360-degree Video Stitching for Dual-fisheye Lens Cameras based on Rigid Moving Least Squares

Tuan Ho[*†], **Ioannis Schizas**[*], **K. R. Rao**[*], **Madhukar Budagavi**[†]

[*] Electrical Engineering Dept., University of Texas–Arlington, TX USA

[†] Samsung Research America, Richardson, TX USA

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** field of view (cannot produce 360x180-degree images).
  - Polydioptric: using multiple wide-view lenses in a same rig. Having seam artifact but getting much improved by hardware/algorithm.

Columbia Univ.'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
- Dual-fisheye camera:
  - Use two fisheye lenses
  - Compact
  - Affordable (less than $400/each)

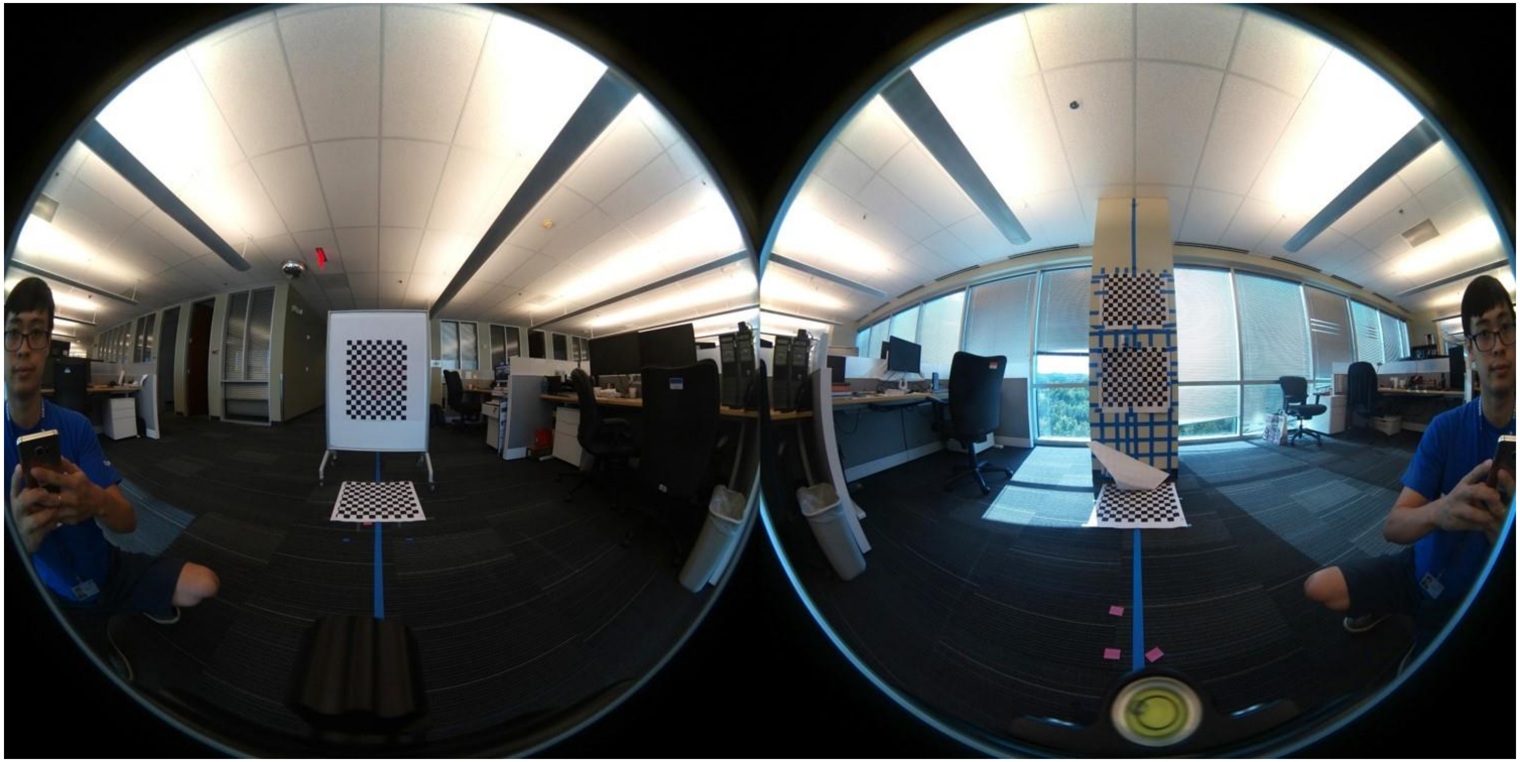
Dual-fisheye camera for 360-degree video:

- **Affordable optics and simpler hardware.**
- **Complexity shifted to stitching algorithm.**

Samsung Gear 360

Challenges

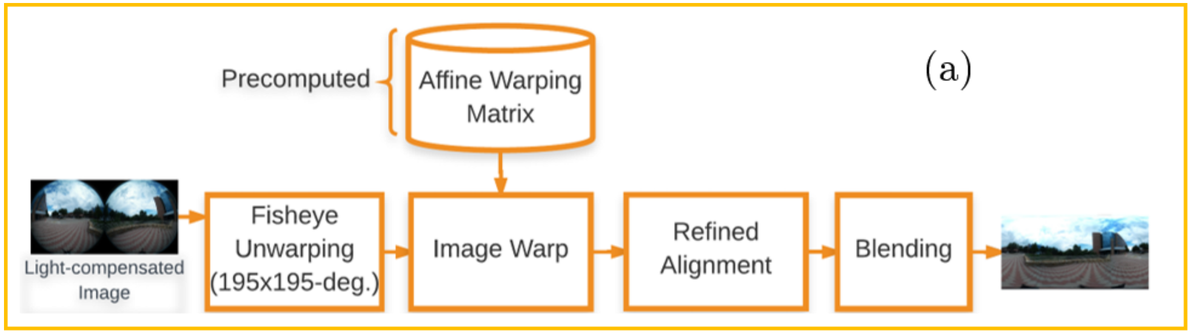
- Little overlap between two fisheye images taken by the dual-fisheye lens (each lens of the Gear 360 camera has 195-degree field of view).
- Parallax introduced by the two fisheye lenses.



