# CS543 Assignment 2

**Your Name:** Kaylin Chen
**Your NetId:** tc49

# Part 1 Fourier-based Alignment:

You will provide the following for each of the six low-resolution and three high-resolution images:

- Final aligned output image
- Displacements for color channels
- Inverse Fourier transform output visualization for ***both*** channel alignments ***without*** preprocessing
- Inverse Fourier transform output visualization for ***both*** channel alignments ***with*** any sharpening or filter-based preprocessing you applied to color channels

You will provide the following as further discussion overall:

- Discussion of any preprocessing you used on the color channels to improve alignment and how it changed the outputs
- Measurement of Fourier-based alignment runtime for high-resolution images (you can use the python time module again). How does the runtime of the Fourier-based alignment compare to the basic and multiscale alignment you used in Assignment 1?

## A: Channel Offsets

Replace <C1>, <C2>, <C3> appropriately with B, G, R depending on which you use as the base channel. Provide offsets in the **original image coordinates** (after the image has been divided into three equal parts corresponding to each channel) and be sure to account for any cropping or resizing you performed.

Low-resolution images (using channel B as base channel):

| Image | G (h,w) offset | R (h,w) offset |
|---|---|---|
| 00125v.jpg | (5,2) | (10,1) |

| | | |
|---|---|---|
| 00149v.jpg | (4,2) | (9,2) |
| 00153v.jpg | (7,3) | (14,5) |
| 00351v.jpg | (4,1) | (13,1) |
| 00398v.jpg | (5,3) | (11,4) |
| 01112v.jpg | (0,0) | (5,1) |

High-resolution images (using channel R as base channel):

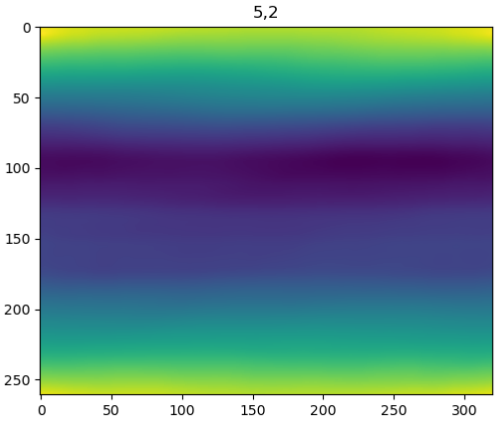
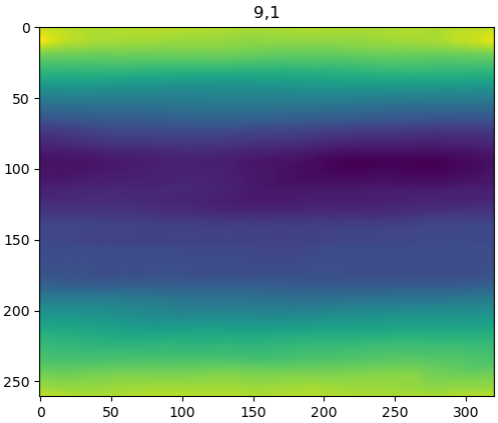
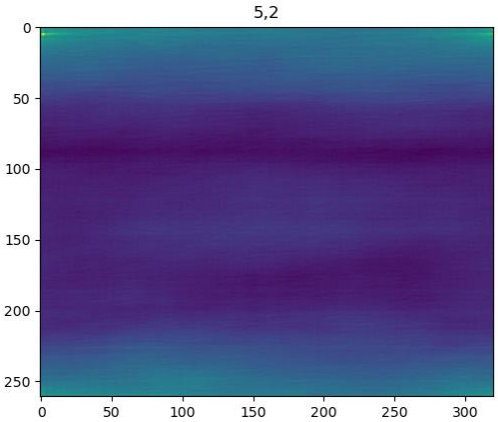
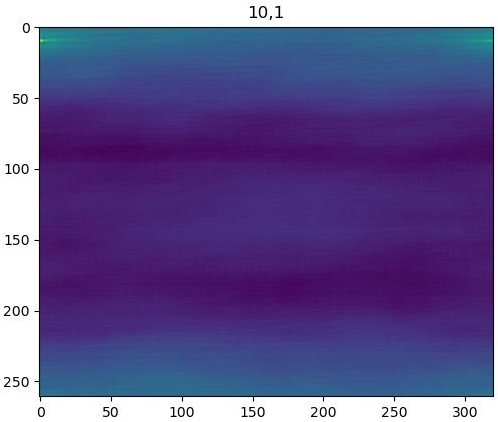
| Image | G (h,w) offset | B (h,w) offset |
|---|---|---|
| 01047u.tif | (25, 20) | (71,33) |
| 01657u.tif | (51, 9) | (112,12) |
| 01861a.tif | (70, 37) | (146,62) |

## B: Output Visualizations

For each image, insert 5 outputs total (aligned image + 4 inverse Fourier transform visualizations) as described above. When you insert these outputs be sure to clearly label the inverse Fourier transform visualizations (e.g. “G to B alignment without preprocessing”).

### 00125v.jpg

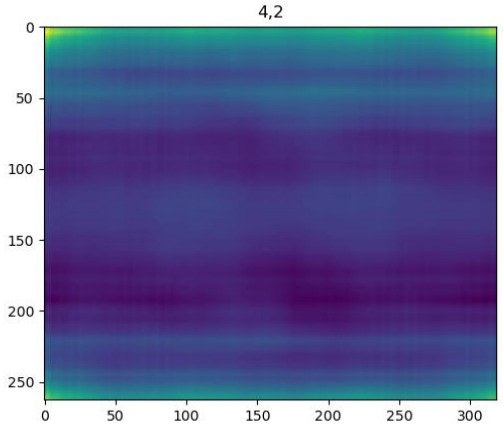
| Aligned image |
|---|

| G to B alignment **without** preprocessing | R to B alignment **without** preprocessing |
|---|---|
|  |  |
| G to B alignment **with** preprocessing | R to B alignment **with** preprocessing |
|  |  |

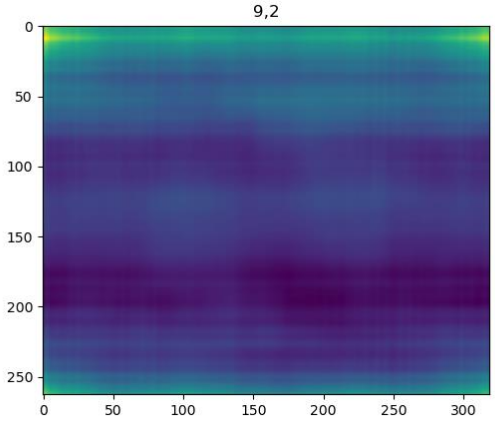
00149v.jpg

Aligned image

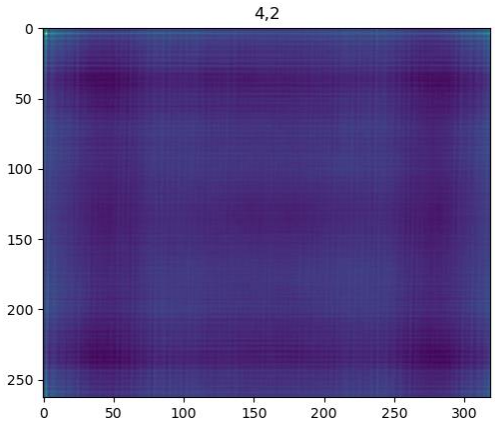
G to B alignment **without** preprocessing



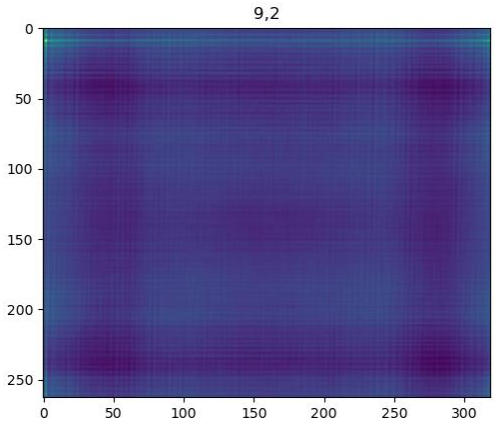

R to B alignment **without** preprocessing




G to B alignment **with** preprocessing



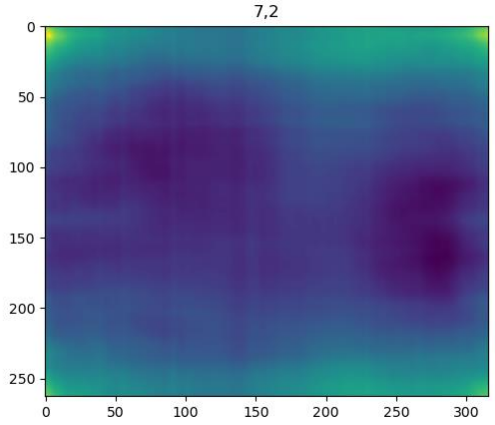

R to B alignment **with** preprocessing



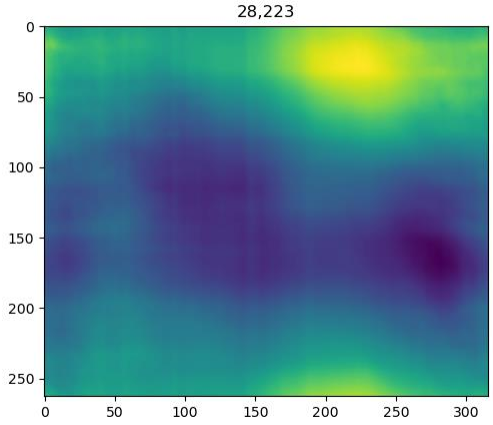
00153v.jpg

Aligned image

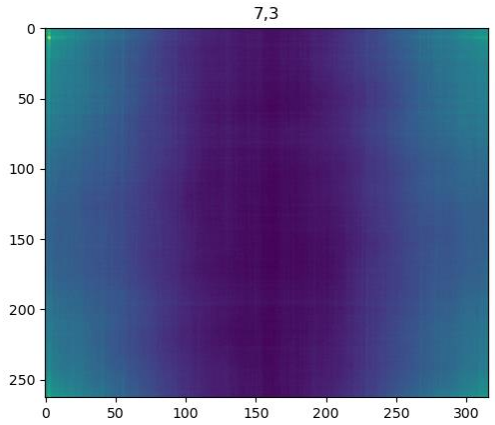
G to B alignment **without** preprocessing



