

**CSCI 677**

**HW3 Report**

**SIFT Features and Object Matching via RANSAC for Homography  
Estimation**

**Dhruv Parikh**

## Preliminaries

The code has been sectionalized into several functions –

- i. function for SIFT
- ii. function for brute-force matching
- iii. function for refining matches via ratio test
- iv. function for yielding top matches using distance
- v. function for homography (via RANSAC), inlier matches
- vi. several functions for visualization

The codes are self-explanatory and their explanations can be found in the Python Notebook submitted.

For brute-force matching with KNN to refine matches, KNN neighbor value that has been used is 2, however the code is written for an arbitrary value of the KNN neighbors used for matching. Irrespective, however, of the  $k$  value selected, the final refined, valid, good features for a single source to destination match, has been obtained via the simple ratio test – and thus using  $k = 2$  makes the most sense.

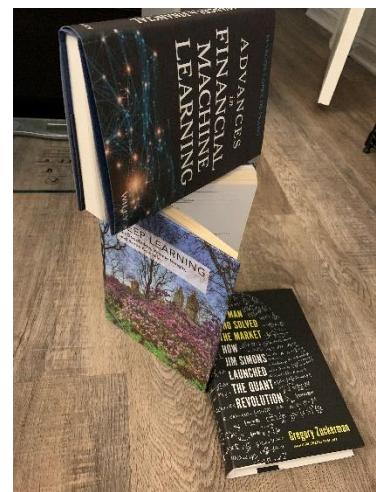
These single source to destination matches, or good matches, are then used for computing homography matrix estimate (via RANSAC) and the corresponding inlier and outliers.

The top 20 matches prior to homography and the top 10 inlier matches post homography have also been obtained and used for visualization.

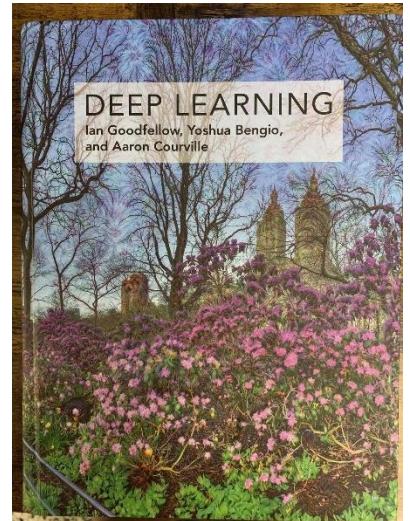
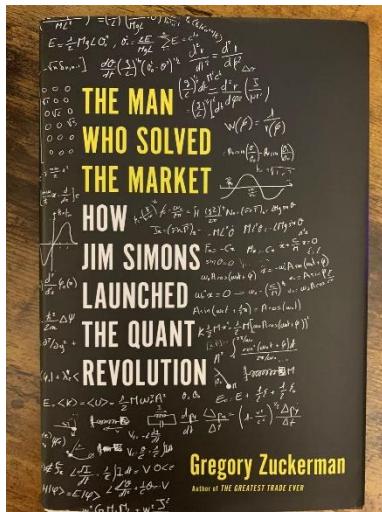
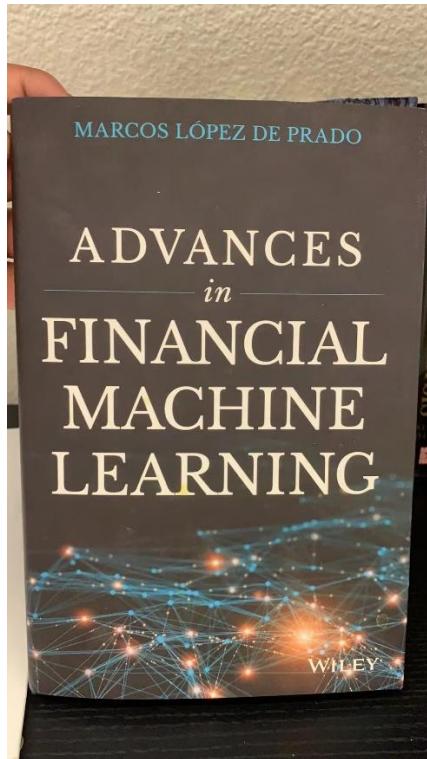
## Source and Destination Images



Destination – Flat



Destination – Tall



Source – Market

Source – Deep

Source – Finance

As seen in these images and their corresponding labels, for ease of usage the sources and destinations have been remained using their generic image characteristics.

