CMU Fall24 16820 Homework 2

Patrick Chen

September 21, 2024

Q1.1 at page 2

Ans:

1. It is the partial derivative of warping of x with respect to the parameters, which is :

$\frac{\partial W(x;p)}{\partial \mathbf{p}^T} = \frac{\partial \begin{bmatrix} x + p_x \\ y + p_y \end{bmatrix}}{\partial p^T}$ with $\mathbf{p} = \begin{bmatrix} p_x \\ p_y \end{bmatrix}$

$\rightarrow \frac{\partial W(x;p)}{\partial \mathbf{p}^T} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

2. A and b according to equation (5) are as the following:

A = $\sum_{x \in \mathbb{N}} \nabla I \frac{\partial W(x;p)}{\partial \mathbf{p}^T}$, where $\nabla I = \frac{\partial \mathcal{I}_{t+1}(x')}{\partial \mathbf{x}'^T}$, and x' = W(x; p)

$= \begin{bmatrix} \frac{\partial I_{t+1}(x'_1)}{\partial x'} & \frac{\partial I_{t+1}(x'_1)}{\partial y'} \\ \frac{\partial I_{t+1}(x'_2)}{\partial x'} & \frac{\partial I_{t+1}(x'_2)}{\partial y'} \\ \vdots & \vdots \\ \frac{\partial I_{t+1}(x'_D)}{\partial x'} & \frac{\partial I_{t+1}(x'_D)}{\partial y'} \end{bmatrix}$

b = $\mathcal{I}_t(\mathbf{x}) - \mathcal{I}_{t+1}(\mathbf{x}')$, where x' = W(x; p)

$= \begin{bmatrix} I_t(x_1) - I_{t+1}(x_1 + \mathbf{p}) \\ I_t(x_2) - I_{t+1}(x_2 + \mathbf{p}) \\ \vdots \\ I_t(x_D) - I_{t+1}(x_D + \mathbf{p}) \end{bmatrix}$

3. $A^T A$ must be invertible, in other words, must be full rank, that is, the matrix A must be full rank.

Q1.2 at page 2

Ans:

The code is implemented in the file LucasKanade.py, and the results are shown in Q1.3 and Q1.4 in the following pages.

Q1.3 at page 2

Ans:

The tracking results of carseqrects.npy at frames 1, 100, 200, 300, and 400 are shown in Figure 1 and Figure 2 below with –threshold setting to 1e-4 and 1e-5 respectively. The command line to regenerate the following results can be found in README chapter at the end of this writeup.

Figure 1: Car Seq Tracking Results Frames 1, 100, 200, 300, 400 (left to right) w/ –threshold=1e-4

Figure 2: Car Seq Tracking Results Frames 1, 100, 200, 300, 400 (left to right) w/ –threshold=1e-5

The tracking results of girleqrects.npy at frames 1, 20, 30, 60 and 80 are shown in Figure 3 and Figure 4 below with –threshold setting to 1e-4 and 1e-5 respectively. The command line to regenerate the following results can be found in README chapter at the end of this writeup.

Figure 3: Girl Seq Tracking Results Frames 1, 20, 30, 60, 80 (left to right) w/ –threshold=1e-4

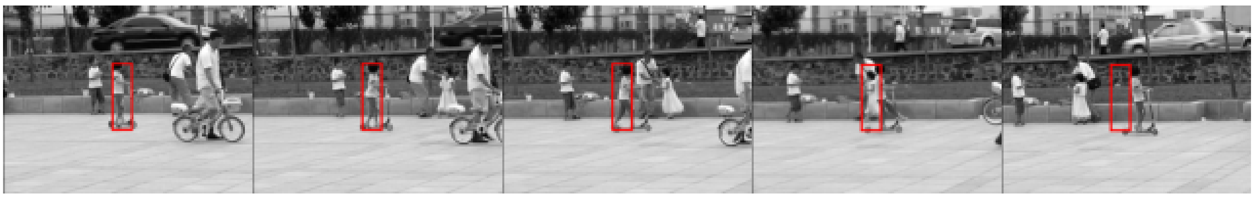

Figure 4: Girl Seq Tracking Results Frames 1, 20, 30, 60, 80 (left to right) w/ –threshold=1e-5

As I decrease the –threshold parameter, the tracking performance would improve as the red rectangles would capture the target more closely. It is particularly obvious on carseqrects.npy

Q1.4 at page 3

Ans:

The tracking results of carseqrects.npy at frames 1, 100, 200, 300, and 400 are shown in the following Figure 5 - Figure 10 below with –threshold setting to $1e{-}4$ and $1e{-}5$ respectively, and –template_threshold setting to 1, 5, and 10 respectively. The blue rectangles are the results without drifting correction (Q1.3), and the red squares are the results with drifting correction (Q1.4). The command line to regenerate the following results can be found in README chapter at the end of this writeup.