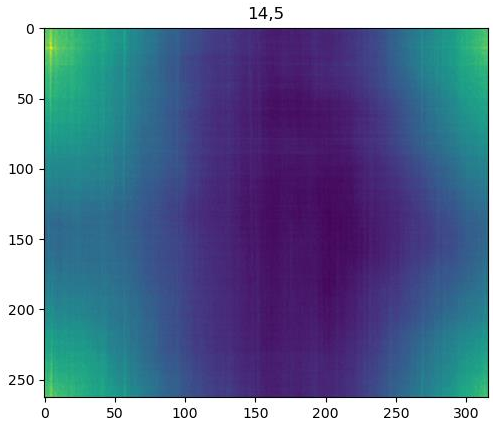

R to B alignment **without** preprocessing




G to B alignment **with** preprocessing



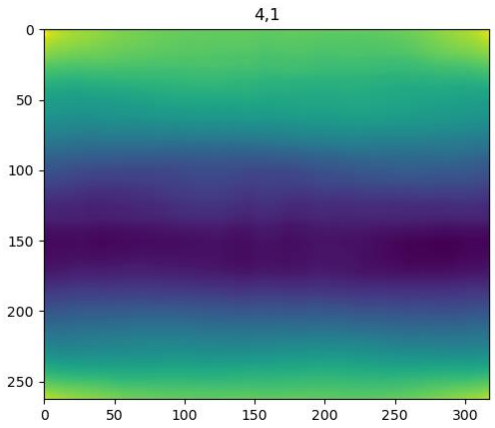

R to B alignment **with** preprocessing



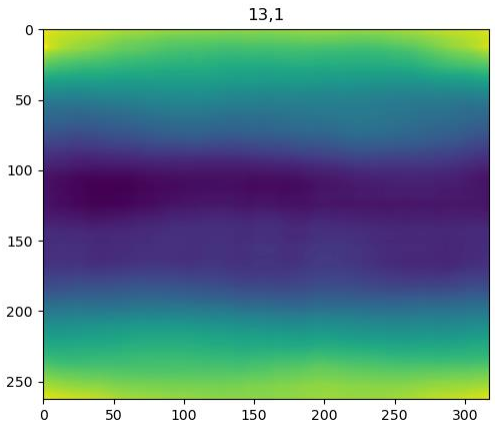
00351v.jpg

Aligned image

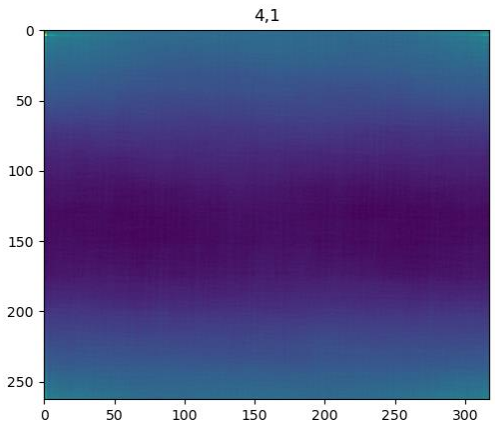
G to B alignment **without** preprocessing



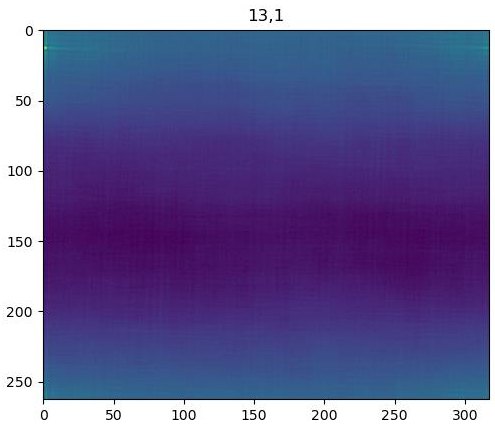

R to B alignment **without** preprocessing




G to B alignment **with** preprocessing



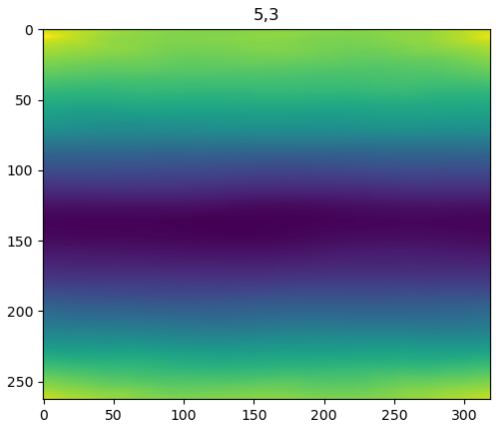

R to B alignment **with** preprocessing



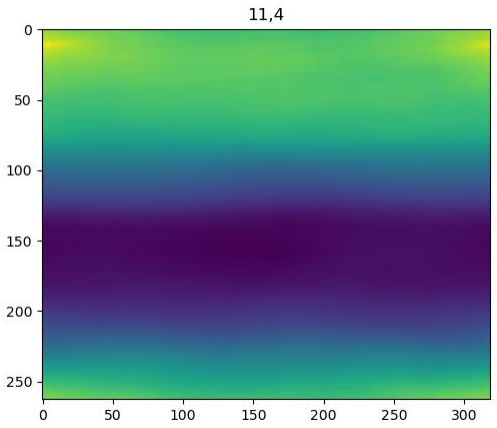
00398v.jpg

Aligned image

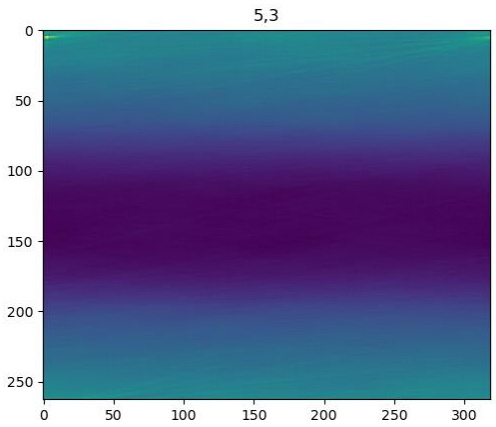
G to B alignment **without** preprocessing



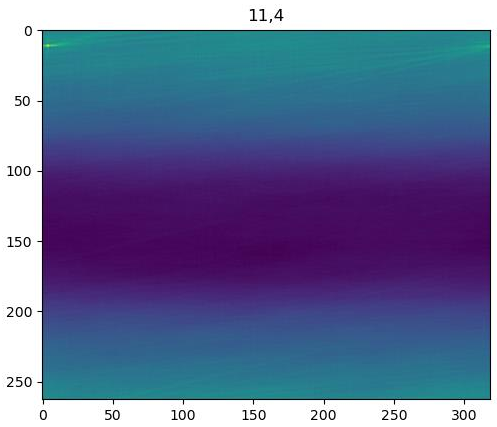

R to B alignment **without** preprocessing




G to B alignment **with** preprocessing



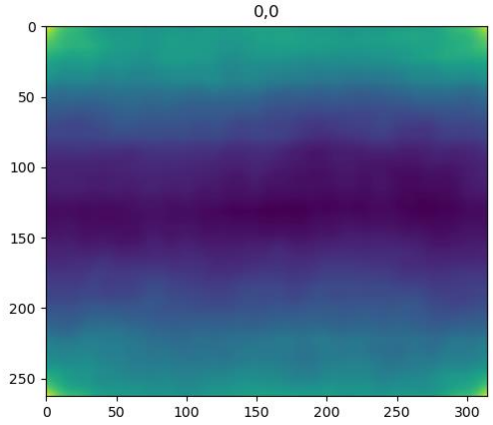

R to B alignment **with** preprocessing



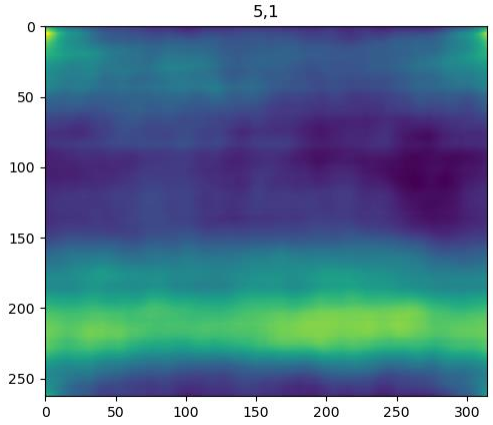
01112v.jpg

Aligned image

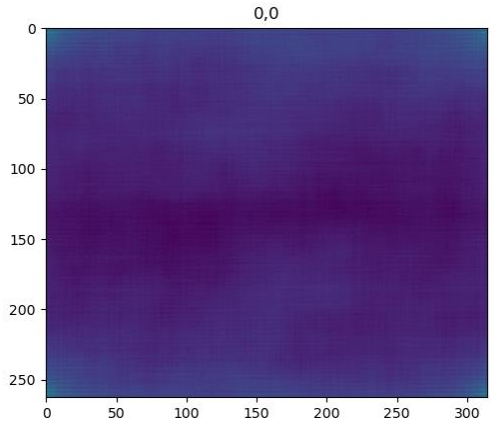
G to B alignment **without** preprocessing



