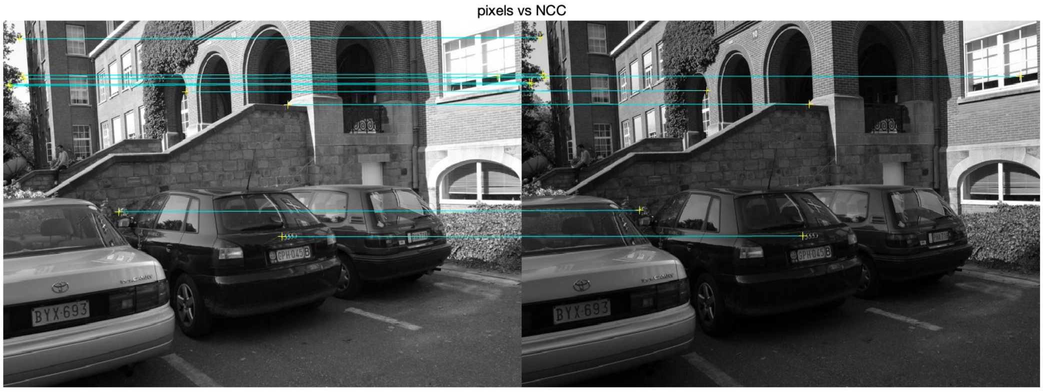
For the Lab5 report

P1:

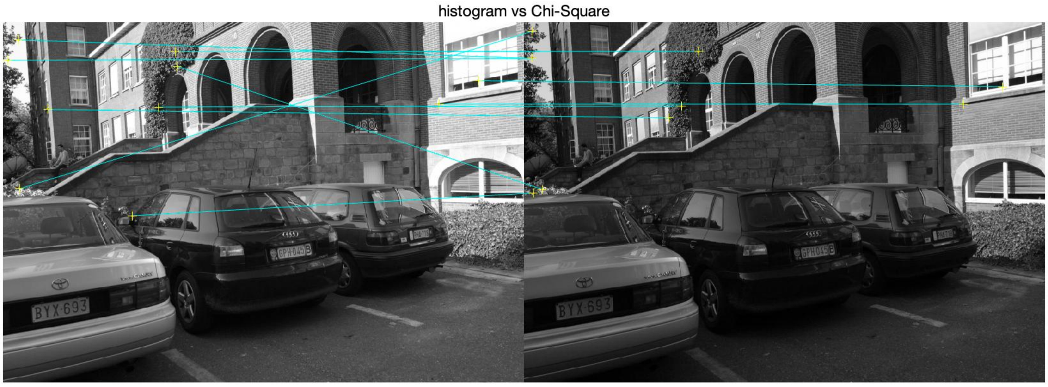
- For the images in illumination_change: SSD performs well