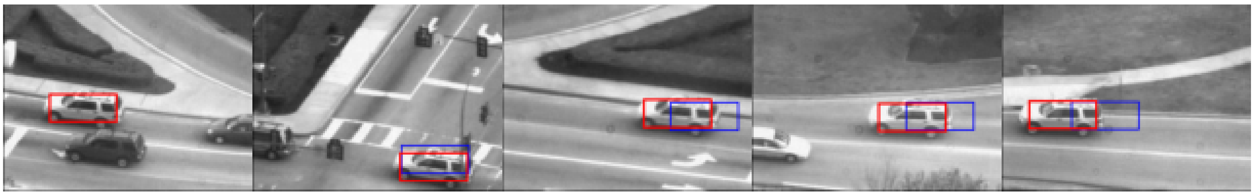

Figure 5: Car Seq Tracking Results Frames 1, 100, 200, 300, 400 (left to right) w/ –threshold=1e-4 –template_threshold=1

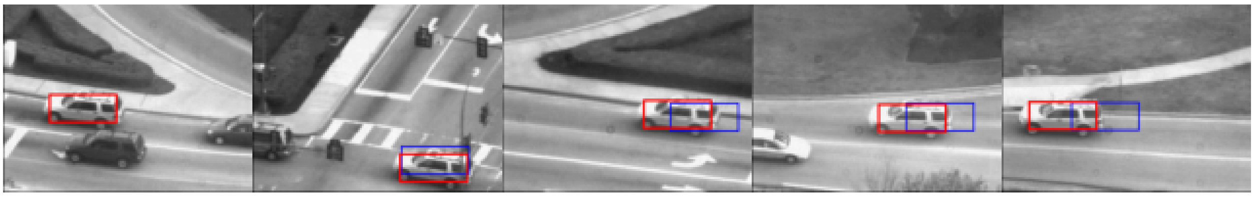

Figure 6: Car Seq Tracking Results Frames 1, 100, 200, 300, 400 (left to right) w/ –threshold=1e-4 –template_threshold=5

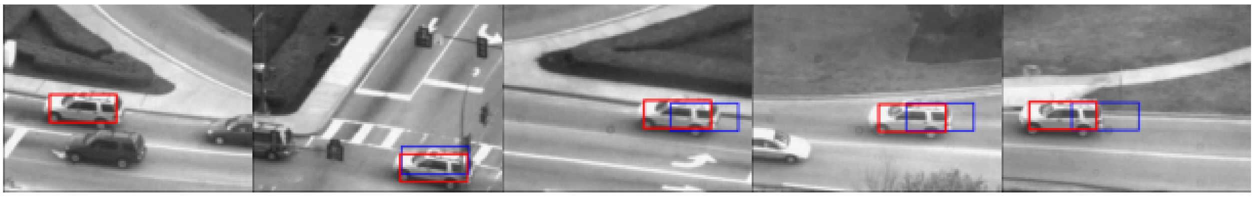

Figure 7: Car Seq Tracking Results Frames 1, 100, 200, 300, 400 (left to right) w/ –threshold=1e-4 –template_threshold=10

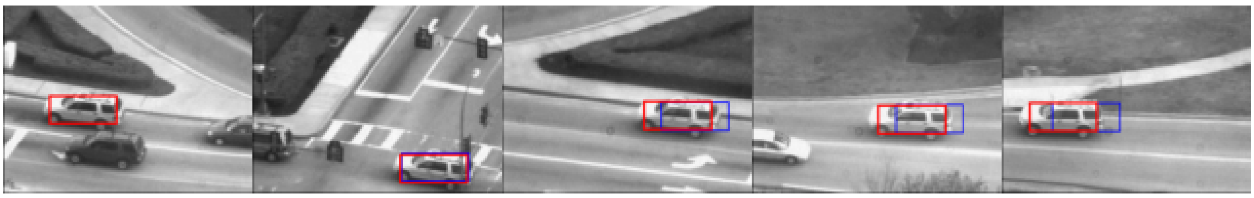

Figure 8: Car Seq Tracking Results Frames 1, 100, 200, 300, 400 (left to right) w/ –threshold=1e-5 –template_threshold=1

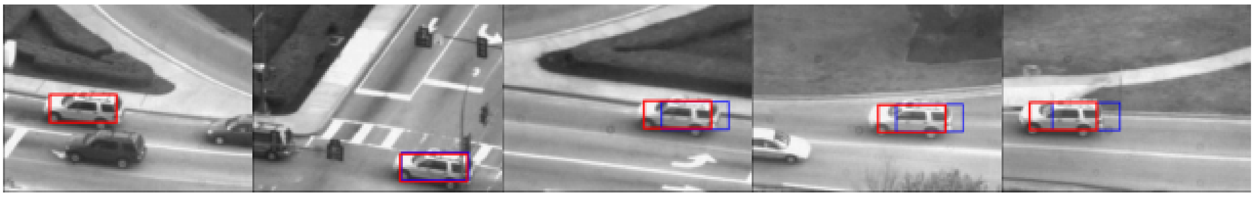

Figure 9: Car Seq Tracking Results Frames 1, 100, 200, 300, 400 (left to right) w/ –threshold=1e-5 –template_threshold=5