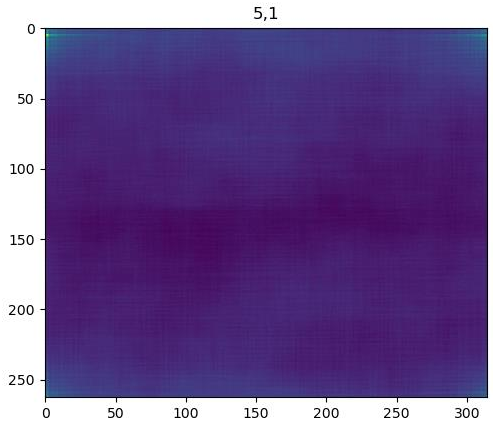

R to B alignment **without** preprocessing




G to B alignment **with** preprocessing

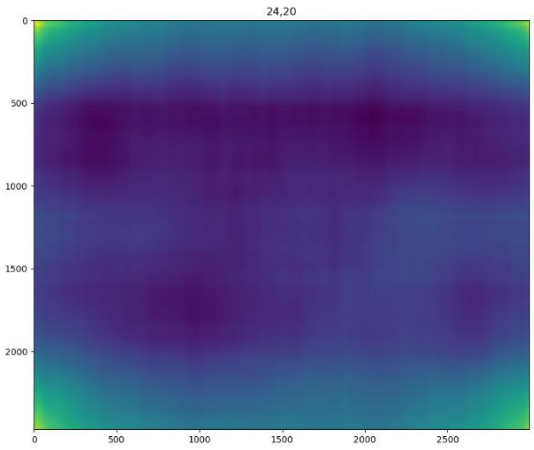



R to B alignment **with** preprocessing

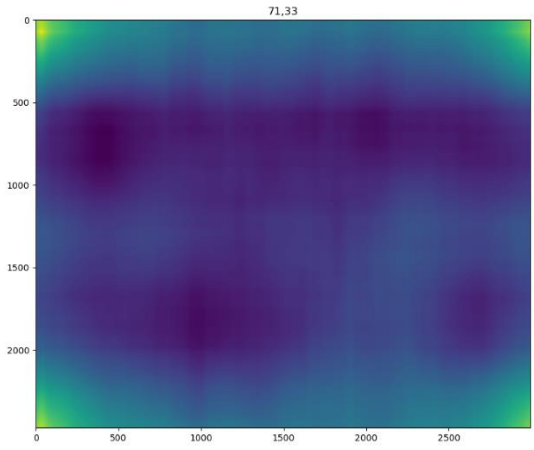


01047u.tif

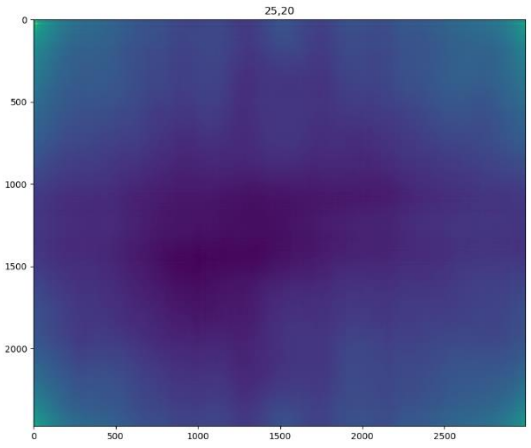
| Aligned image | |
|---|---|
|  | |
| G to R alignment **without** preprocessing | B to R alignment **without** preprocessing |

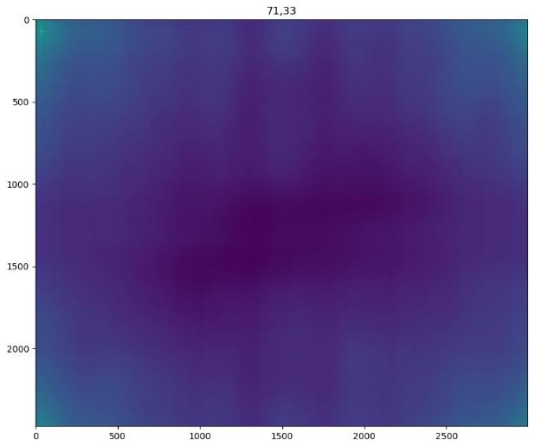



G to R alignment **with** preprocessing



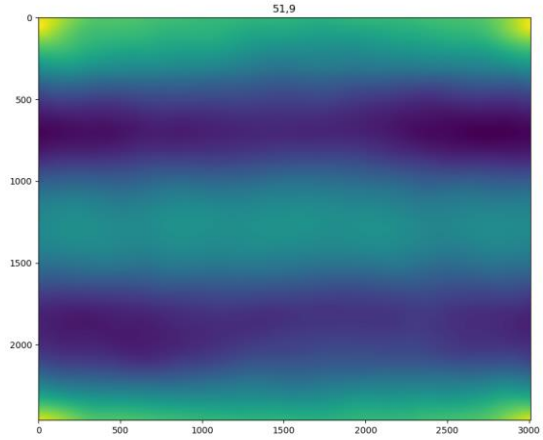
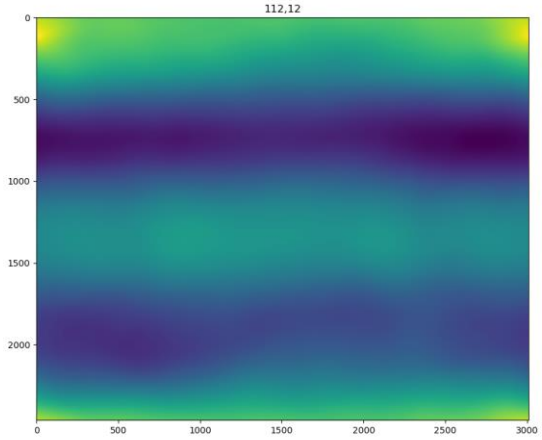

B to R alignment **with** preprocessing

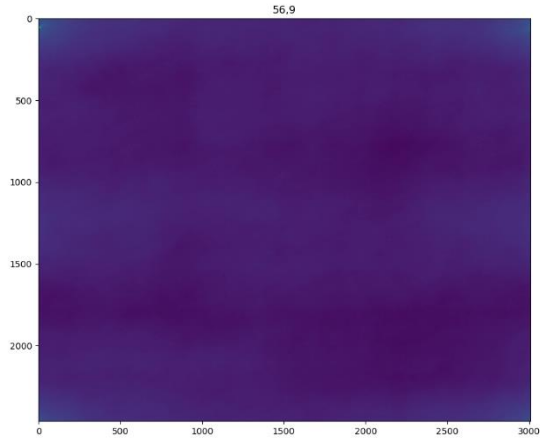
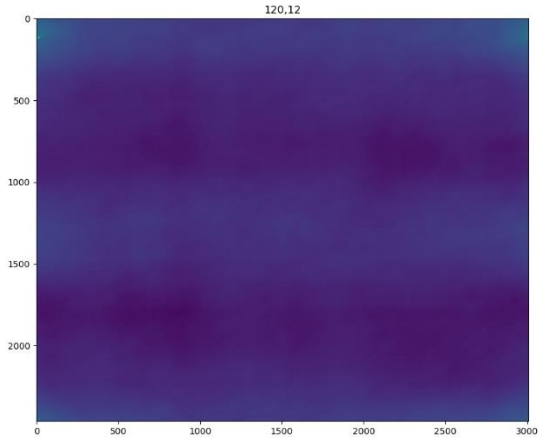



01657u.tif

Aligned image

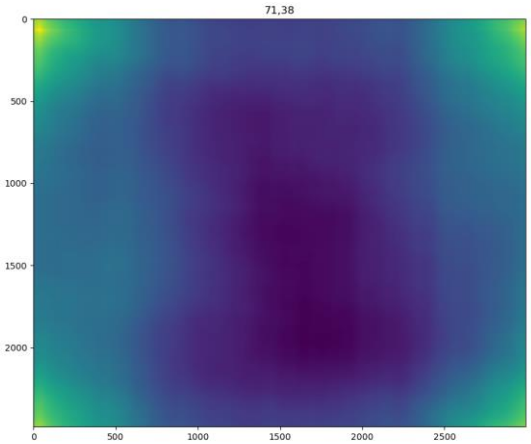
| G to R alignment **without** preprocessing | B to R alignment **without** preprocessing |
| --- | --- |
|  |  |

| G to B alignment **with** preprocessing | B to R alignment **with** preprocessing |
| --- | --- |
|   |   |