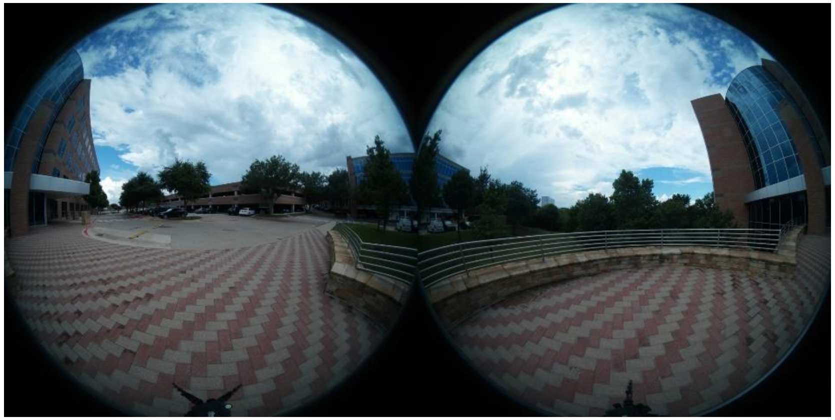
Example image taken by the dual-fisheye lens camera (Samsung Gear 360)

Related Work

- A system to produce stitched 360x180-degree images for dual-fisheye lens camera was proposed by T. Ho & M. Budagavi [7].



[7]



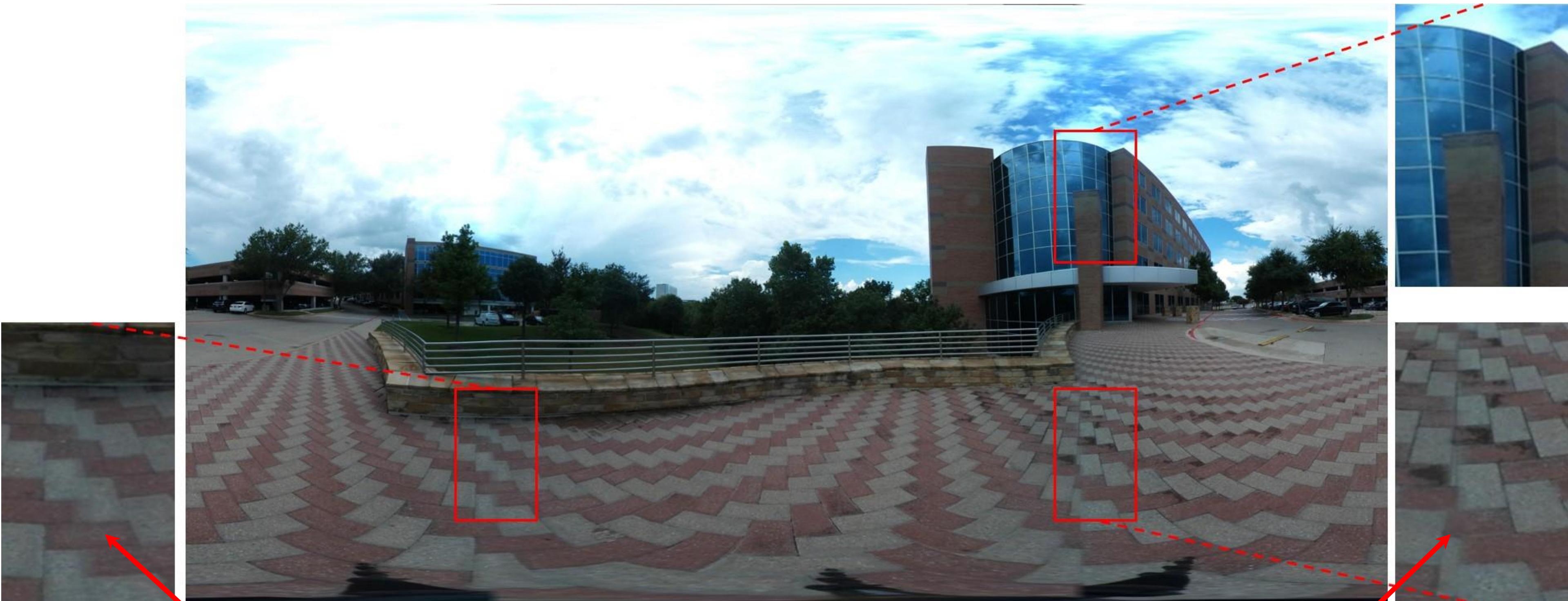
Images taken by dual-fisheye lens camera (Samsung Gear 360)

Stitcher

360x180-degree image stitched by [7]

[7] T. Ho and M. Budagavi, "Dual-fisheye lens stitching for 360-degree imaging," in Proc. of the 42nd IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'17), 2017.