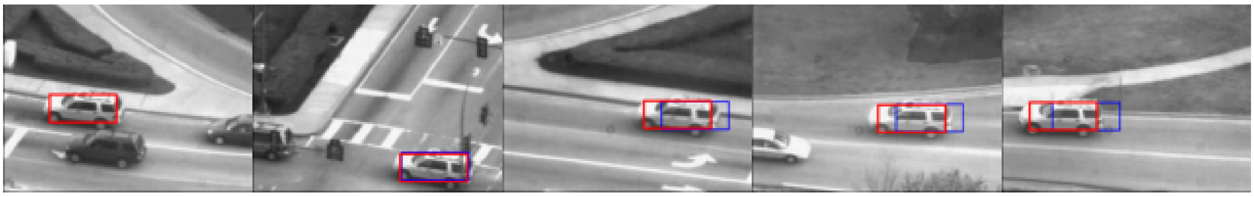

Figure 10: Car Seq Tracking Results Frames 1, 100, 200, 300, 400 (left to right) w/ –threshold=1e-5 –template_threshold=10

The tracking results of girlseqrects.npy at frames 1, 20, 40, 60 and 80 are shown in the following Figure 11 - Figure 16 below with –threshold setting to $1e-4$ and $1e-5$ respectively, and –template_threshold setting to 1, 5, and 10 respectively. The blue rectangles are the results without drifting correction (Q1.3), and the red squares are the results with drifting correction (Q1.4). The command line to regenerate the following results can be found in README chapter at the end of this writeup.

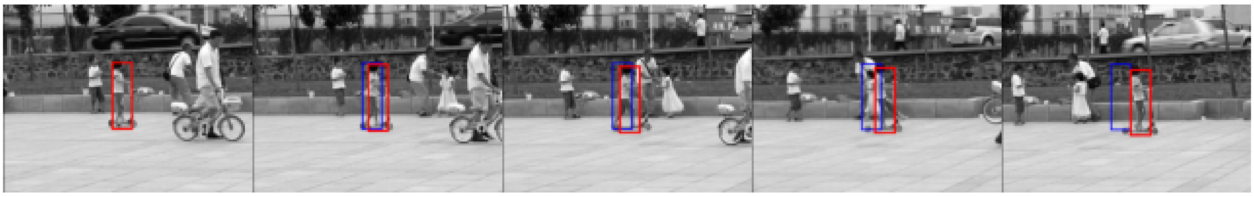

Figure 11: Car Seq Tracking Results Frames 1, 20, 40, 60, 80 (left to right) w/ –threshold=1e-4 –template_threshold=1

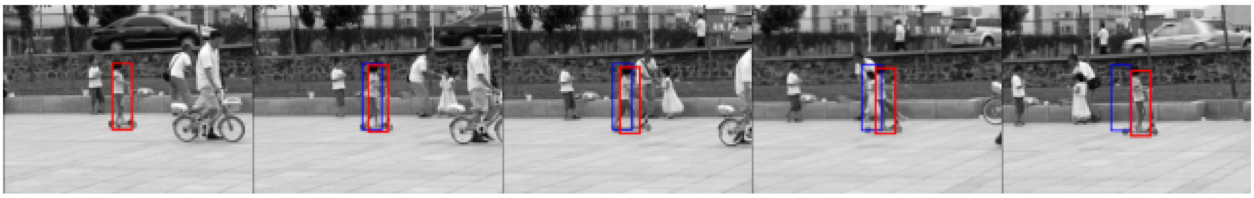

Figure 12: Girl Seq Tracking Results Frames 1, 20, 40, 60, 80 (left to right) w/ –threshold=1e-4 –template_threshold=5

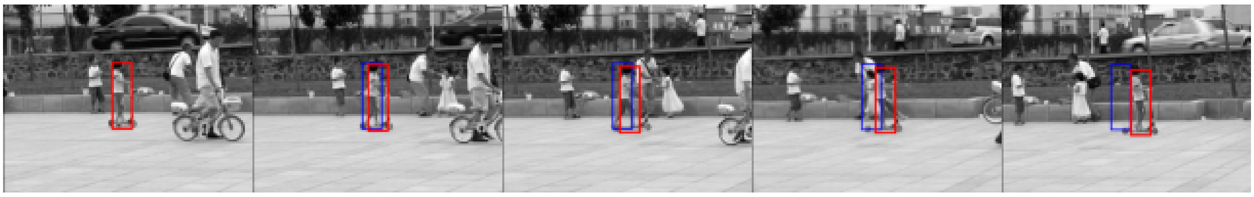

Figure 13: Girl Seq Tracking Results Frames 1, 20, 40, 60, 80 (left to right) w/ –threshold=1e-4 –template_threshold=10

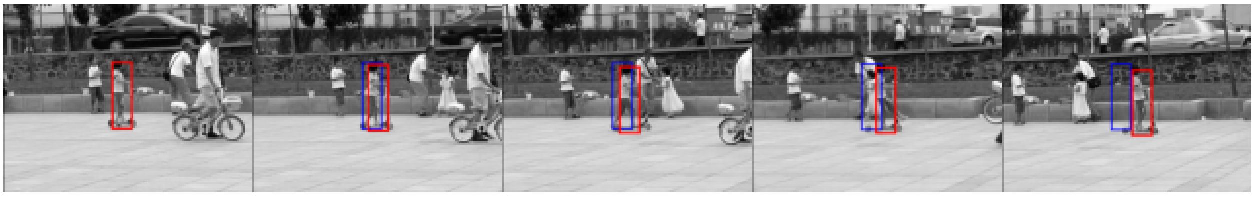

Figure 14: Girl Seq Tracking Results Frames 1, 20, 40, 60, 80 (left to right) w/ –threshold=1e-5 –template_threshold=1