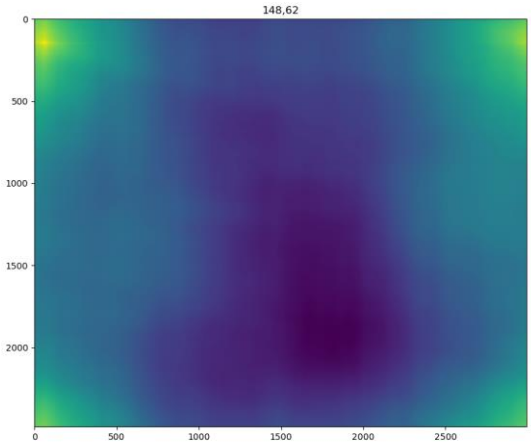
## 01861a.tif

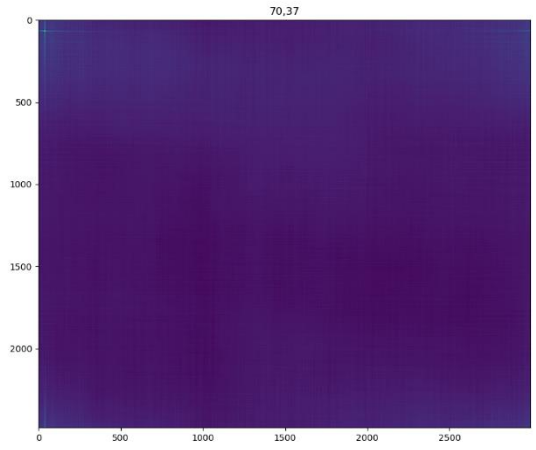
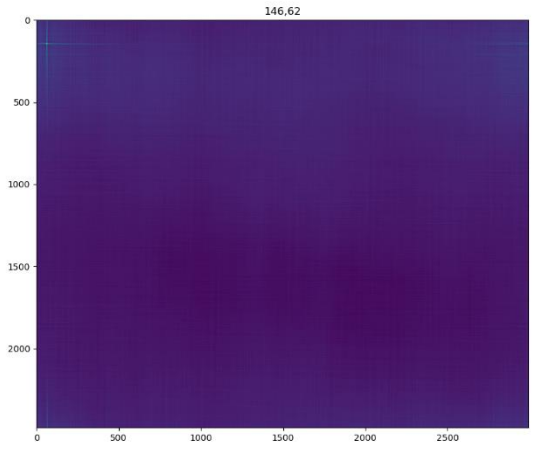
| Aligned image |
| --- |

G to R alignment **without** preprocessing




B to R alignment **without** preprocessing



| G to R alignment **with** preprocessing | B to R alignment **with** preprocessing |
| --- | --- |
|   |   |

## C: Discussion and Runtime Comparison

For the preprocessing, I try on both sharpen and Laplacian Gaussian and find out the Laplacian provide a good results for 00153v. Before applying the preprocessing, it can only align two channel, but with the filter it align perfectly.

By Fourier-based, the runtime for high-resolution images is around 3 seconds, slightly slower than the multi-scale approach in the previous assignment, which takes about 1 second; and it is much faster than single layer approach in MP1, which I believe will take 5 hours approximately.

# Part 2 Scale-Space Blob Detection:

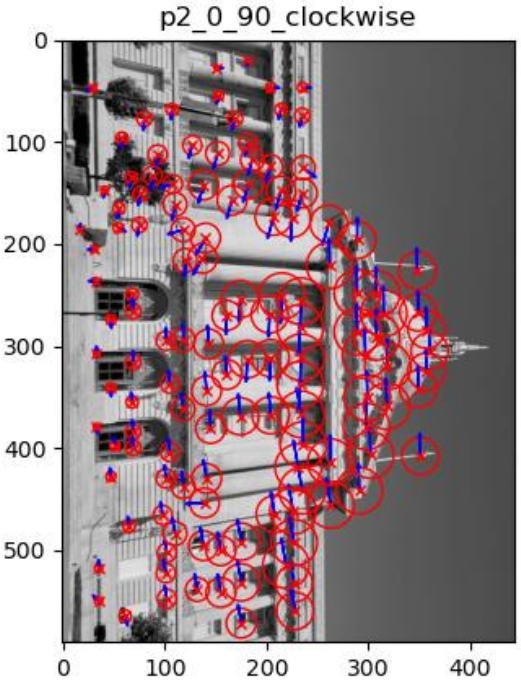
You will provide the results for ***4 different examples chosen by your own***:

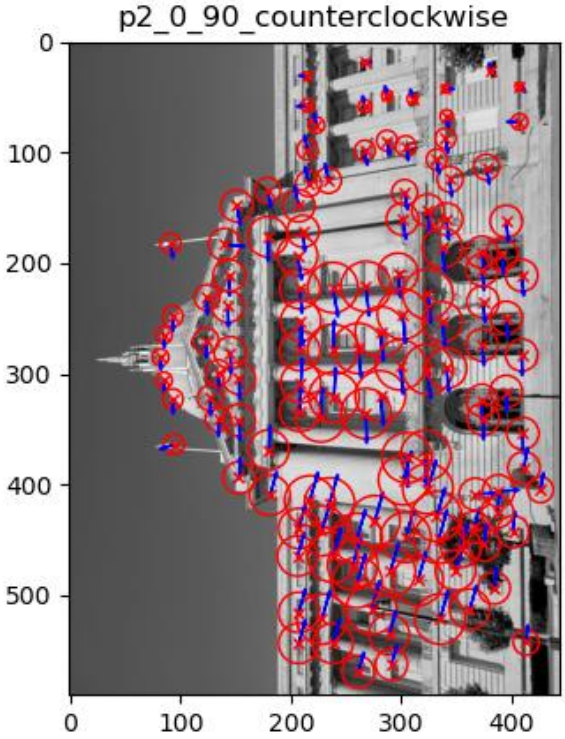
- Original image
- Each of the five modified images (shift, rotate, scale)

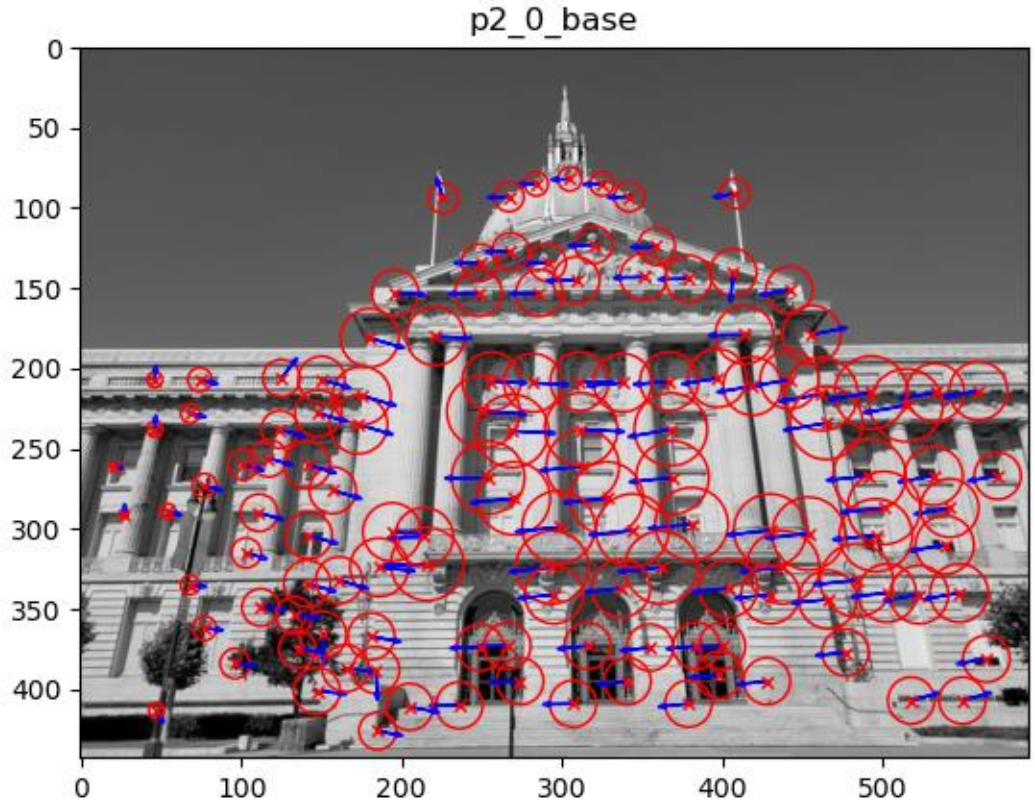
You will provide the following as further discussion overall:

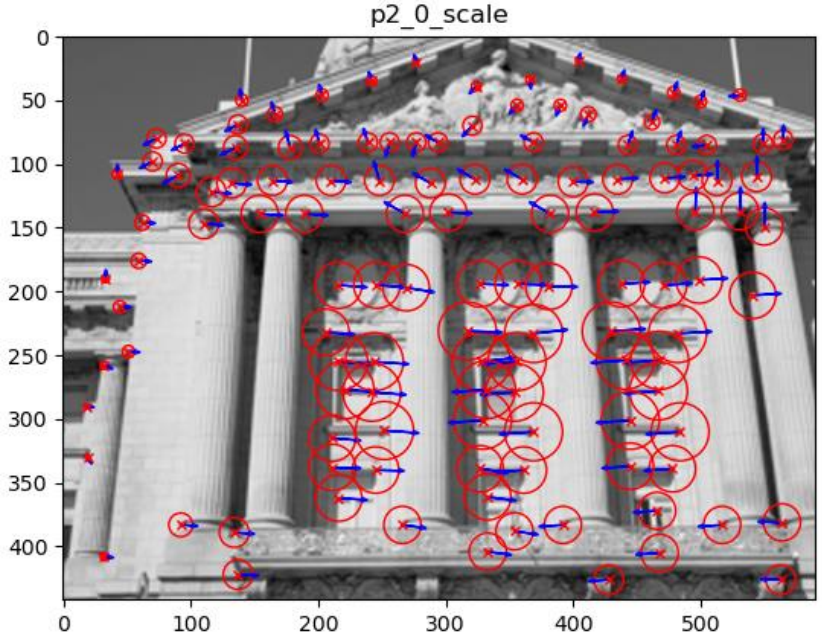
- Explanation of any "interesting" implementation choices that you made.