Dual-fisheye Stitching Discontinuity

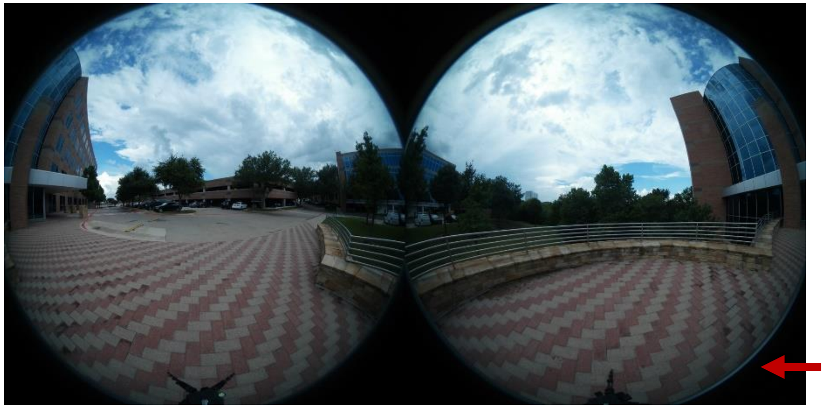


360x180-degree image stitched by [7]

Discontinuity in the stitching regions
due to parallax

Dual-fisheye Stitching Discontinuity Cause

- The affine warping in [7] aims to minimize the discontinuity globally.
- Parallax artifacts stacked up in the dual-fisheye lens system because:
  - The number of lenses is minimum (2 lenses)
  - The natural distortion of the wide field of view (195 degrees) fisheye lenses, esp. along the circumference of the circular fisheye images
- More parallax at the stitching boundaries.

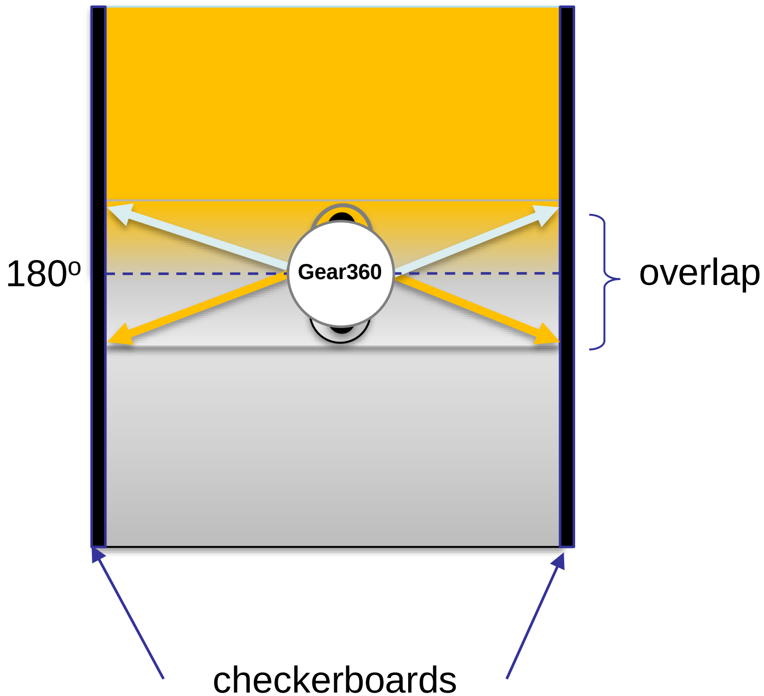


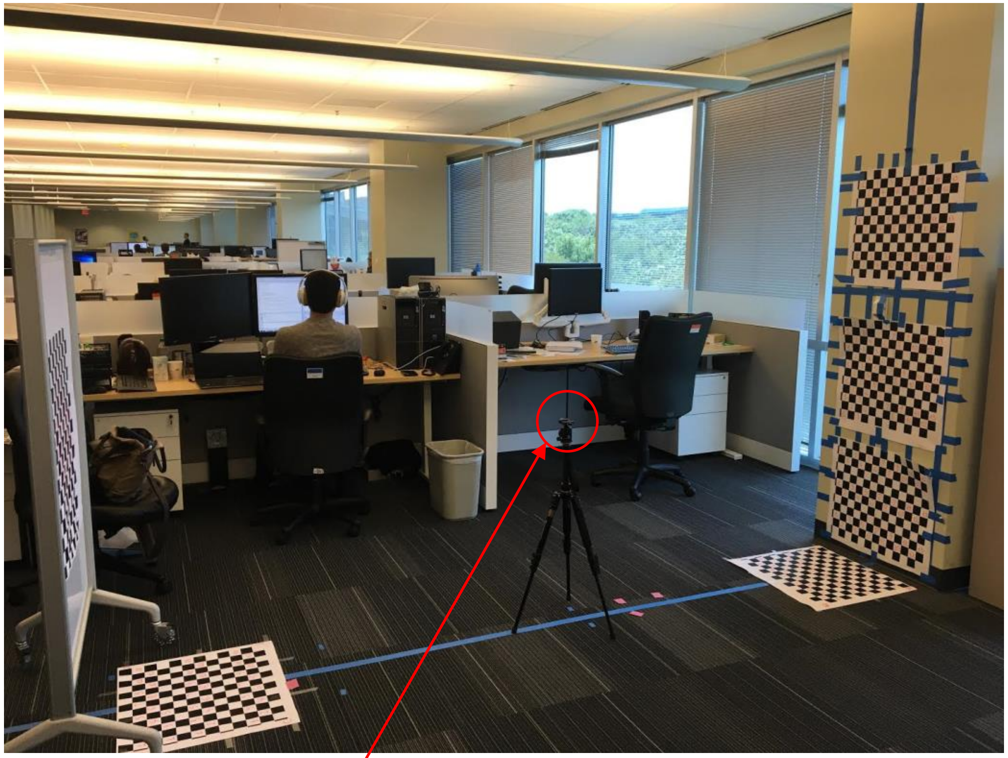
Fisheye 360-degree images. Severe distortion can be found along their circumferences.