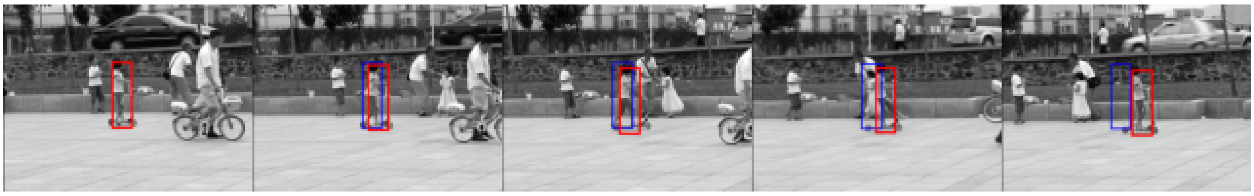

Figure 15: Girl Seq Tracking Results Frames 1, 20, 40, 60, 80 (left to right) w/ –threshold=1e-5 –template_threshold=5

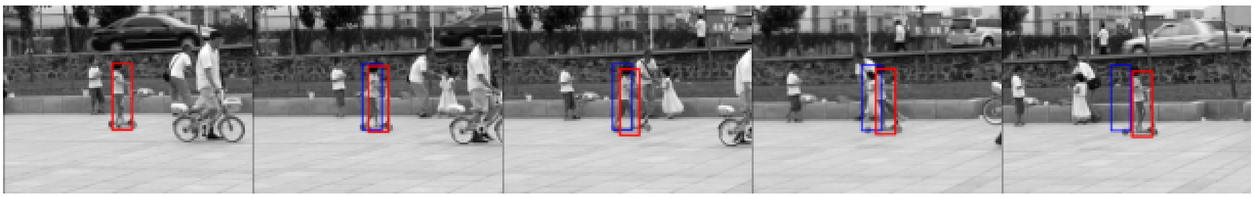

Figure 16: Girl Seq Tracking Results Frames 1, 20, 40, 60, 80 (left to right) w/ –threshold=1e-5 –template_threshold=10

As I decrease the –threshold parameter, the tracking performance would improve as the red rectangles would capture the target more closely. On the other hand, when I decrease or increase the –template_threshold, there is no big difference on the tracking results.

Q2.1 at page 4

Ans:

The code is implemented in the file LucasKanadeAffine.py, and the results are shown in Q2.3 in the following pages.

Q2.2 at page 4

Ans:

The code is implemented in the file SubstractDominantMotion.py, and the results are shown in Q2.3 in the following pages.

Q2.3 at page 5

Ans:

Following are the results of aerialseq.npy on frames 30, 60, 90, and 120. The command line to regenerate the following results can be found in README chapter at the end of this writeup. The parameter –threshold is set to 1e-11, and the parameter –tolerance is set to 0.064.