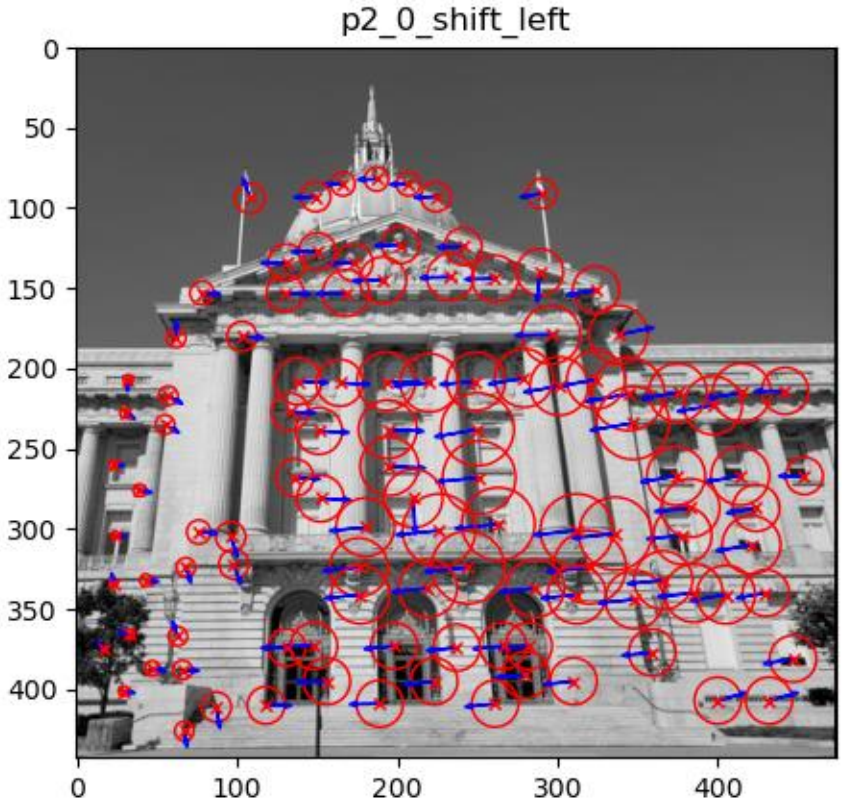
**Example 1:**

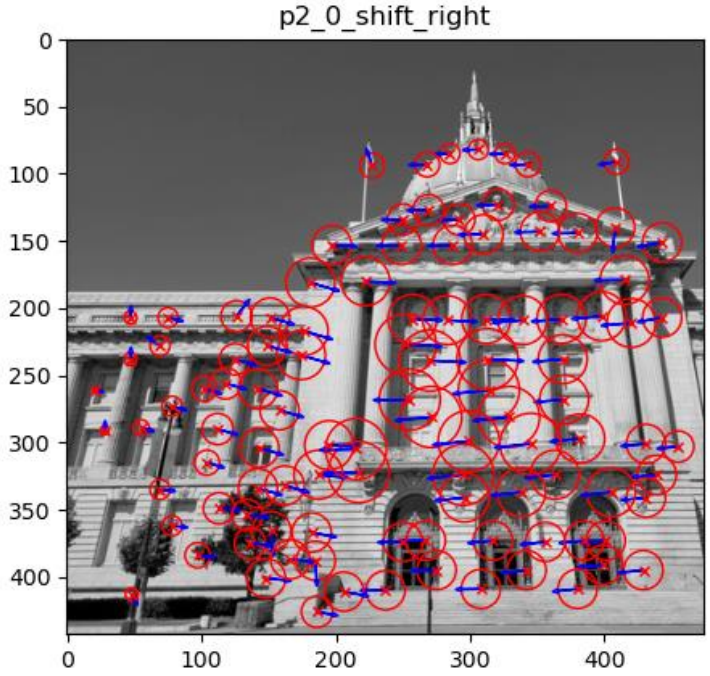

p2_0_90_clockwise
0
100
200
300
400
500
0
100
200
300
400


p2_0_90_counterclockwise
0
100
200
300
400
500
0
100
200
300
400

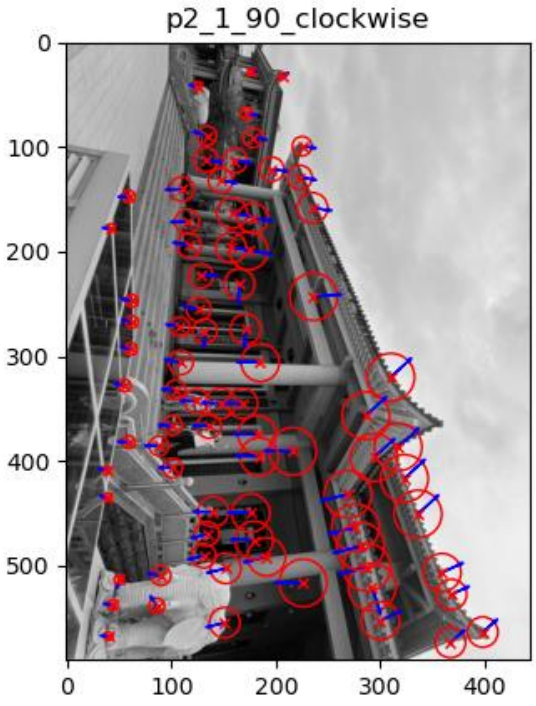

p2_0_base
0
50
100
150
200
250
300
350
400
0
100
200
300
400
500

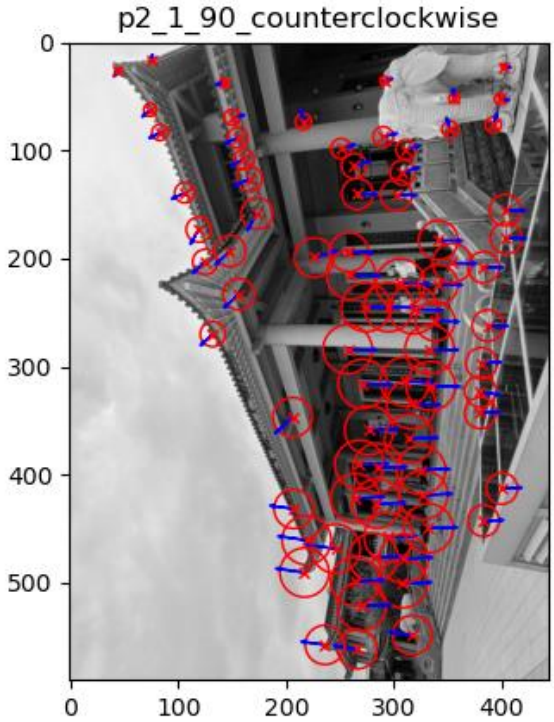

p2_0_scale
0
50
100
150
200
250
300
350
400
0
100
200
300
400
500

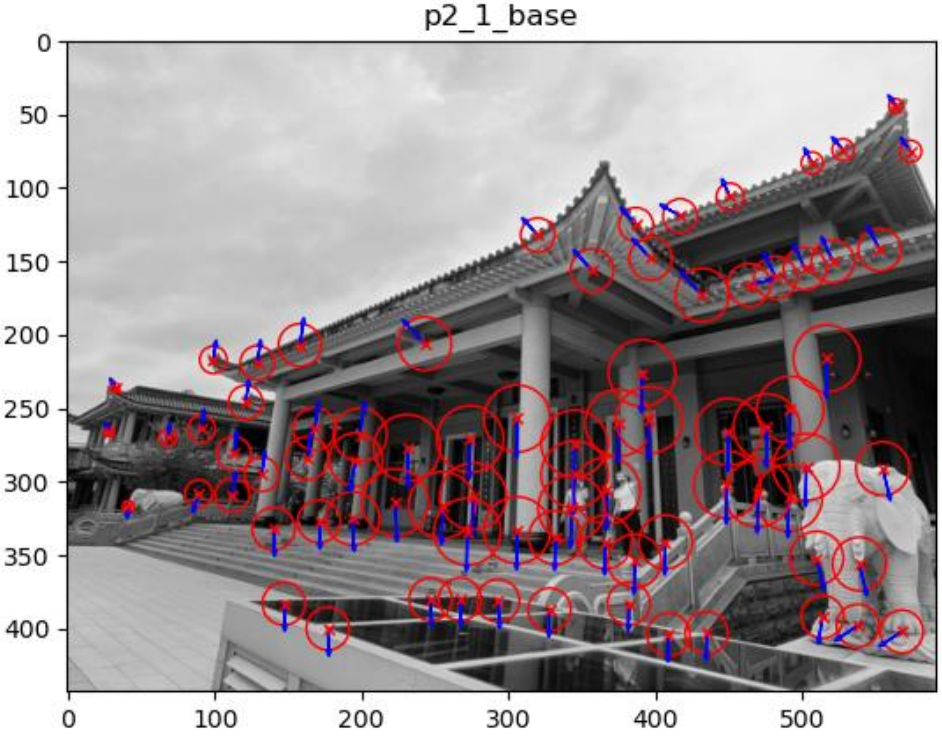

p2_0_shift_left
0
50
100
150
200
250
300
350
400
0
100
200
300
400

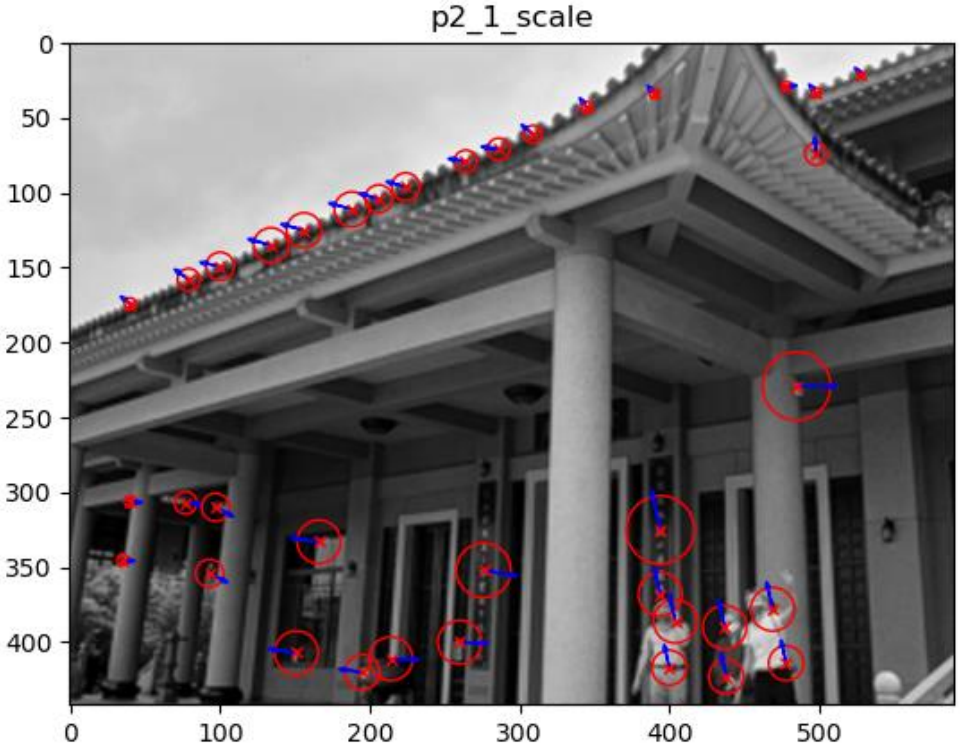

p2_0_shift_right
0
50
100
150
200
250
300
350
400
0
100
200
300
400

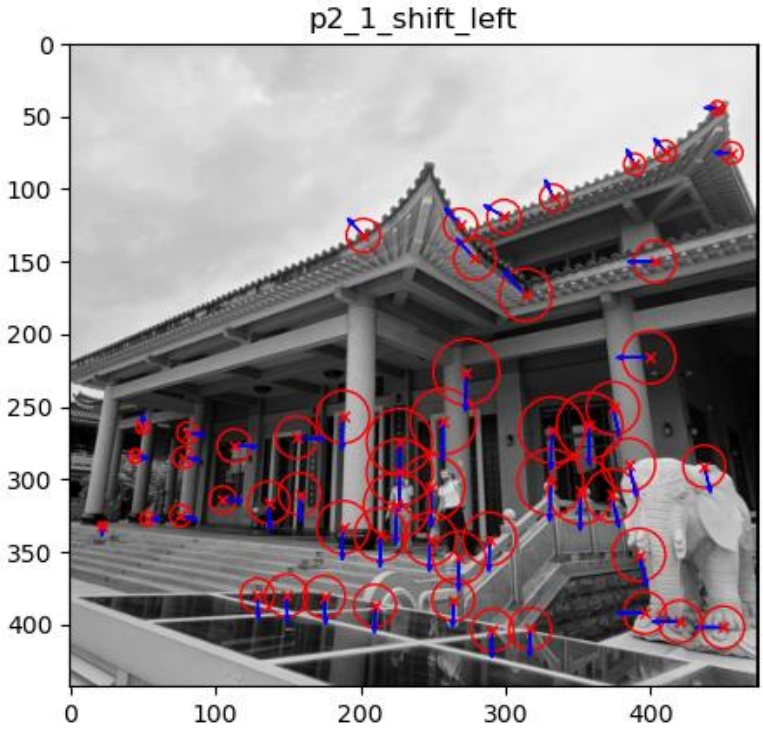
**Example 2:**

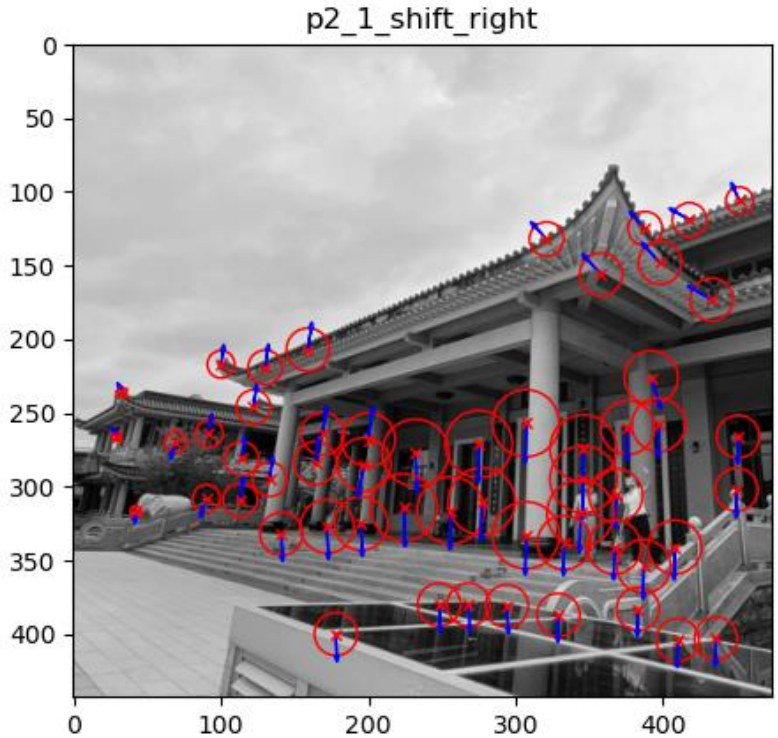