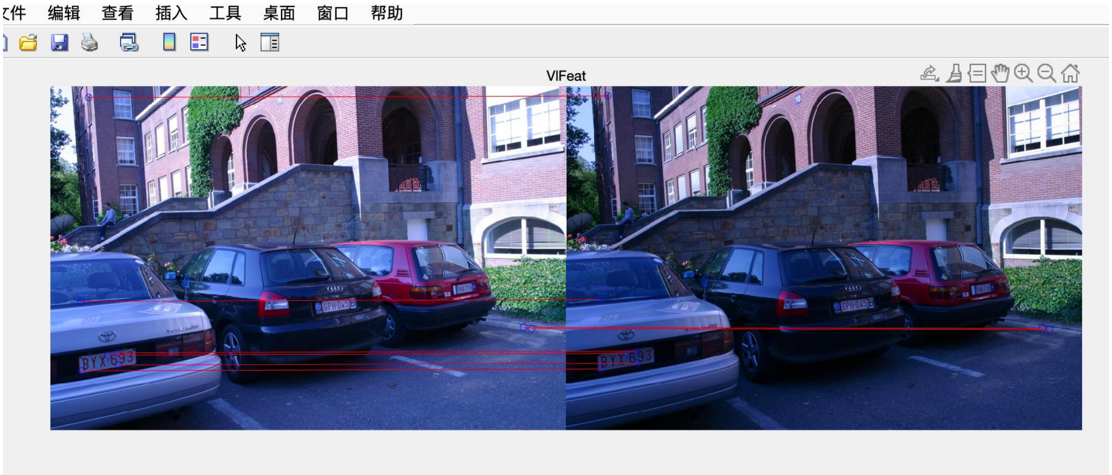
pixels vs SSD

pixels vs NCC



histogram vs Chi-Square





Accuracy for pixels vs SSD: 0.974359

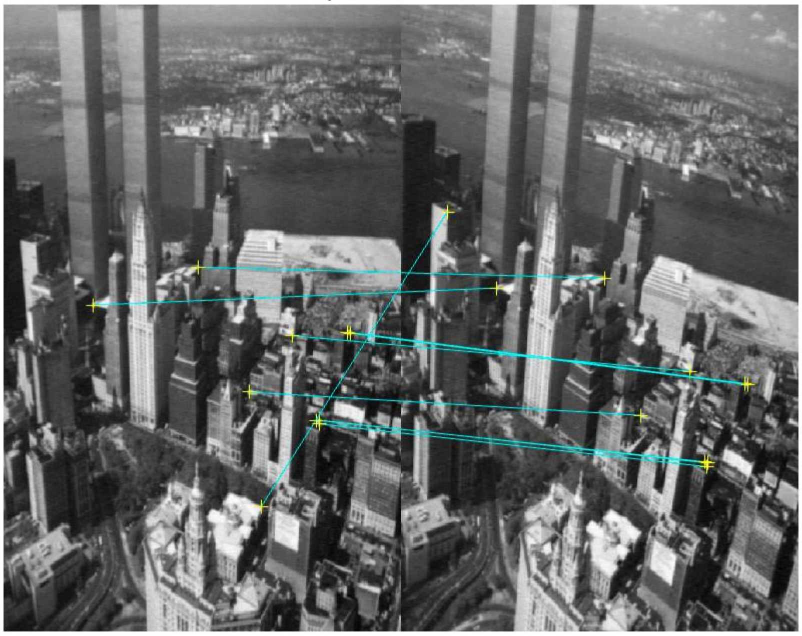
Accuracy for pixels vs NCC: 0.745000

Accuracy for histogram vs Chi-Square: 0.666667