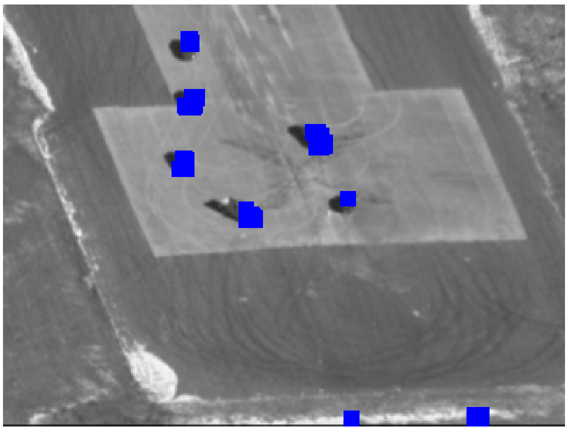

Figure 17: Aerial Sequence Frame 30 LucasKanadeAffine() Result

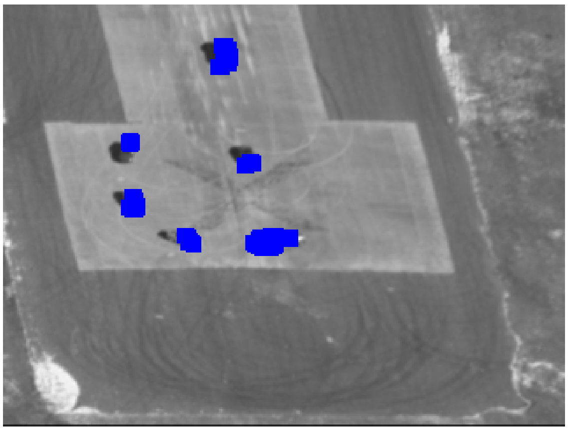

Figure 18: Aerial Sequence Frame 60 LucasKanadeAffine() Result

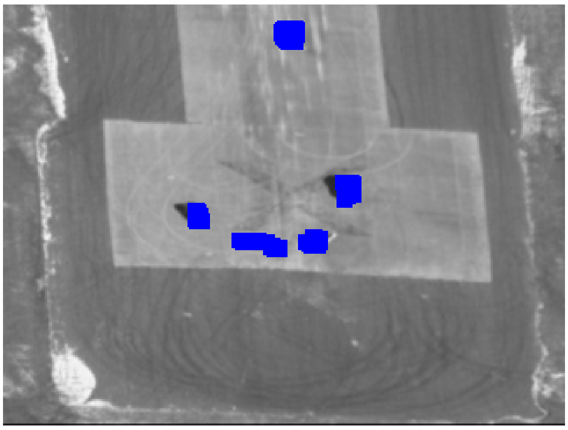

Figure 19: Aerial Sequence Frame 90 LucasKanadeAffine() Result

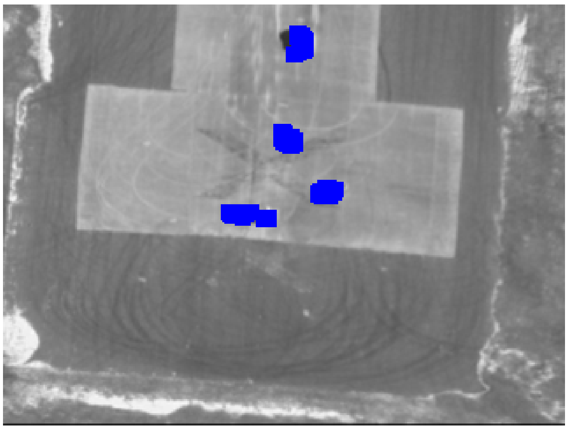

Figure 20: Aerial Sequence Frame 120 LucasKanadeAffine() Result

Following are the results of antseq.npy on frames 30, 60, 90, and 120. The command line to regenerate the following results can be found in README chapter at the end of this writeup. The parameter –threshold is set to 1e-11, and the parameter –tolerance is set to 0.025.

