Matlab Build in function:
CM = cornermetric(imGray);
Matlab build Harris corner detector:
```matlab
cimg = corner(IM);
```
Failed to separate points evenly
Follow the steps on the function specification
Adaptive Non-Maximal Suppression

Keep top 10% points from score matrix

For each pixel calculate the distance from all the points which have a greater score

find the minimum distance

Sort all those distances and pick top 40

When the image is large, and has lots of corners, this non-maximal suppression function could be very slow!!
Approximation

Local window, use the number of the pixels, in that window, have a greater score than the current pixel as a new score. Pick all the pixels have a score 0 as the final feature corners.

No sorting, no distance calculation, only comparison and summation. This method is lightening fast and has almost the same good result. Cannot control the number of the features got from suppression, but only can guarantee smaller or greater than a specified number.

Adaptive Non-Maximal Suppression

40 feature corners

proximation method: more than 40 points
Extract
Feature Descriptors
& Matching

5 by 5 Gaussian kernel
With kernel variance of 5

Mirror the edge for sample

Threshold is 0.5 for descriptors matching
Find the most votes

Use **SVD** to refine the homography
Stitching & Blending

Blending
Find the optimal seam between two image
Gradient Domain blending (1D)

Seamless Image Stitching in the Gradient Domain, Anat Levin, et al
Mosaics of Scenes with Moving Objects, James Davis, et al
Stitching & Blending
Stitching & Blending
Stitch 6 815×454 Images in 12 seconds
On an i5 Machine
Lessons

1. Vectorization or for loop
   A trade off of time and memory

2. Approximation
   Wait less, 30 seconds for 2 image VS. 12 second for 6 images