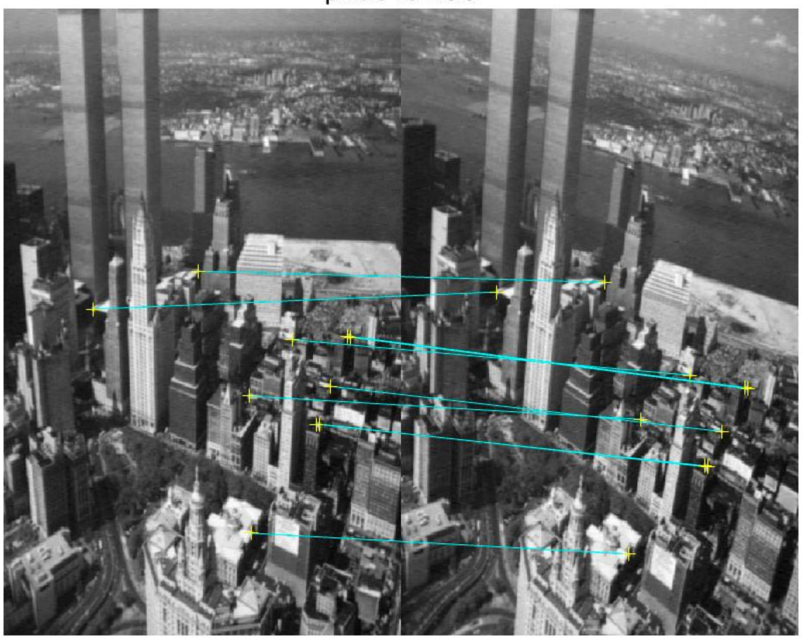
Accuracy for SIFT matches: 1.000000

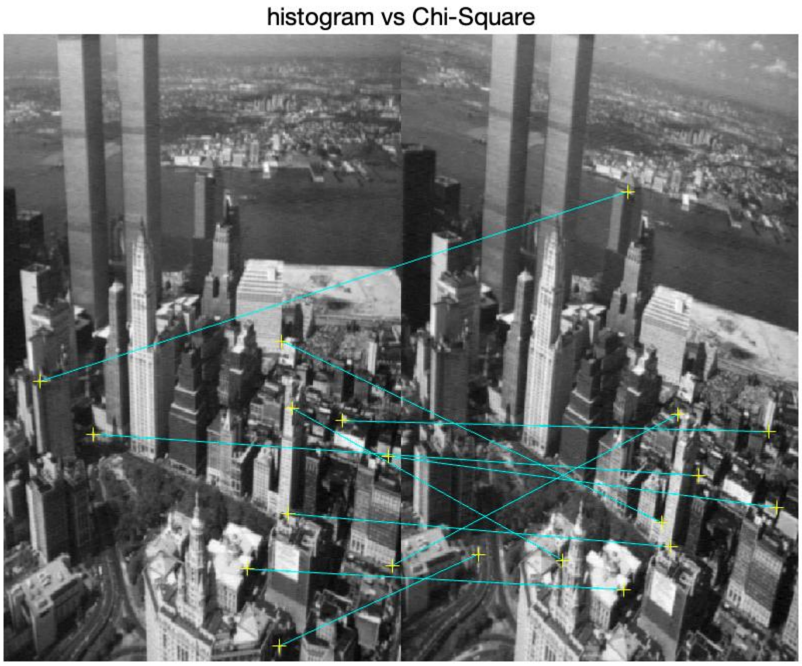
- For the images in planar_rotation_change: SSD and NCC performs well. Chi-Square is not particularly sensitive to planar rotation but was not specifically designed to handle it.