Discontinuity in the stitching regions due to parallax [7]

Dual-fisheye Stitching Parallax Artifact

- Study on parallax artifact of dual-fisheye lenses (with Samsung Gear 360 cameras)

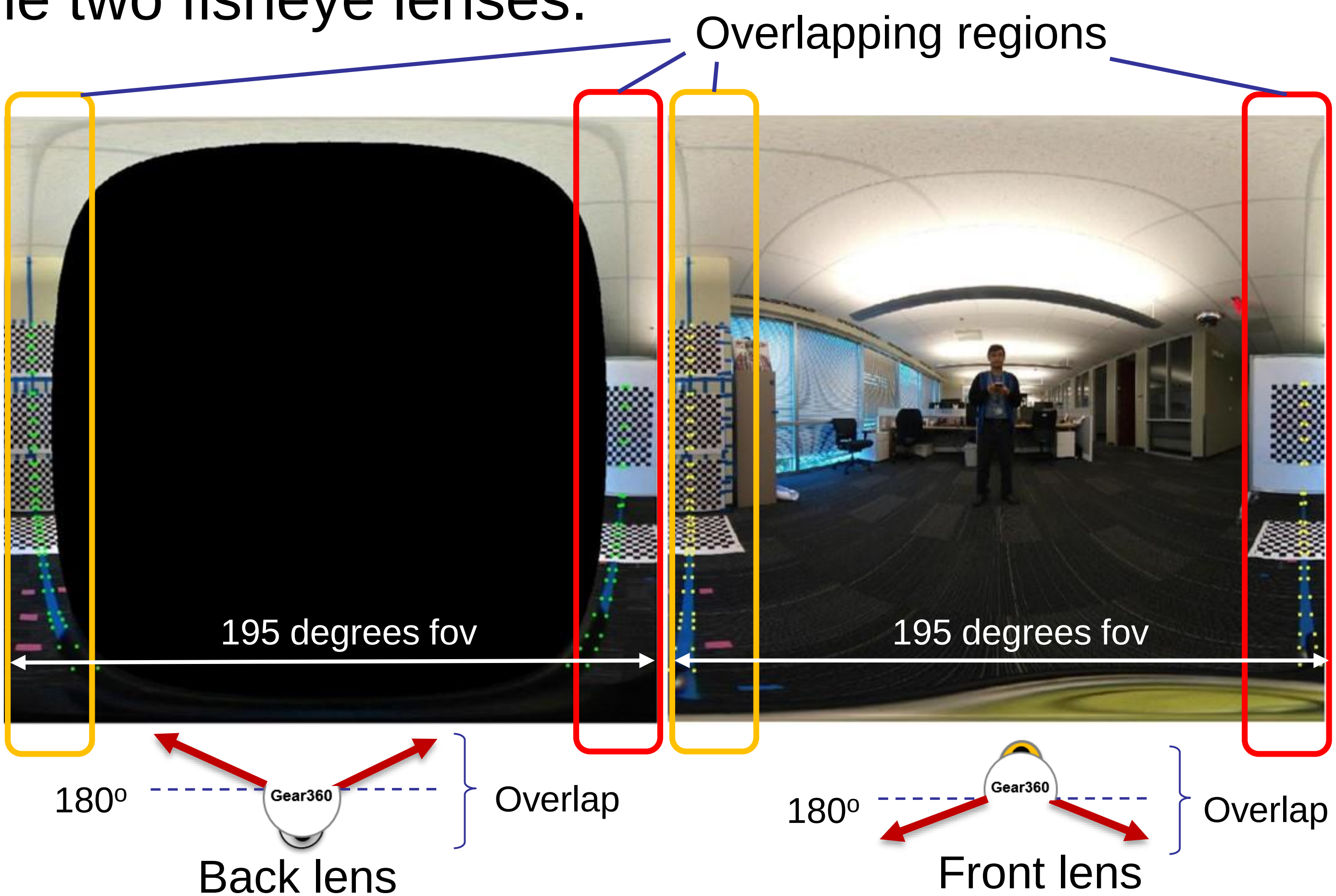




Each of all test  cameras is put here

Dual-fisheye Stitching Parallax Artifact

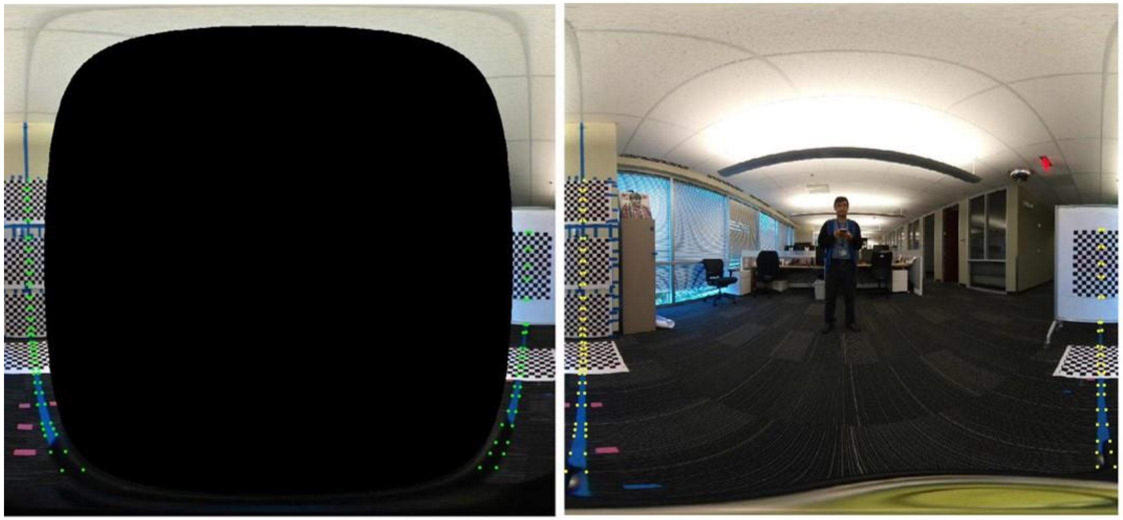
- Overlapped regions after unwarping the input fisheye images are shown below.
- The green and yellow points are **same** points viewed by the two fisheye lenses.