### Parameter Values

As stated before,  $k$  was taken 2. The threshold number of matches to decide whether a source object was found or not in the destination image was 10 matches (10 good matches, obviously). Thus, according to this threshold, all objects were found in the destinations. Additionally, the inlier threshold for deciding on a match being an inlier or outlier match, via the L2 error, was set at 5.0. Thus, if projected source points and destination match points have a disparity more than this, than they are termed as outliers, and if this disparity is contained within the threshold than they are termed to be inliers. Other parameters that are relevant can be found in the code.

### Comment on Result Folders

A result folder corresponds to a source-destination pair (corresponding to the new names given). Within this folder there are the following items –

1. SIFT features visualized for source and destination.
2. All matches between source and destination (for some  $k$  selected, this will correspond to all  $k$  matches of each source point visualized in the destination image with its corresponding matches).
3. Good matches – we filter out these  $k$  matches for each source point using the ratio test – we also simultaneously filter weak matches. This gives us a match set containing a single match

for each source and only for matches that aren't weak according to the ratio test. Ratio constant was 0.75.

4. Top 20 matches – we further filter the good matches to yield the top 20 matches prior to homography, again for visualization.

5. Inlier matches – the good matches are filtered using homography into inliers and outliers. The inlier matches are visualized.

6. Top 10 inlier matches – the inlier matches are further refined to select the top 10 matches with the least projection error. Again, these are visualized.

7. Source projection into destination to find the bounding box using a perspective transform of the source edges via the homography matrix so computed. This final step gives us the corresponding object match boundaries.

Thus, each folder contains a total of **8** images (2 SIFTs, 3 pre-homography visualizations, 3 post-homography visualizations).

Each folder also contains a **.txt file** which contains the following information –

Source and Destination tags, total SIFT features, total matches, total inliers and outliers, flag that shows if a match is found or not and the homography matrix computed from RANSAC.