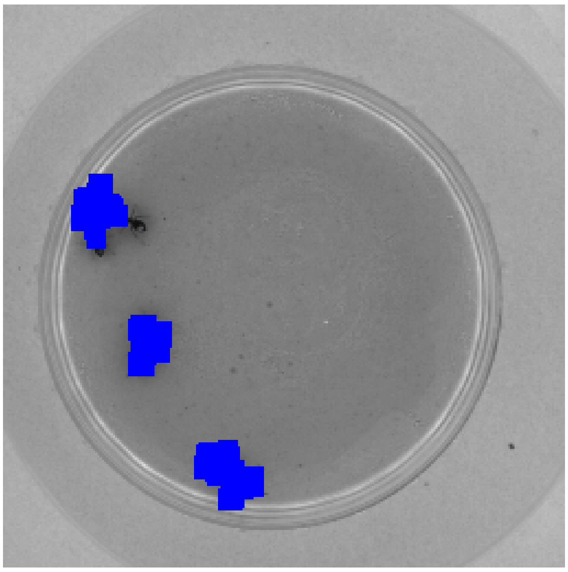

Figure 21: Ant Sequence Frame 30 LucasKanadeAffine() Result

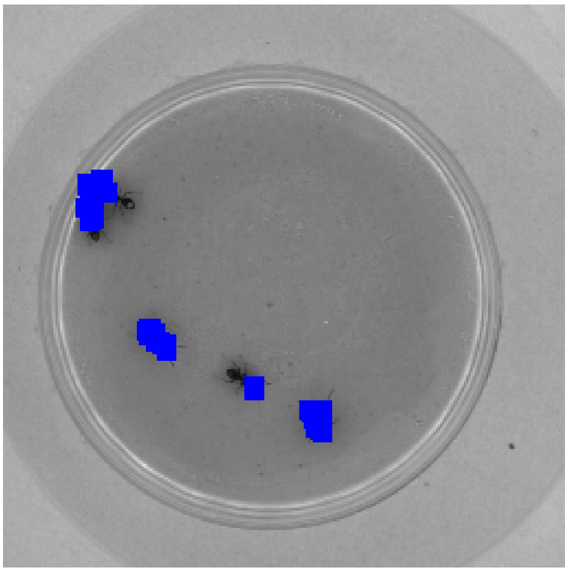

Figure 22: Ant Sequence Frame 60 LucasKanadeAffine() Result

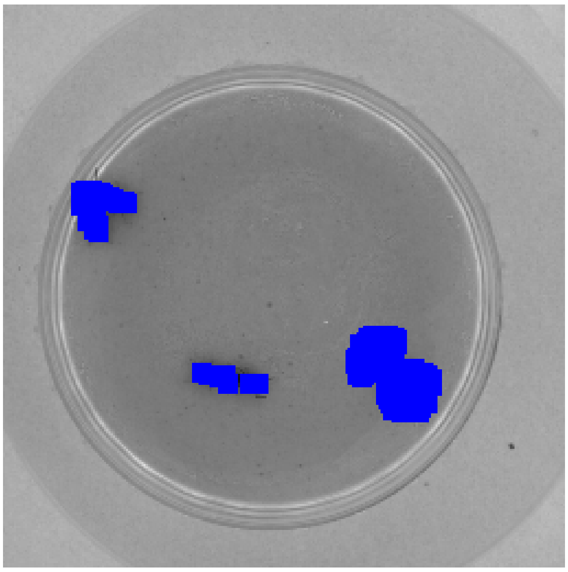

Figure 23: Ant Sequence Frame 90 LucasKanadeAffine() Result

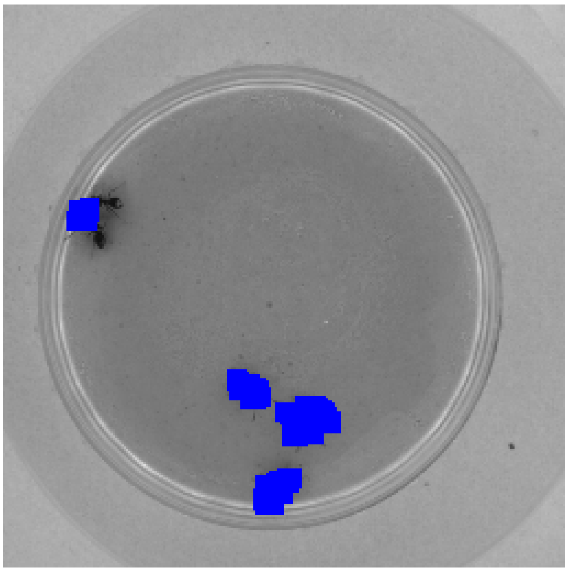

Figure 24: Ant Sequence Frame 120 LucasKanadeAffine() Result

Q3.1 at page 5

Ans:

Following are the results of aerialseq.npy after applying InverseCompositionAffine() on frames 30, 60, 90, and 120. The command line to regenerate the following results can be found in README chapter at the end of this writeup. The parameter –threshold is set to 1e-11, and the parameter –tolerance is set to 0.064.