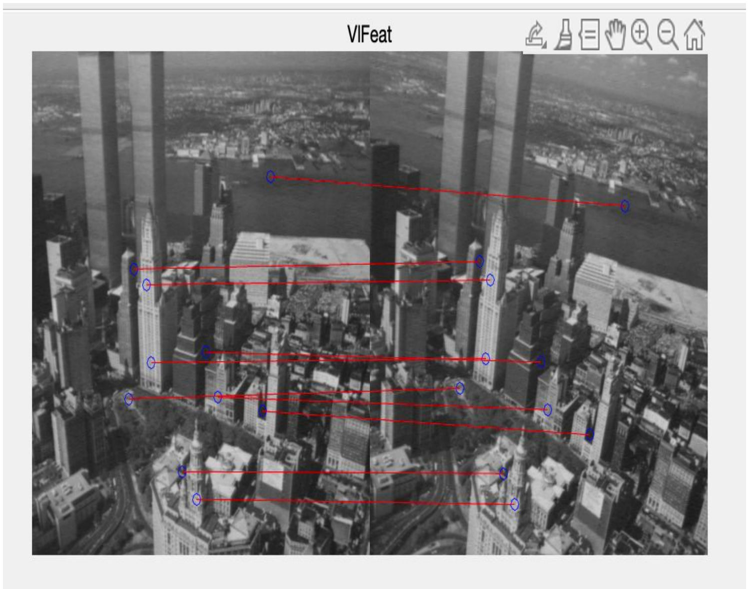
pixels vs SSD



pixels vs NCC







Accuracy for pixels vs SSD: 0.993750

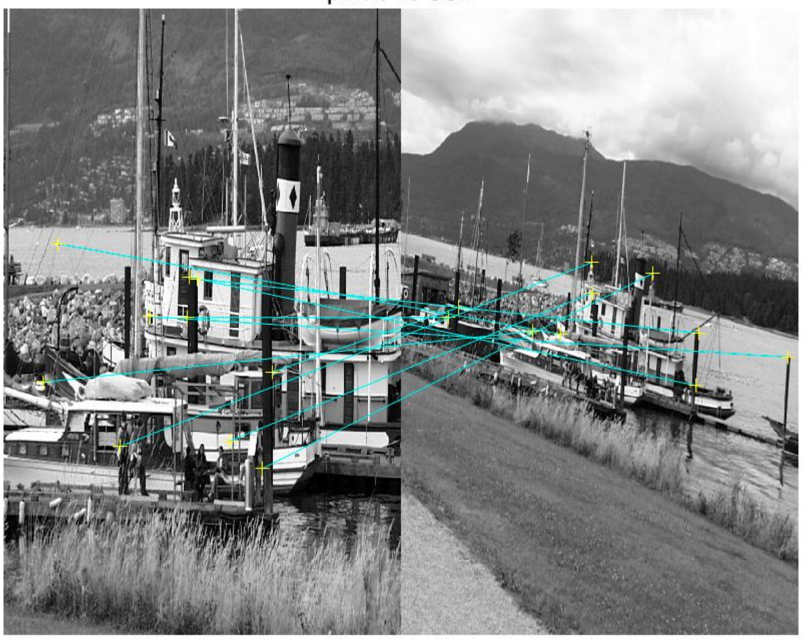
Accuracy for pixels vs NCC: 0.920000

Accuracy for histogram vs Chi-Square: 0.708333

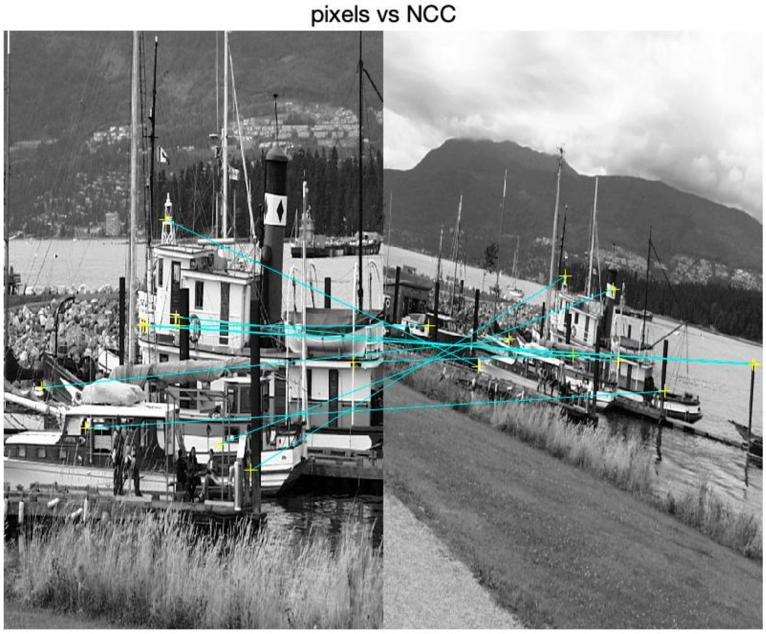
Accuracy for SIFT matches: 1.000000

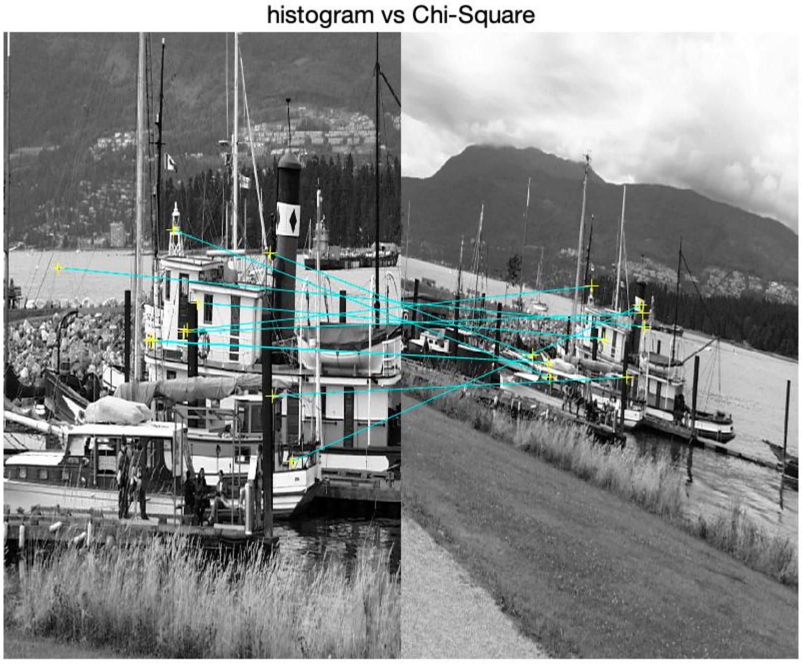
- For the images in scale_rotation_change: Chi-Square performs well. SSD lacks scale and rotation invariance, making it perform poorly under these conditions. NCC is not designed to be invariant to scale and rotation, so it may fail when these changes occur.

pixels vs SSD



pixels vs NCC





Accuracy for pixels vs SSD: 0.650602

Accuracy for pixels vs NCC: 0.610000

Accuracy for histogram vs Chi-Square: 0.805195

Accuracy for SIFT matches: 1.00000

- For the images in view_change: Chi-Square performs well.

histogram vs Chi-Square

VlFeat

Accuracy for pixels vs SSD: 0.625000

Accuracy for pixels vs NCC: 0.390000

Accuracy for histogram vs Chi-Square: 0.739130

Accuracy for SIFT matches: 0.900000

P2:

After employing these two methods, the accuracy increases.