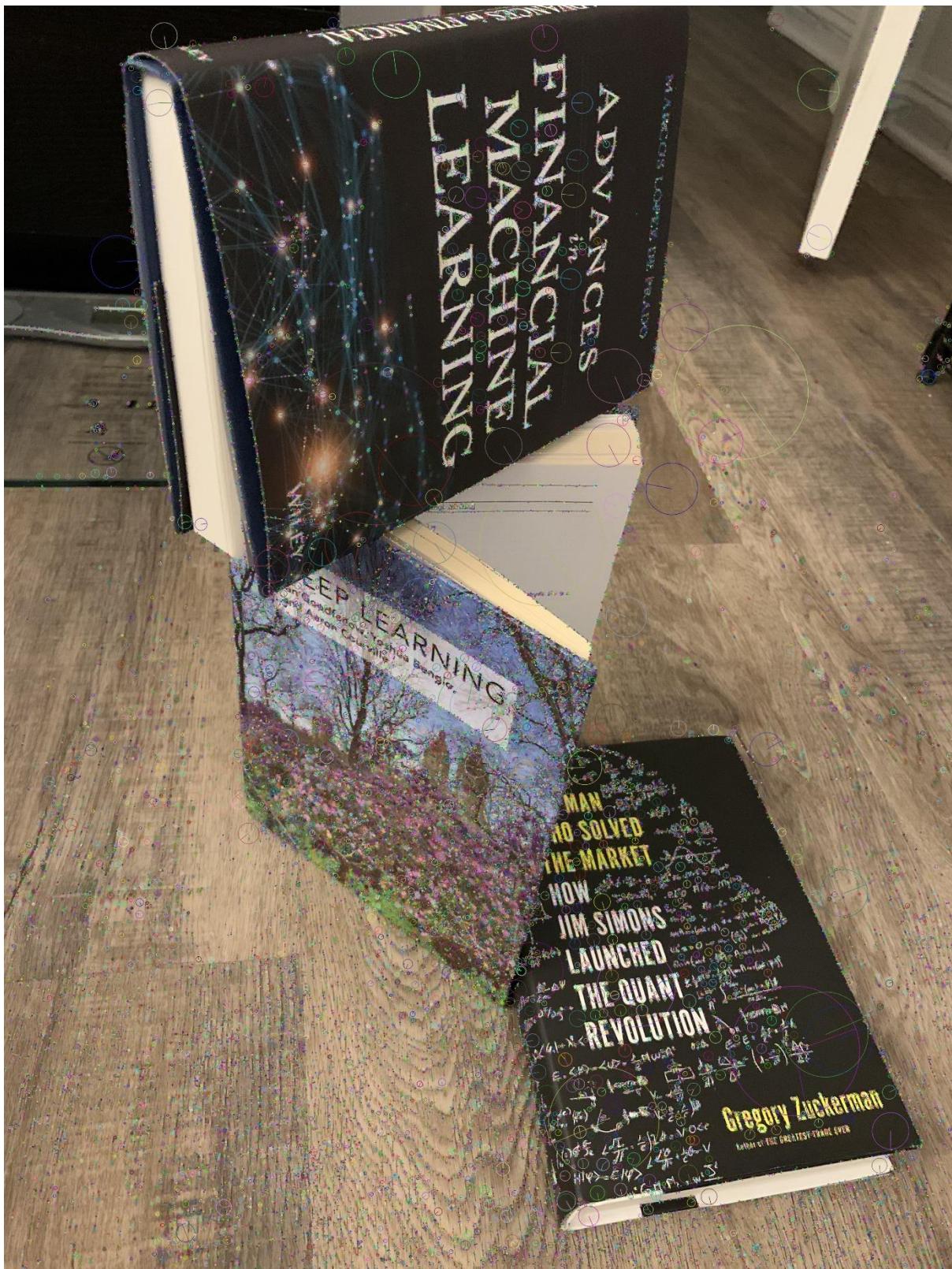
### **Result Summary**

Again, the detailed results can be seen in the folders, here most important results (and asked results) will be summarized.

### **Destination SIFTs**

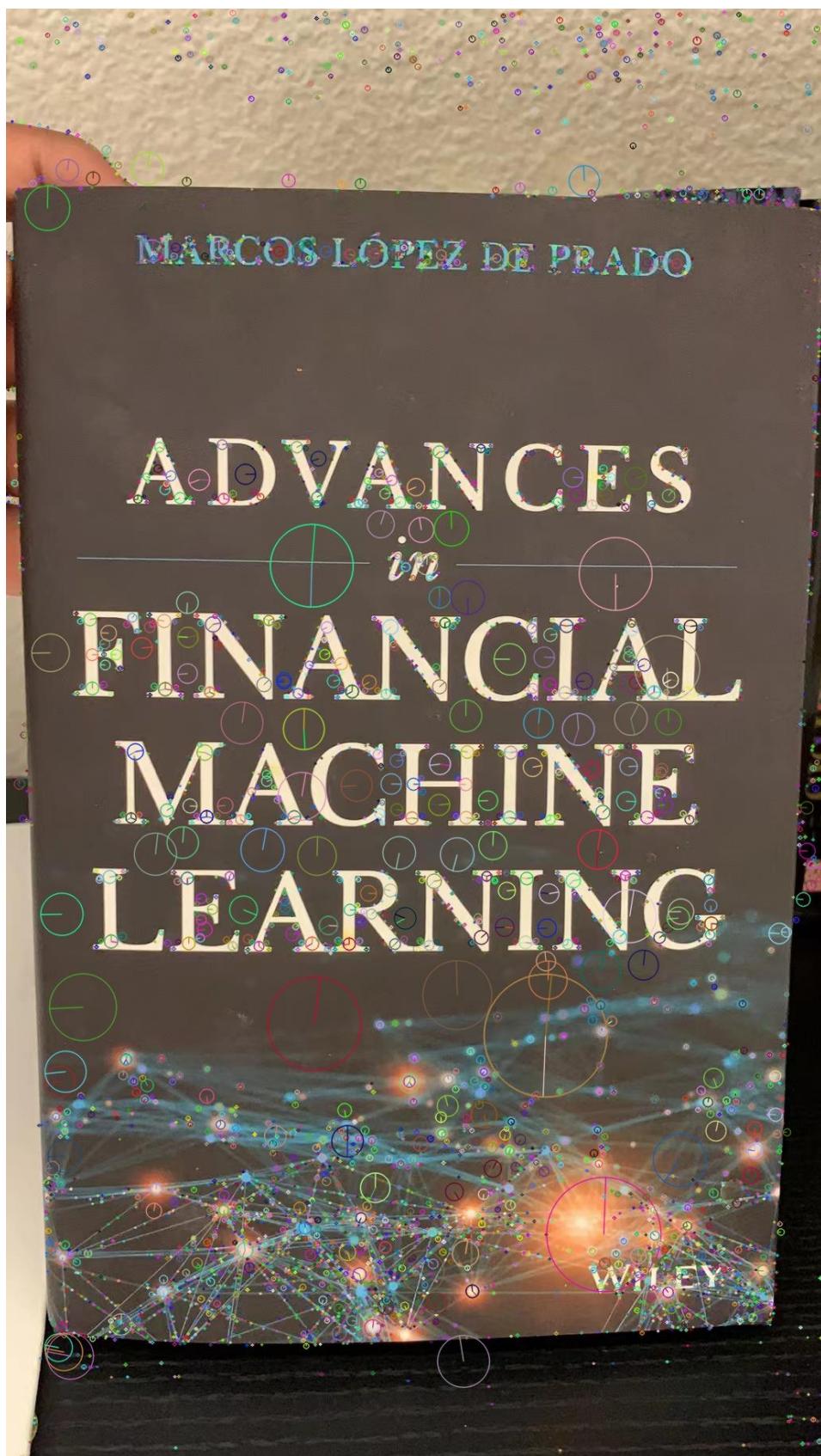


(SIFT for flat)