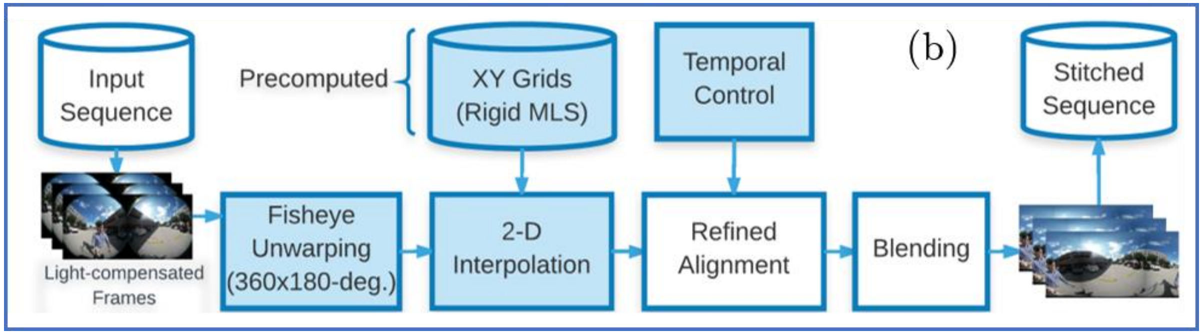
fov: field of view

Dual-fisheye Stitching Parallax Solution

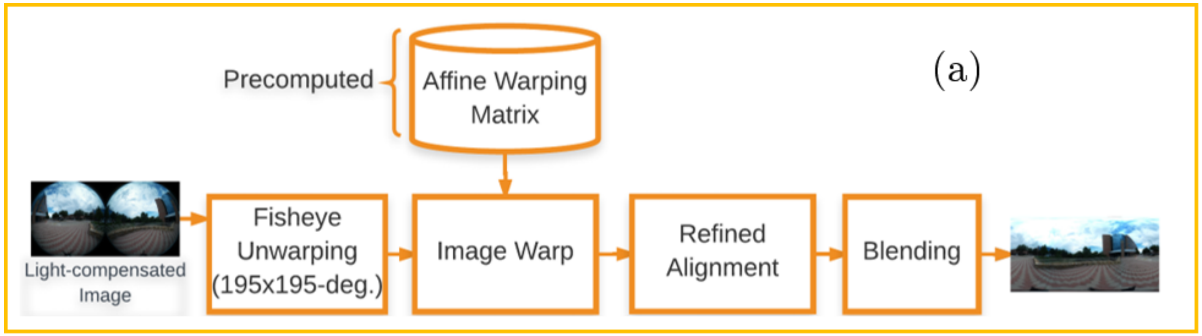
- To reduce the parallax impact, a local minimization approach is preferred.
- To this end, we introduce a rigid moving least squares-based approach for better stitching quality.
- **Goal**: deform the image on the right so that
  - In the overlapping regions: the yellow points become green points, thus aligning the two images
  - Else where: minimal changes.



The Proposed Stitching System



Our approach



[7]

What's new?

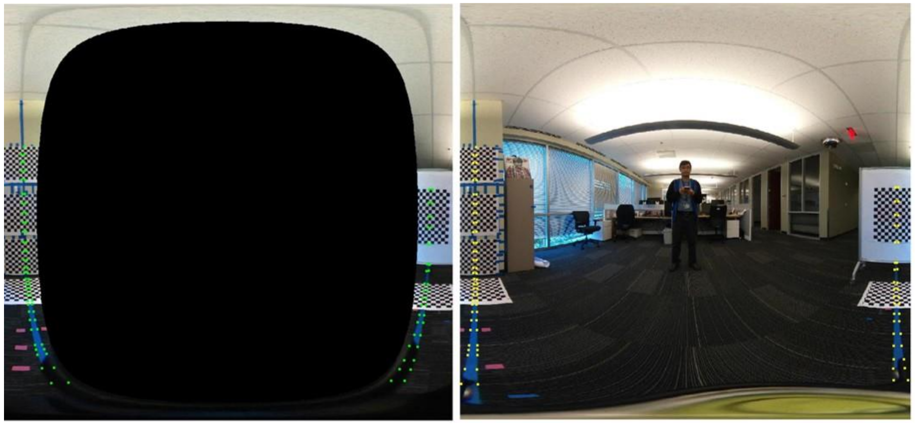
- Using rigid moving least squares instead of linear least square (affine matrix) to produce much better stitched images.
- Supporting 360-degree video.

Rigid Moving Least Squares

- For every point in the image, we solve for a transformation matrix M that minimizes the weighted least squares:

$$\underset{M}{\operatorname{argmin}} \sum_i w_i \left\| \hat{p}_i M - \hat{q}_i \right\|^2$$

$\mathbf{p}$ (green points) and $\mathbf{q}$ (yellow points): the control points in the overlapping regions of the original and deformed images respectively. $\hat{p}_i = v - p^*$ and $\hat{q}_i = q^*$ are derived from each point $\mathbf{v}$ in the image on the right and the weighted centroids of p^* and q^*.



