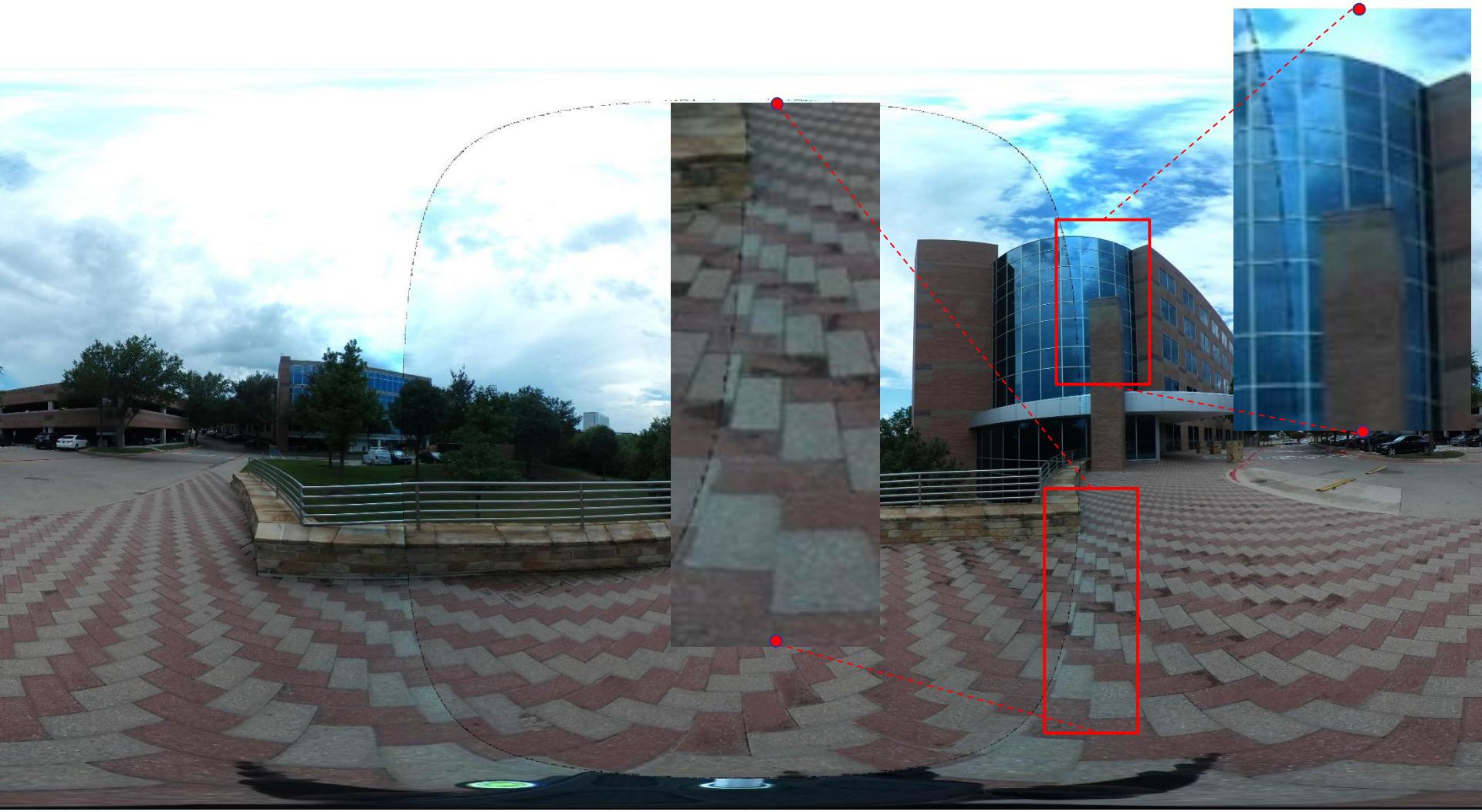


Improve Stitching Quality (cont.)

- The discontinuity cause: this approach solves an overdetermined system of control points for a warping matrix
  - → resulted in an approximated solution in a least-squares sense
  - → cannot transform all control points to desire positions.
- Solution: solve for an interpolation grids (of unwarped image size) by weighted least-squares to deform the image.
  - → points in the overlapping area are transformed to desired positions precisely.
  - → points outside the overlapping area are transformed to desired positions less accurately [minimal impact on stitching quality].

Improve Stitching Quality (cont.)

- Preliminary result (no blending yet, seam visible)



Improve Stitching Quality (cont.)

The current method

The method being developed (without blending)

The current method

The method being developed (without blending)

360-degree video for dual-fisheye lens camera

- Please see demo.

Thank You!

References

- [Map1] Map Projection: http://www.progonos.com/furuti/MapProj
- [Map2] Map Projection: https://en.wikipedia.org/wiki/Map_projection
- [J. P. Lewis95] “Fast Normalized Cross-Correlation”, Industrial Light & Magic, 1995