p2_1_90_clockwise
0
100
200
300
400
500
0
100
200
300
400


p2_1_90_counterclockwise
0
100
200
300
400
500
0
100
200
300
400

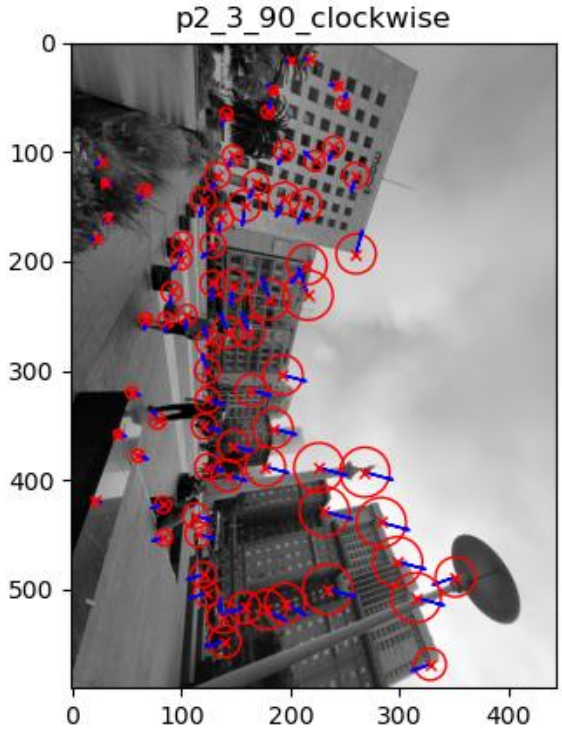

p2_1_base
0
50
100
150
200
250
300
350
400
0
100
200
300
400
500

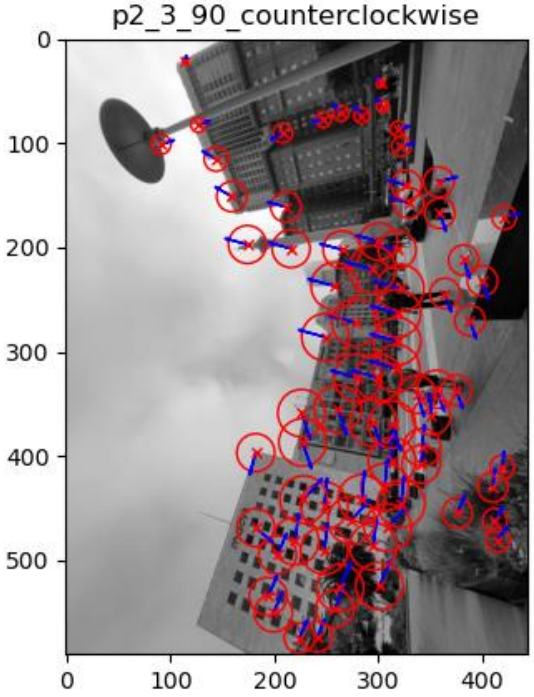

p2_1_scale
0
50
100
150
200
250
300
350
400
0
100
200
300
400
500

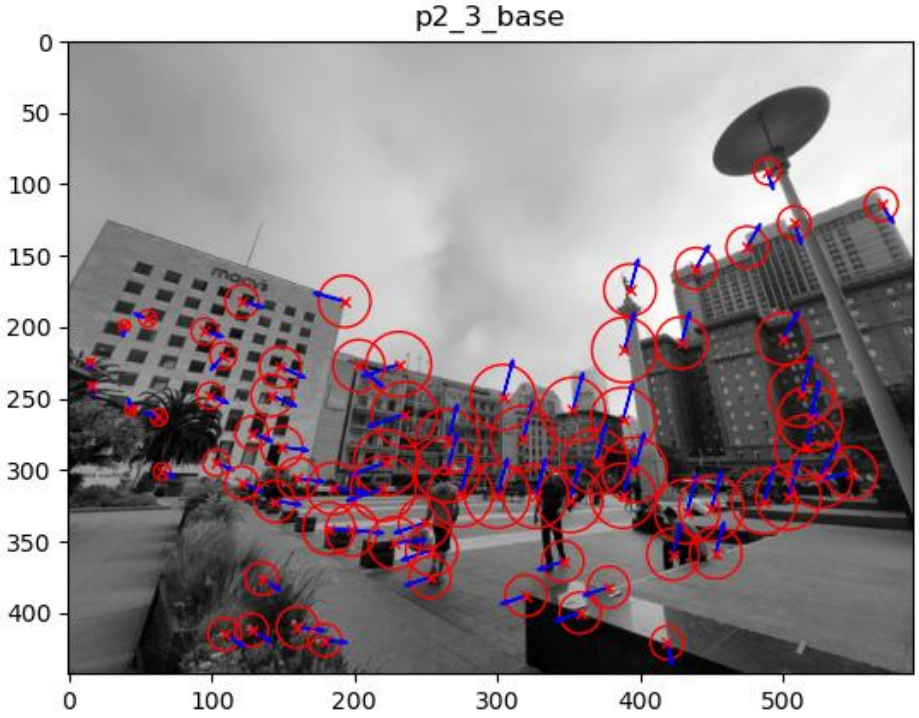

p2_1_shift_left
0
50
100
150
200
250
300
350
400
0
100
200
300
400

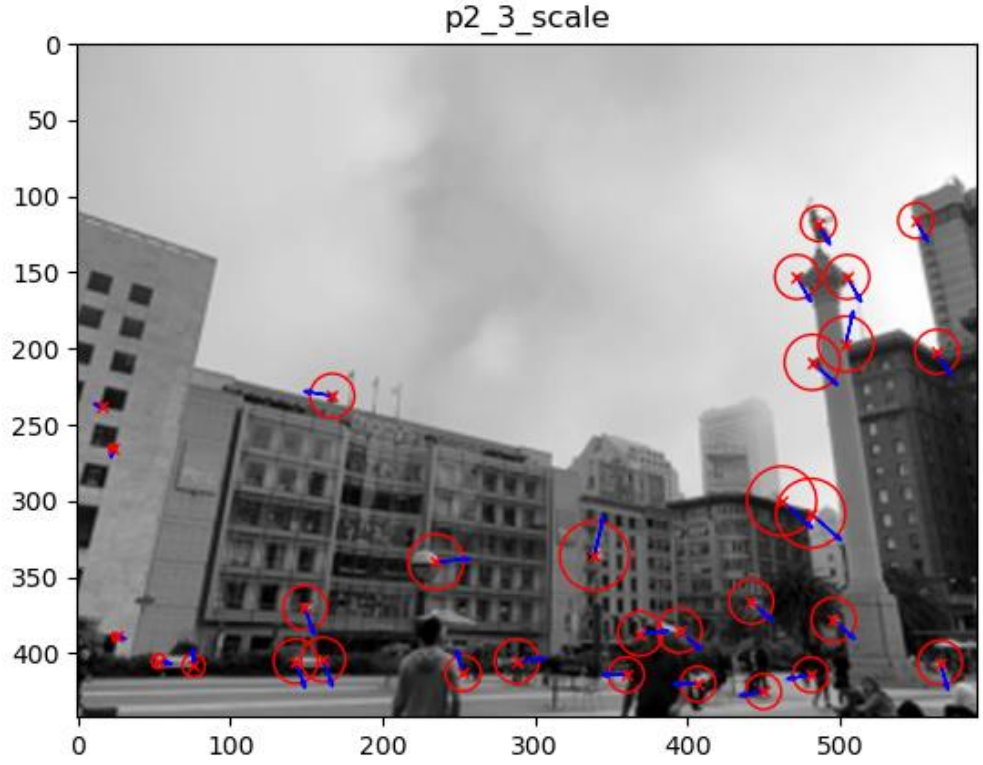

p2_1_shift_right
0
50
100
150
200
250
300
350
400
0
100
200
300
400

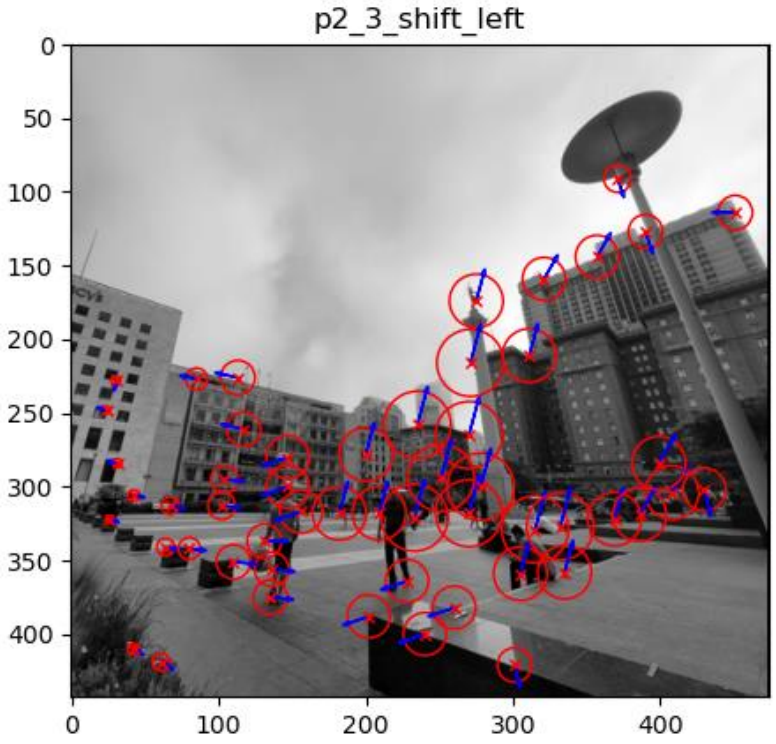
**Example 3:**

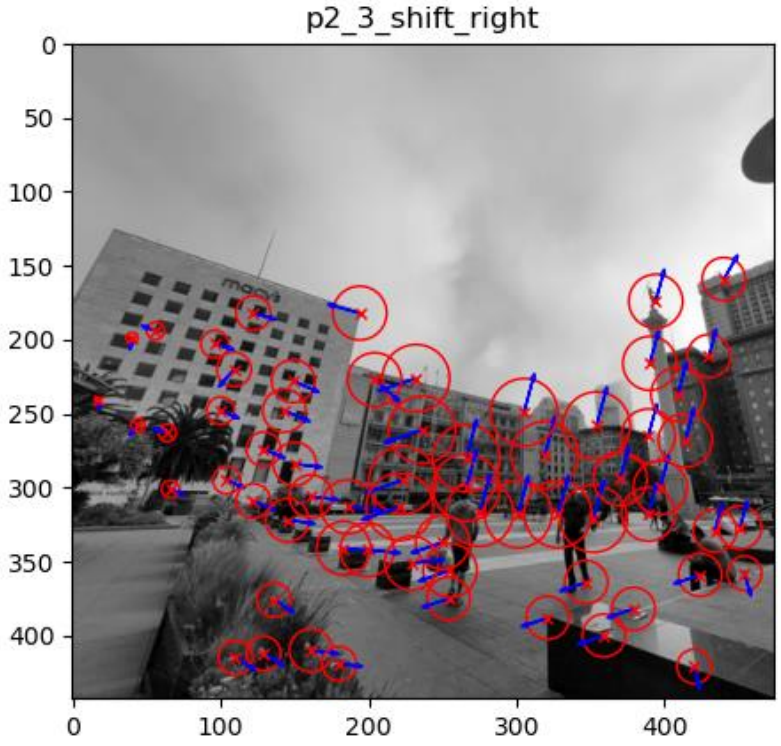

p2_3_90_clockwise
0
100
200
300
400
500
0
100
200
300
400