Rigid Moving Least Squares (MLS)

- The weights w_i are calculated as follows:

$$w_i = \frac{1}{|p_i - v|^{2\alpha}}$$

The weights w_i :

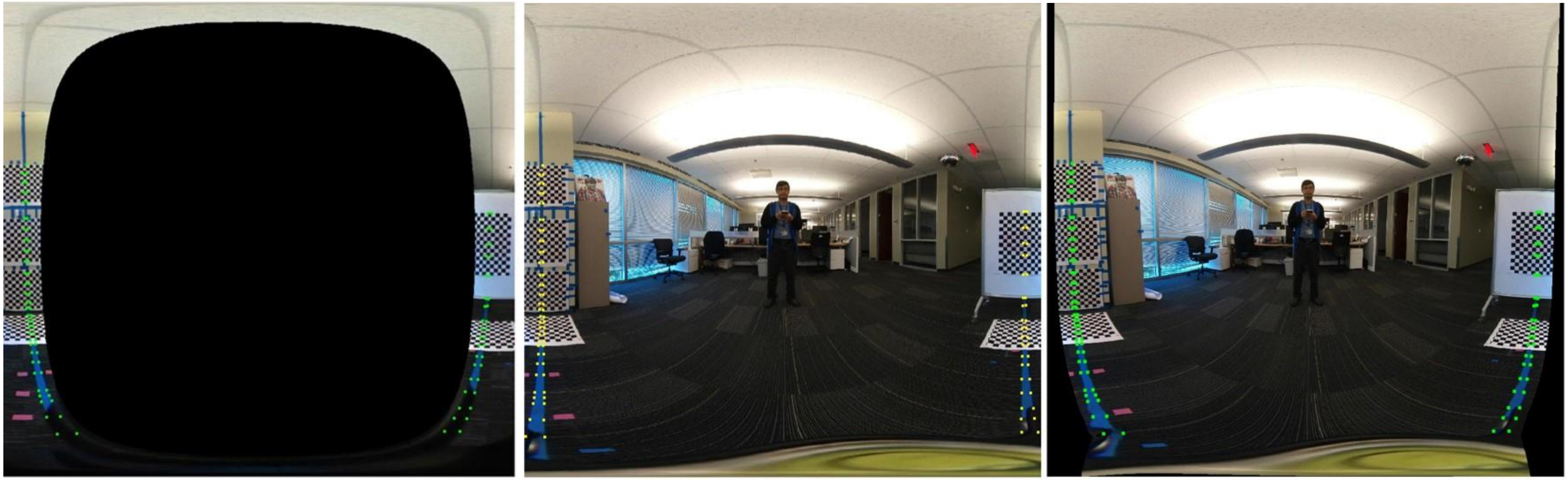
- proportional to the distance between the image point v and the control point p_i .
- gets smaller when v moves further away from p_i (i.e. the least squares minimization depends on the point of evaluation, thus the name moving least squares). When $v \neq p_i$, f interpolates $f(v) = q_i$.
- in other words, high-valued weights inside the overlapping regions and small-valued weights elsewhere.

- The rigid transformation is preferred to the affine and similarity transformation $[10]$.

[10] S. Schaefer, T. McPhail, and J. Warren, "Image deformation using moving least squares," in Proc. of ACM SIGGRAPH '06, 2006.

Rigid Moving Least Squares (MLS)

- Since the two lenses are fixed into the camera system, the rigid MLS-based optimization can be done offline and produce the interpolation grids.
- The grids are then used to deform the image by 2-D interpolation.



(a) **(b)** **(c)** Deformed version of image (b) by rigid MLS, to align it with (a)

Unwarped images

Refined Alignment

- A linear least squares is applied on top of the MLS deformation to further align the images, i.e. normalized cross-correlation-based matching [7].
- This refined alignment works adaptively to the scene, i.e. to align objects with arbitrary depth to the cameras.

(a) (b)

The person sitting very close to the <u>boundary</u> of the dual-fisheye lens camera.
(a) With MLS deformation. (b) With MLS deformation + refined alignment.

Still-image Stitching Results

(a) Image stitched by the proposed approach

(b) Image stitched by [7]

(a1)

(b1)

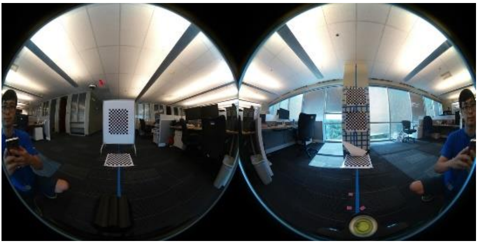
(a2)

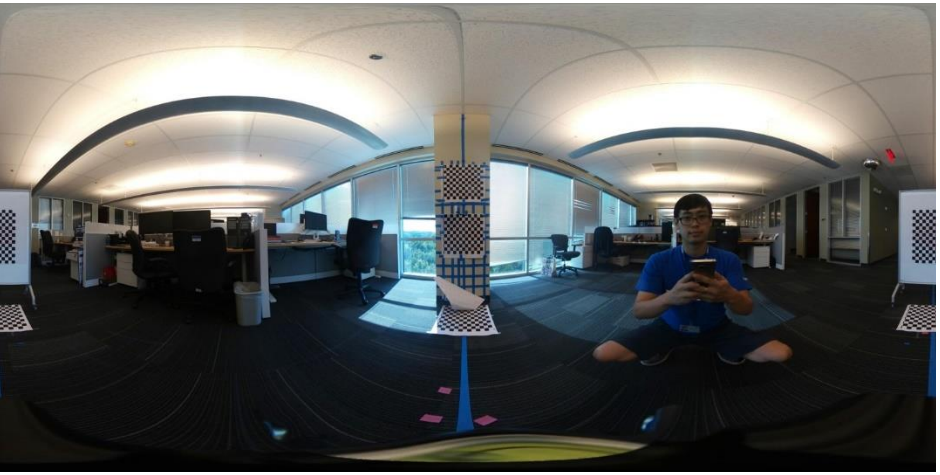
(b2)

(a3)