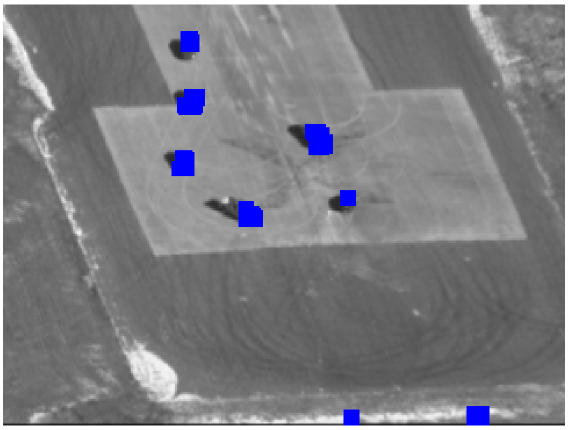

Figure 25: Aerial Sequence Frame 30 InverseCompositionAffine() Result

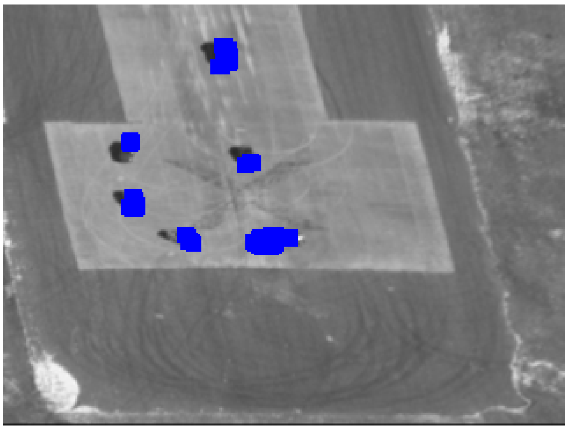

Figure 26: Aerial Sequence Frame 60 InverseCompositionAffine() Result

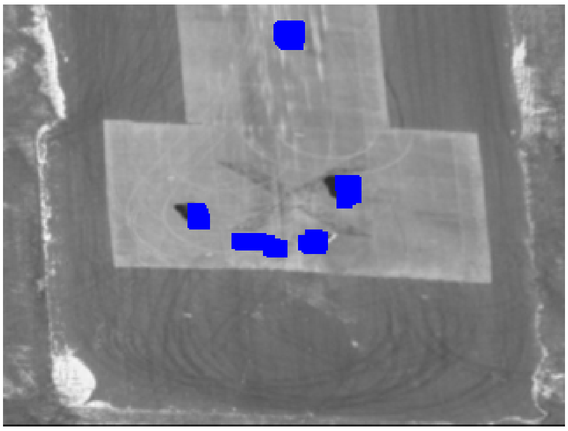

Figure 27: Aerial Sequence Frame 90 InverseCompositionAffine() Result

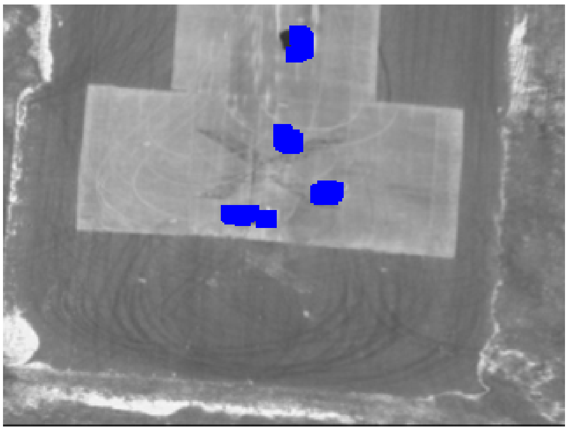

Figure 28: Aerial Sequence Frame 120 InverseCompositionAffine() Result

Following are the results of antseq.npy after applying InverseCompositionAffine() on frames 30, 60, 90, and 120. The command line to regenerate the following results can be found in README chapter at the end of this writeupe. The parameter –threshold is set to 1e-11, and the parameter –tolerance is set to 0.025.

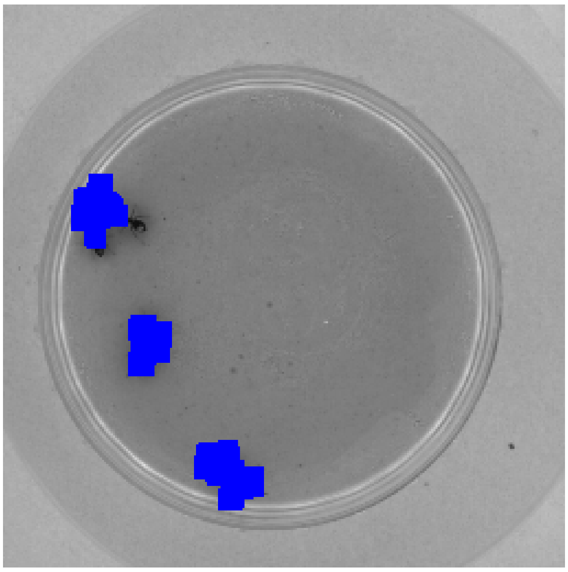

Figure 29: Ant Sequence Frame 30 InverseCompositionAffine() Result

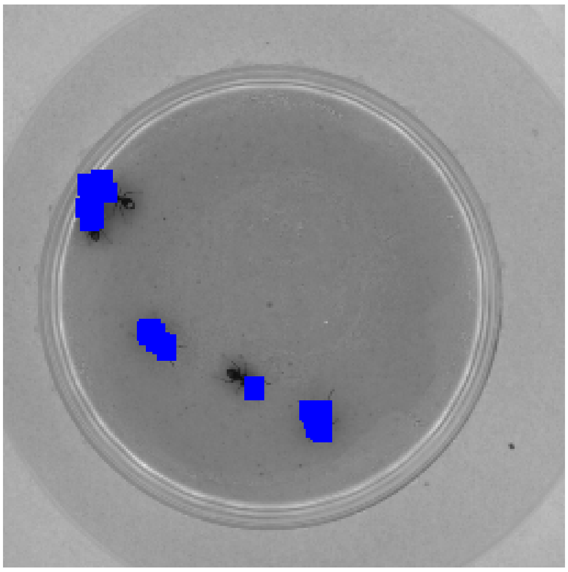

Figure 30: Ant Sequence Frame 60 InverseCompositionAffine() Result

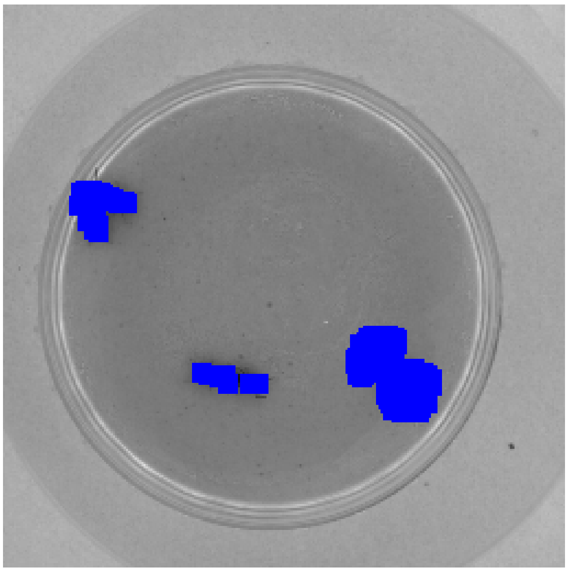

Figure 31: Ant Sequence Frame 90 InverseCompositionAffine() Result

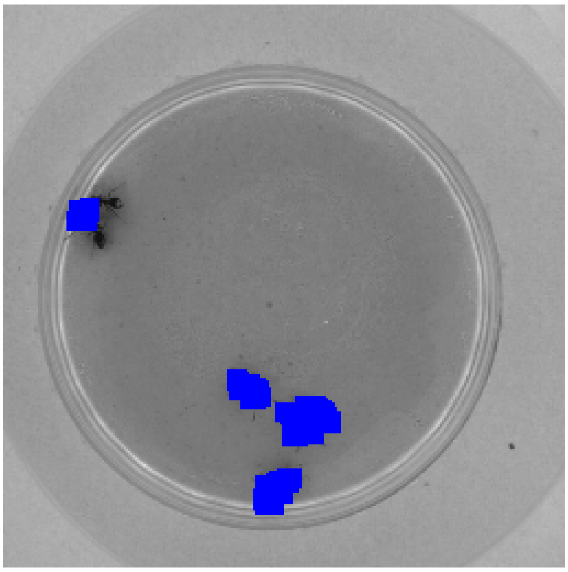

Figure 32: Ant Sequence Frame 120 InverseCompositionAffine() Result

Q3.2 at page 5

Ans:

The reason that inverse compositional approach is more computationally efficient than the classical approach is that it takes the calculation of Hession Matrix, Hession Matrix Inverse, Jacobian Matrix, and steepest descent images $\nabla T \frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ out of the Δp iteration loop and the values now can be pre-computed. Besides, the images on which the gradient is calculated on is changed from the warped version of It1 to the non-warped version of It (template). These are the reasons that inverse compositional is much more efficient than the classic version.

Collaborations

Ans:

Though I do not have collaborators, I found the following websites helpful on understanding the concepts in this homework.

1. `https://www.ri.cmu.edu/pub_files/pub3/baker_simon_2003_3/baker_simon_2003_3.pdf.`
2. `http://16385.courses.cs.cmu.edu/spring2024/lecture/track`
3. `https://stats.stackexchange.com/questions/559575/how-does-addition-of-a-regularization-term-ens`
4. `https://docs.opencv.org/4.x/d7/d4d/tutorial_py_thresholding.html`
5. `https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.imsave.html`
6. `https://docs.opencv.org/4.x/d4/d61/tutorial_warp_affine.html`

README

For Q1.3 Please use the following command line to reproduce the result stored in current folder My resulted figures in the writeup report is generated by first running the following commands and copy each of the generated carseqrects.npy or girlseqrects.npy to the same folder with plotRects.py, and then run 'python plotRects.py q1.3 car' and 'python plotRects.py q1.3 girl' and get the generated q1_3_car_collage.png and q1_3_girl_collage.png

```
python ./testCarSequence.py --threshold 1e-4
python ./testCarSequence.py --threshold 1e-5
python ./testGirlSequence.py --threshold 1e-4
python ./testGirlSequence.py --threshold 1e-5
```

For Q1.4 Please use the following command line to reproduce the result stored in current folder My resulted figures in the writeup report is generated by first running the following commands and copy each of the generated carseqrects-wcrt.npy or girlseqrects-wcrt.npy to the same folder with plotRects.py, and then run 'python plotRects.py q1.4 car' and 'python plotRects.py q1.4 girl' and get the generated q1_4_car_collage.png and q1_4_girl_collage.png

```
python ./testCarSequenceWithTemplateCorrection.py --threshold 1e-4 --template_threshold 1
python ./testCarSequenceWithTemplateCorrection.py --threshold 1e-4 --template_threshold 5
python ./testCarSequenceWithTemplateCorrection.py --threshold 1e-4 --template_threshold 10
python ./testCarSequenceWithTemplateCorrection.py --threshold 1e-5 --template_threshold 1
python ./testCarSequenceWithTemplateCorrection.py --threshold 1e-5 --template_threshold 5
python ./testCarSequenceWithTemplateCorrection.py --threshold 1e-5 --template_threshold 10

python ./testGirlSequenceWithTemplateCorrection.py --threshold 1e-4 --template_threshold 1
python ./testGirlSequenceWithTemplateCorrection.py --threshold 1e-4 --template_threshold 5
python ./testGirlSequenceWithTemplateCorrection.py --threshold 1e-4 --template_threshold 10
python ./testGirlSequenceWithTemplateCorrection.py --threshold 1e-5 --template_threshold 1
python ./testGirlSequenceWithTemplateCorrection.py --threshold 1e-5 --template_threshold 5
python ./testGirlSequenceWithTemplateCorrection.py --threshold 1e-5 --template_threshold 10
```

For Q2.3 and Q3.1 Please use the following command line to reproduce the result stored in the folder specified by –output folder argument. (Please mkdir the output folder first.)

```
--use_inverse 0: use LucasKanadeAffine()
--use_inverse 1: use InverseCompositionAffine()

python ./testAerialSequence.py --threshold 1e-11 --tolerance 0.064 --use_inverse 0 --output_folder
../result_aerialseq_th1e-11_tol0.064
python ./testAerialSequence.py --threshold 1e-11 --tolerance 0.064 --use_inverse 1 --output_folder
../result_aerialseq_inverse_th1e-11_tol0.064
python ./testAntSequence.py --threshold 1e-11 --tolerance 0.025 --use_inverse 0 --output_folder
../result_antseq_th1e-11_tol0.025
python ./testAntSequence.py --threshold 1e-11 --tolerance 0.025 --use_inverse 1 --output_folder
../result_antseq_inverse_th1e-11_tol0.025
```