- For the images in illumination_change:

pixels vs SSD

pixels vs NCC

histogram vs Chi-Square

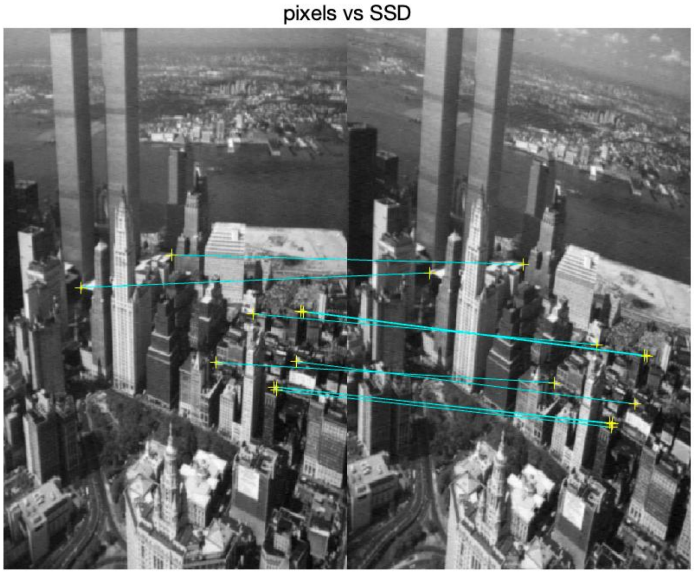
Accuracy for pixels vs SSD: 1.000000

Accuracy for pixels vs NCC: 0.982609

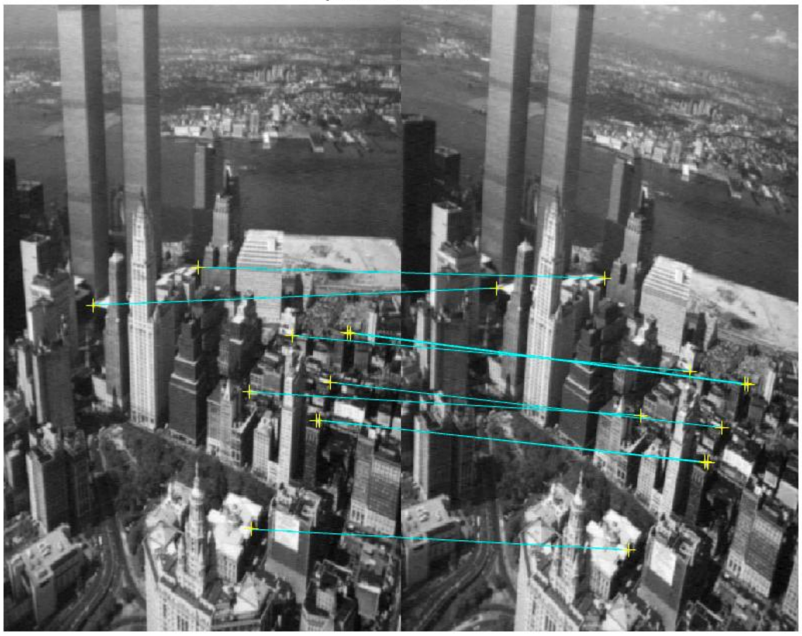
Accuracy for histogram vs Chi-Square: 0.833333

- For the images in planar_rotation_change:

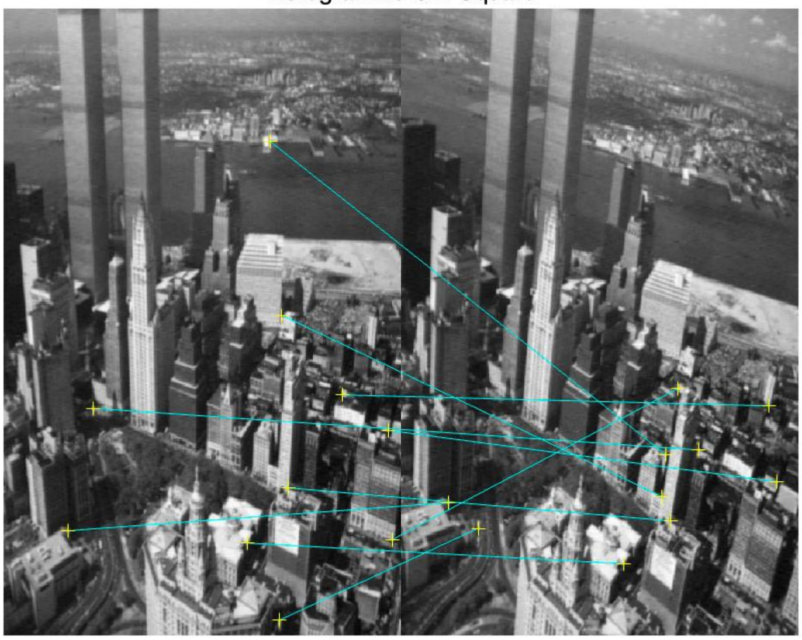
pixels vs SSD



pixels vs NCC



histogram vs Chi-Square

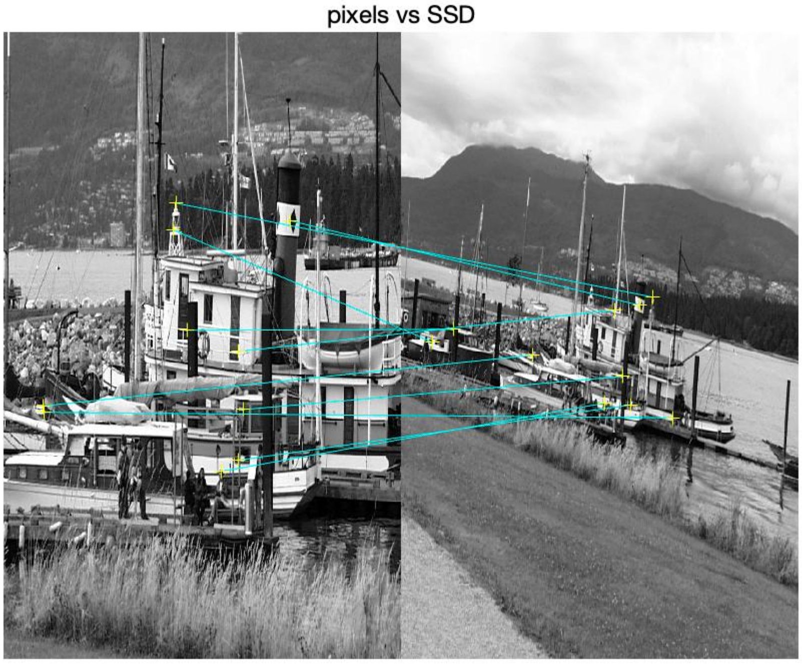


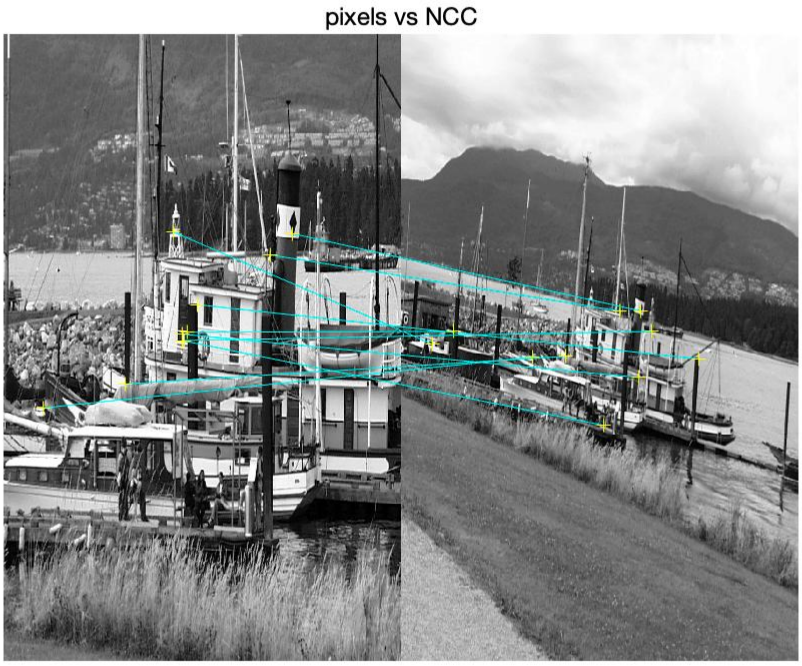
Accuracy for pixels vs SSD: 1.000000

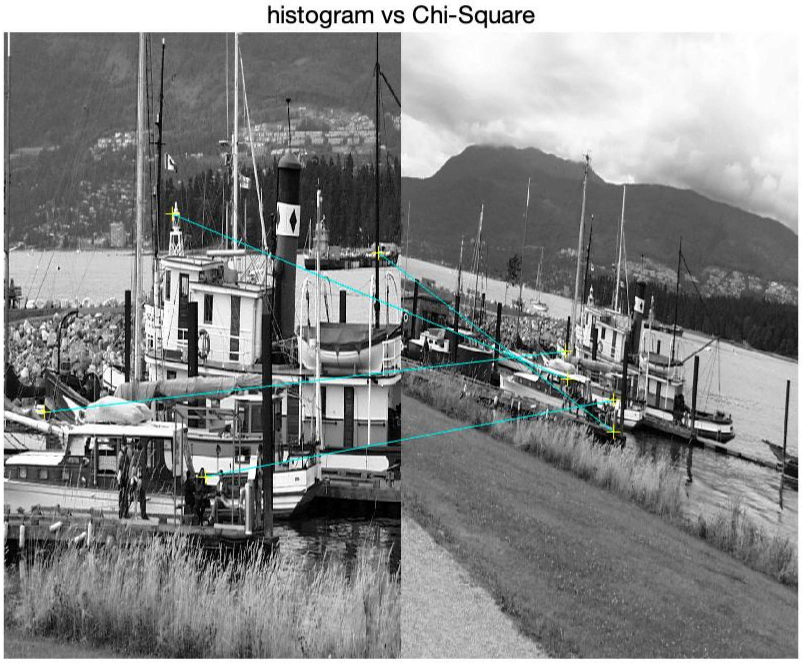
Accuracy for pixels vs NCC: 0.994083

Accuracy for histogram vs Chi-Square: 0.666667

- For the images in scale_rotation_change:





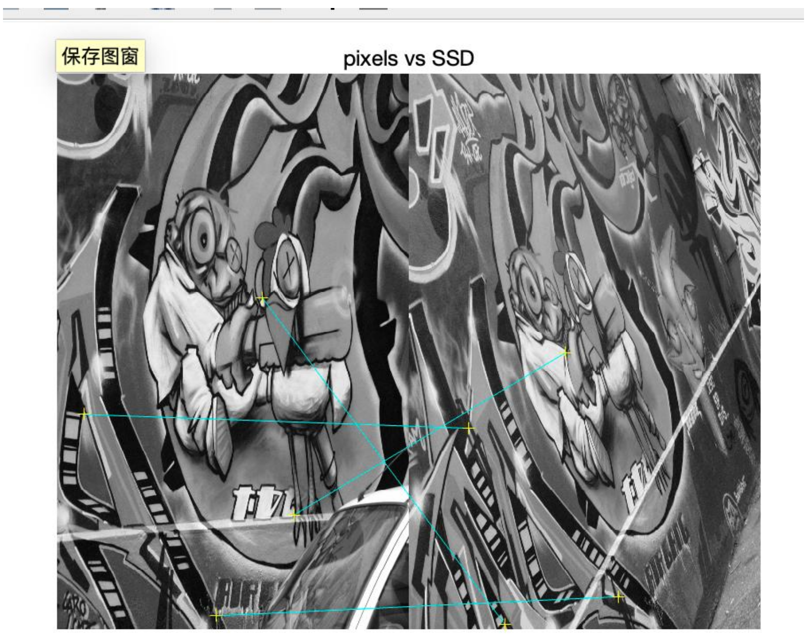


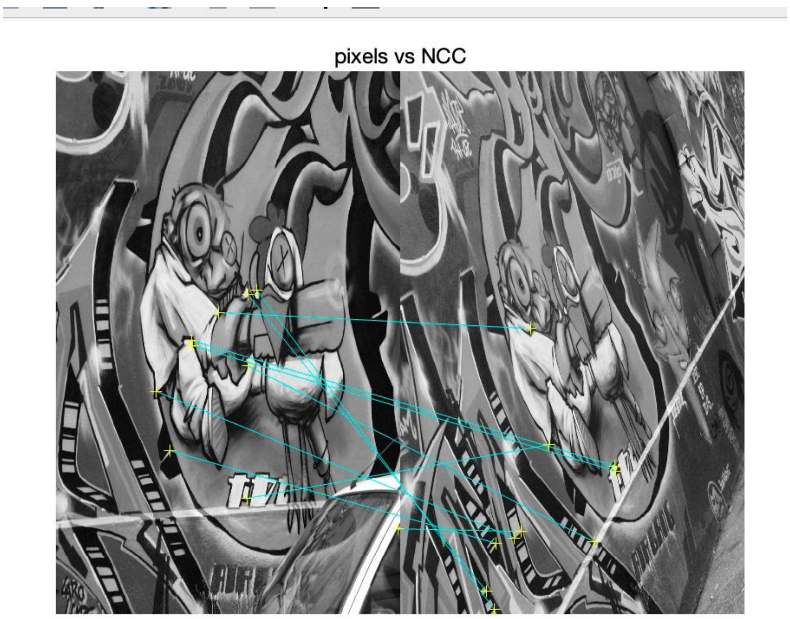
Accuracy for pixels vs SSD: 0.714286

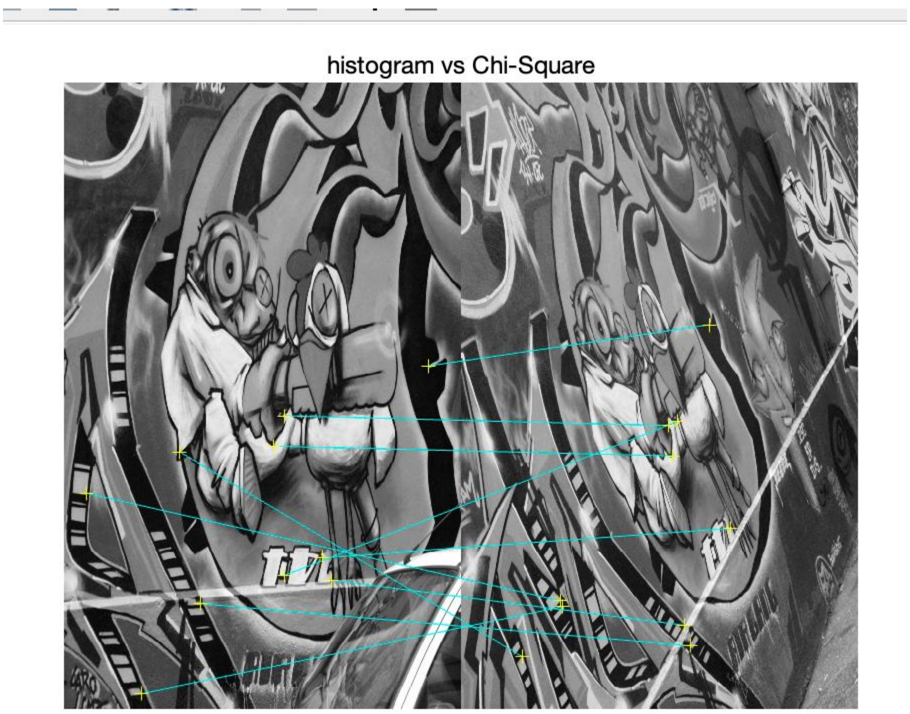
Accuracy for pixels vs NCC: 0.700000

Accuracy for histogram vs Chi-Square: 1.000000

- For the images in view_change:







Accuracy for pixels vs SSD: 0.750000

Accuracy for pixels vs NCC: 0.578947

Accuracy for histogram vs Chi-Square: 0.933333

P3:

How does the BRIEF descriptor describe a feature?

The BRIEF (Binary Robust Independent Elementary Features) descriptor is designed to provide a simpler and faster alternative to the conventional descriptors. Through a series of intensity difference tests, it captures the underlying feature of an image patch in a singular way. In essence, this implies that BRIEF will test pairs of pixels within the patch to determine whether one is brighter than the other rather than relying on precise information like gradients. A string representation of the patch is created using this binary method. It has been demonstrated to tolerate slight degrees of in-plane rotation, but it does not compensate for orientation and is not intrinsically rotationally invariant.

Why is it fast in matching features?

BRIEF offers significant speed advantages, especially during matching. Due to its binary string representation, which only requires calculating the Hamming distance, matching is completed very quickly. Hamming distance calculations for binary strings are computationally far less demanding than approaches that call for computing Euclidean distances in high-dimensional environments. Furthermore, even without special instructions like SSE4.2, BRIEF's method requires less computation time when matching descriptors. In actuality, this means that real-time performance is possible even on machines with constrained computational power.