(b3)

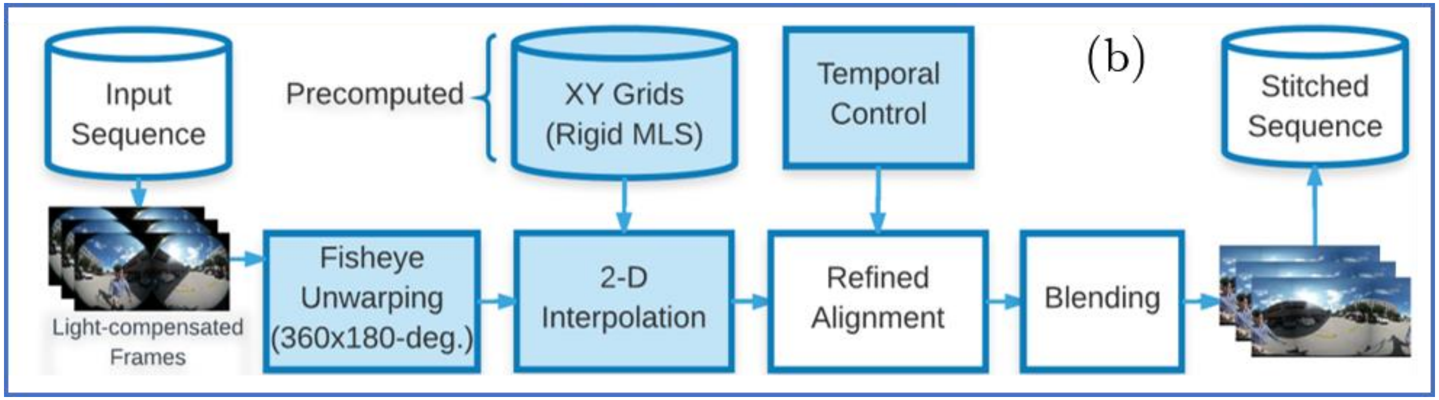
Still-image Stitching Results (This Work)





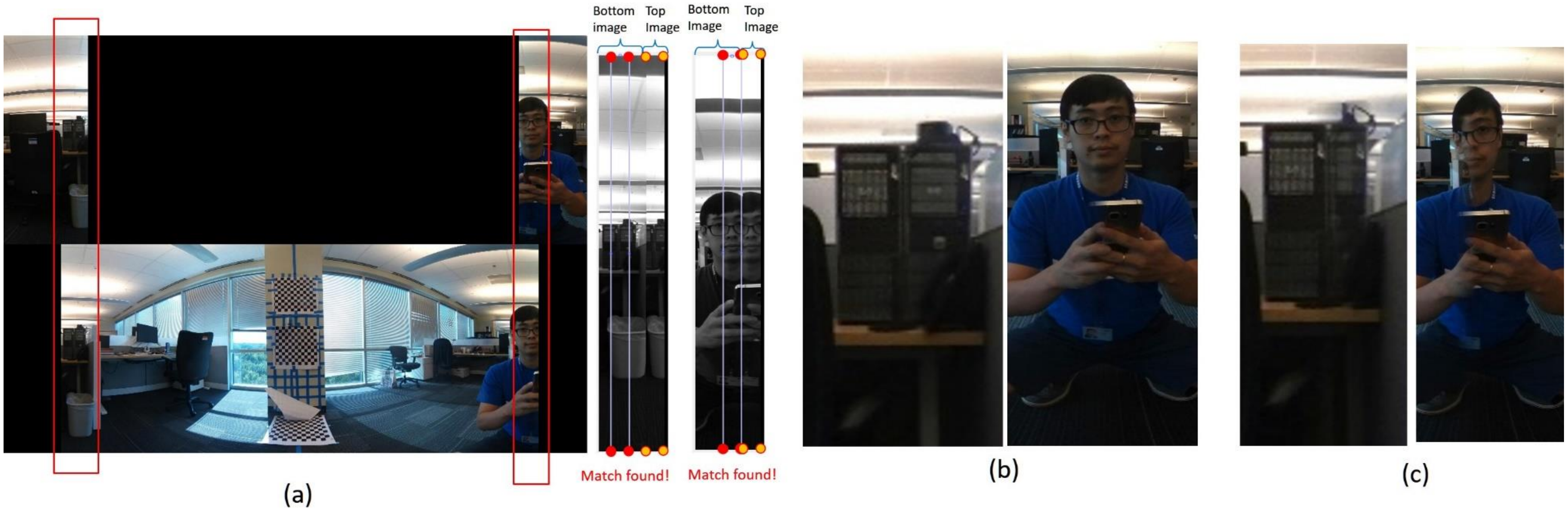
360-degree Video



- Goal: maintain smooth transition between stitched frames in the 360-degree video.
- Visual glitch is not comfortable for viewer.
- The stitching glitch caused by the abrupt change in the warping matrices among adjacent frames.
- A temporal control is added to adjust the template matching in the refined alignment process to minimize glitches.

Temporal Control for Refined Alignment

- The refined alignment [7]
  - is a patch-based matching based on cross-correlation
  - works adaptively to the scene
  - may produce different warping matrices for each frames
- Could cause glitches if the warping matrices largely differentiate among adjacent stitched frames.



(a) Matching process in the refined alignment [7]. Scores are assigned for each stitching boundaries. (b) With the refined aligment. (c) Without the refined alignment.