p2_3_90_counterclockwise
0
100
200
300
400
500
0
100
200
300
400

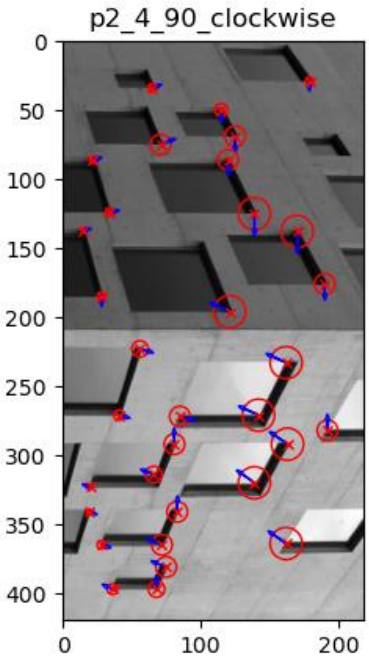

p2_3_base
0
50
100
150
200
250
300
350
400
0
100
200
300
400
500

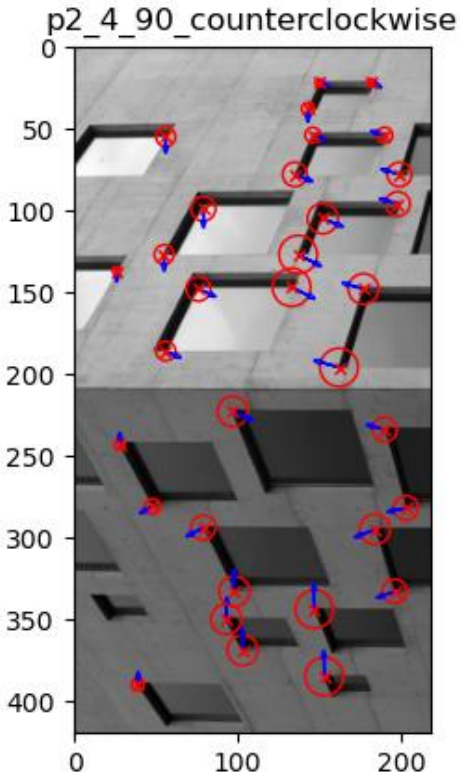

p2_3_scale
0
50
100
150
200
250
300
350
400
0
100
200
300
400
500

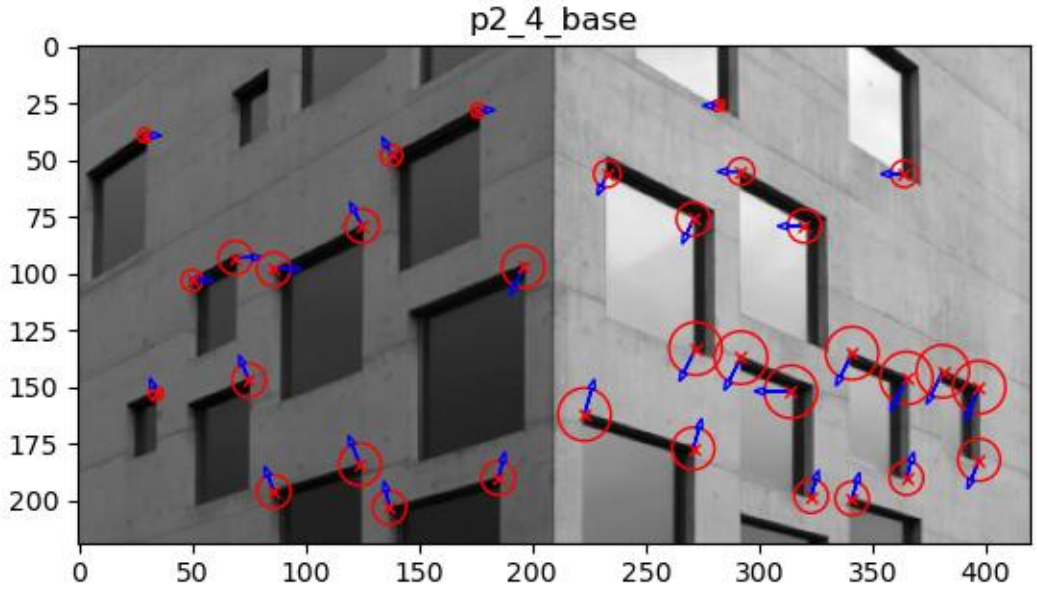

p2_3_shift_left
0
50
100
150
200
250
300
350
400
0
100
200
300
400

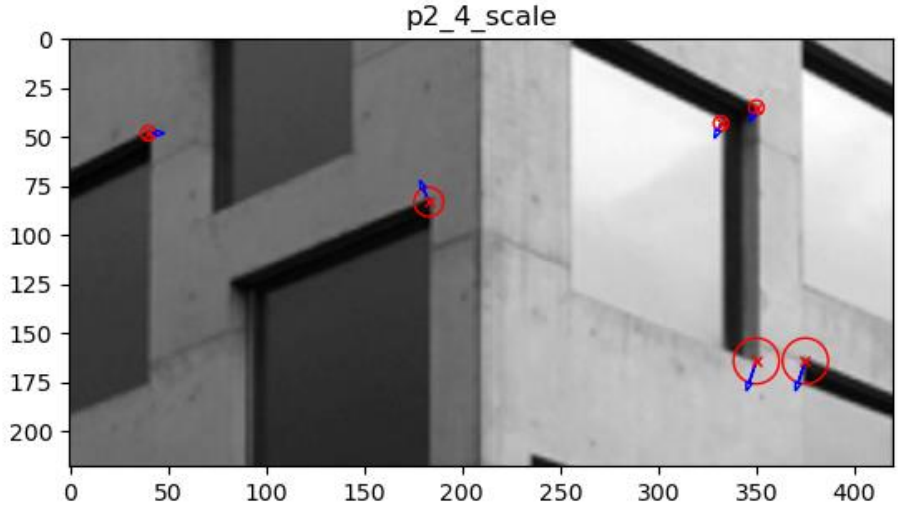

p2_3_shift_right
0
50
100
150
200
250
300
350
400
0
100
200
300
400

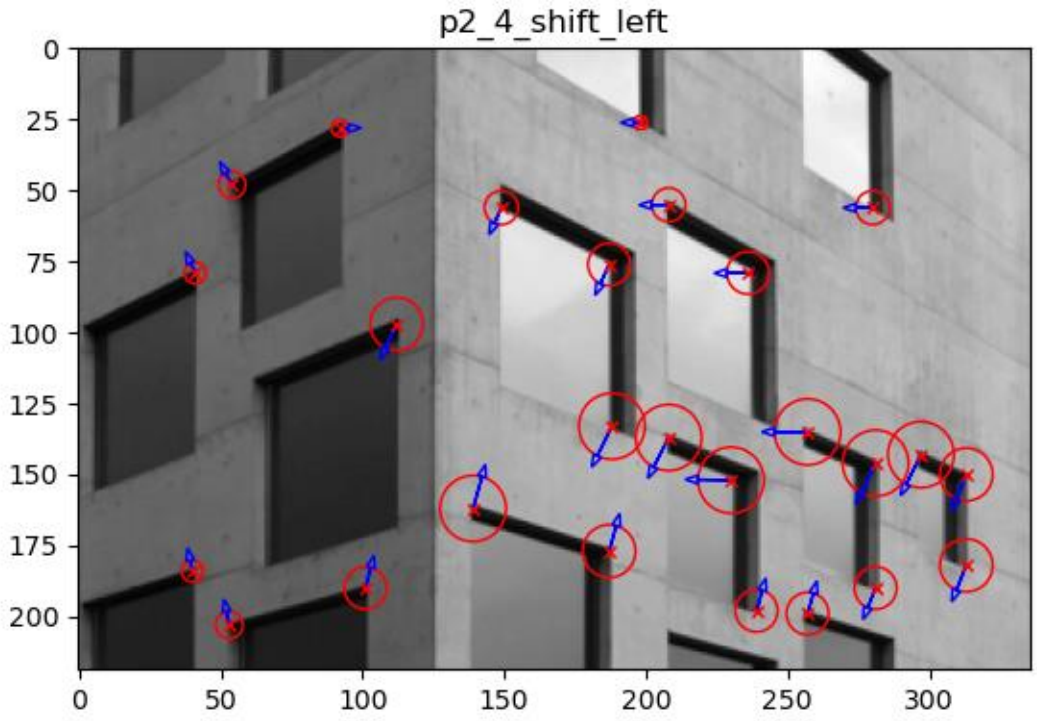
**Example 4:**

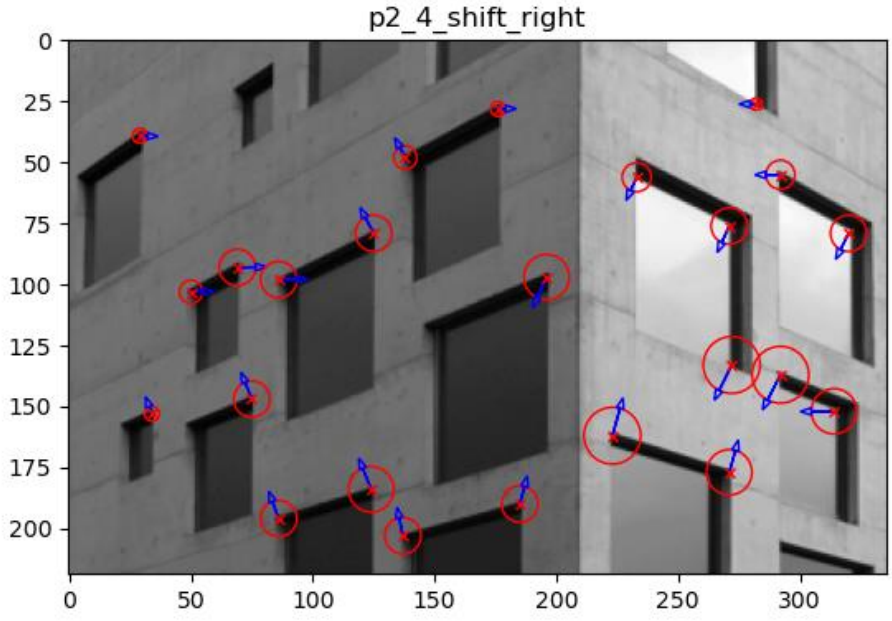

p2_4_90_clockwise
0
50
100
150
200
250
300
350
400
0
100
200


p2_4_90_counterclockwise
0
50
100
150
200
250
300
350
400
0
100
200


p2_4_base
0
25
50
75
100
125
150
175
200
0
50
100
150
200
250
300
350
400


p2_4_scale
0
25
50
75
100
125
150
175
200
0
50
100
150
200
250
300
350
400


p2_4_shift_left
0
25
50
75
100
125
150
175
200
0
50
100
150
200
250
300


p2_4_shift_right
0
25
50
75
100
125
150
175
200
0
50
100
150
200
250
300

## Discussion:

I tried to find the orientation by both histogram method and max magnitude; and I found that the output obtained by using max magnitude is more consistence, I guess that it is because the Harris corner might find different corner after rotation or scaling and change the histogram value.

# Bonus:

## Blob-Detection Extra Credit

- Discussion and results of any extensions or bonus features you have implemented for Blob-Detection