Temporal Control for Refined Alignment

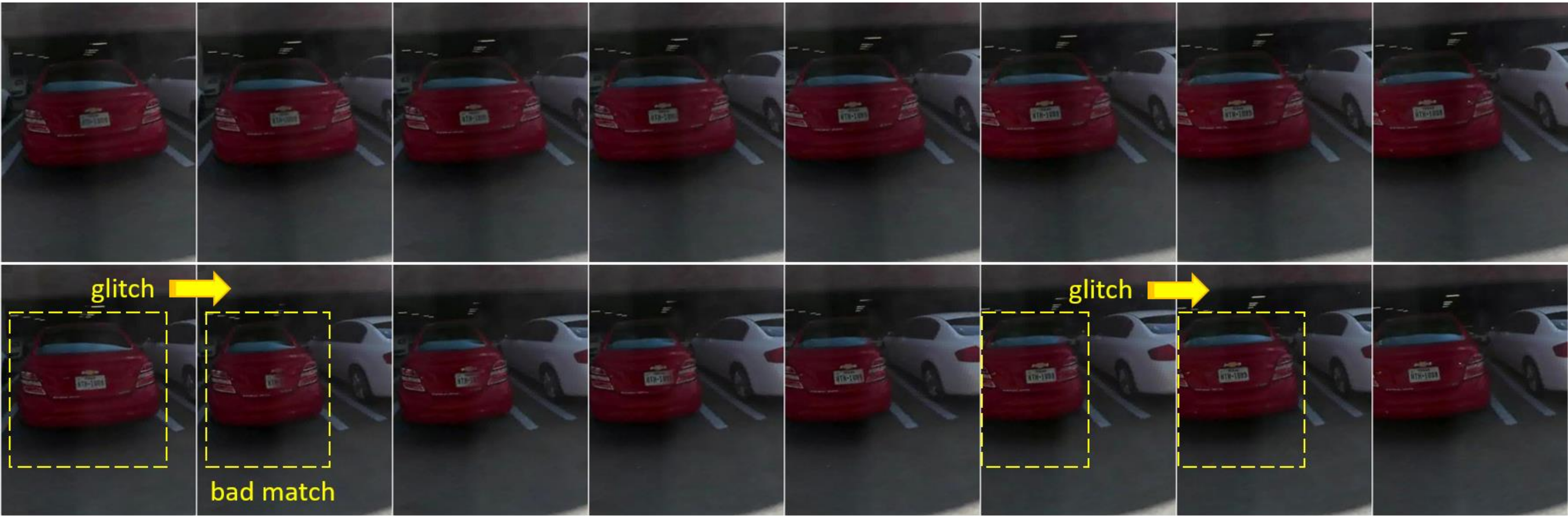
- The matching score threshold is chosen empirically.
- E.g. if both matching scores larger than 0.85, use the estimated warping matrix. (score=1 means the template is same to the reference)
- Otherwise, use the warping matrix of the previous frame
- Or not applying the refined alignment.

Algorithm 1: Refined Alignment (with jitter control)

```
Input: leftImage, rightImage (deformed)
(scoreLeft, scoreRight) ← TemplMatch();
if ( both matching scores are good ) then
    Estimate affine warping matrix affineMat;
    Store affineMat for the next frame;
    warpEn ← 1;
else // bad scores on either boundary
    if matching scores of the previous frame are good
      then
        warpEn ← 1;
        affineMat ← previous affineMat;
    else
        warpEn ← 0; // don't warp image
    end
end
if ( warpEn ) then
    Warp rightImage by affineMat;
end
```

Temporal Control

- The proposed temporal control minimizes the glitches caused by different warping matrices among adjacent frames.



Objects in the stitching boundary.
Top row: with temporal control (no glitch).
Bottom row: without temporal control (visible glitches).

Conclusion

- A system for stitching 360-degree video frames generated by the dual-fisheye lens cameras is presented.
- The proposed system can produce better stitching quality frames and videos with indiscernible glitch, comparing to the state-of-the-art work.
- The proposed stitcher is written in C++ with OpenCV.
- It takes around 500ms to stitch one 3840x1920 360-degree frame on a laptop with Intel core i7 1.8 GHz, 8GB RAM, 256GB SSD. (excluding the one-time configuration setup, such as reading MLS deform 2-D interpolation grids, etc., in prior to the stitching process)

Demo

- Demo time: 360-degree video stitched by the proposed method and the previous work [7].
- View in normal mode: VideoLAN VLC is used to display the videos.
- View in VR mode: the GoPro VR Player (free) is used to play the 360-degree videos.
- The demo can be found in the supplement materials at:

https://sigport.org/documents/360-degree-video-stitching-dual-fisheye-lens-cameras-based-rigid-moving-least-squares

References

[1] “GoPro Odyssey,” https://gopro.com/odyssey, [Retrieved August, 2016].

[2] “Facebook Surround360,” https://facebook360.fb.com/facebook-surround-360, [Retrieved August,2016].

[3] “Google Cardboard,” https://vr.google.com/cardboard, [Retrieved August, 2016].

[4] “Samsung GearVR,” www.samsung.com/us/gearvr, [Retrieved August, 2016].

[5] “Ricoh Theta,” https://theta360.com, [RetrievedAugust, 2016].

[6] “LG 360 Cam,” http://www.lg.com/us/mobile-accessories/lg-LGR105.AVRZTS-360-cam, [Retrieved August, 2016].

[7] T. Ho and M. Budagavi, “Dual-fisheye lens stitching for 360-degree imaging,” in Proc. of the 42nd IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP’17), 2017.

[8] M. Brown and D.G. Lowe, “Automatic panoramic image stitching using invariant features,” International Journal of Computer Vision, vol. 74, pp. 59–73, August 2007.

[9] R. Szeliski, Computer Vision: Algorithms and Applications, Springer, London, UK, 1st edition, 2011.

[10] S. Schaefer, T. McPhail, and J. Warren, “Image deformation using moving least squares,” in Proc. of ACM SIGGRAPH ’06, 2006.

Thank You!