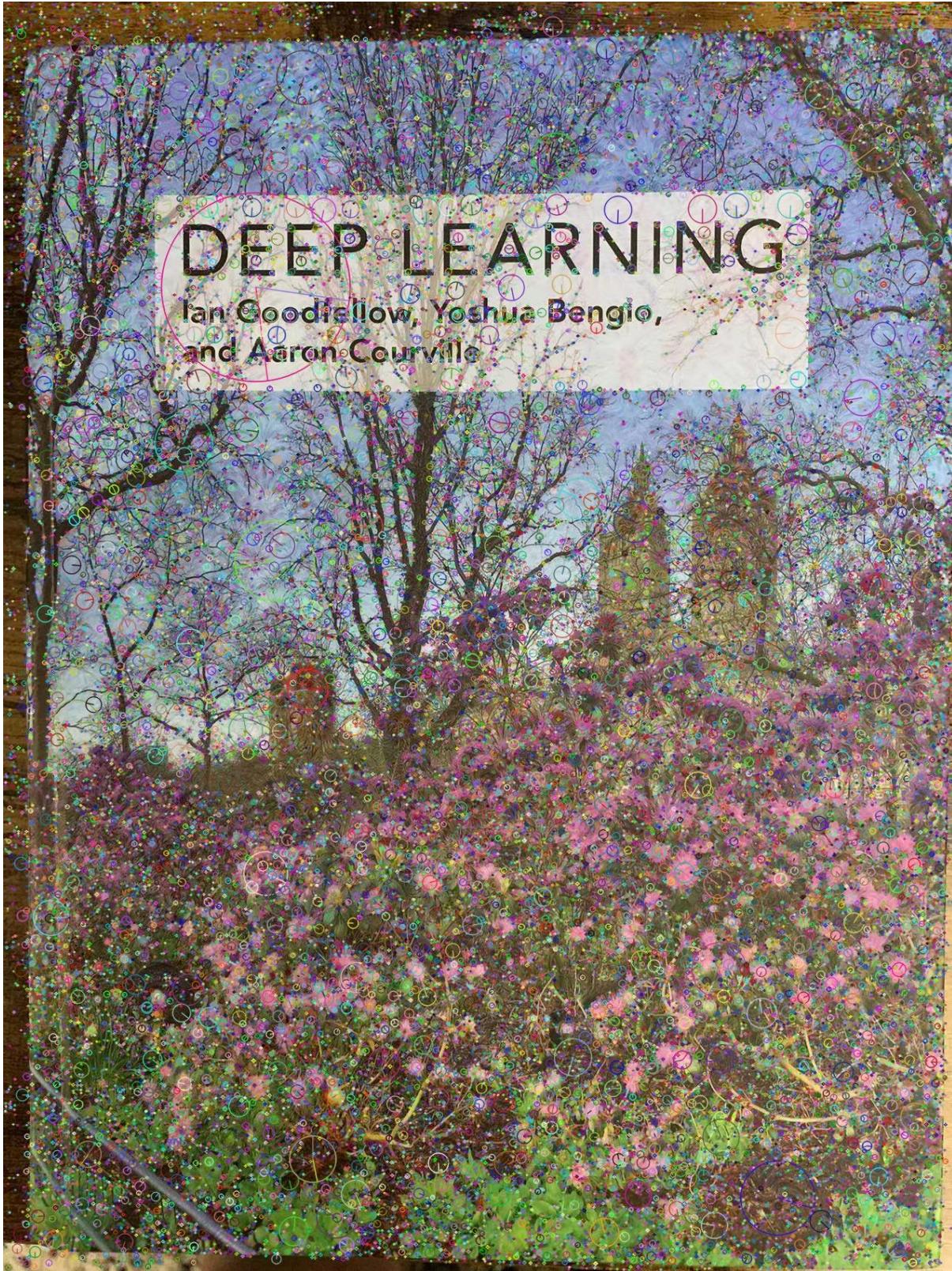


(SIFT for tall)

Source SIFTs



(SIFT for finance)



(SIFT for deep)

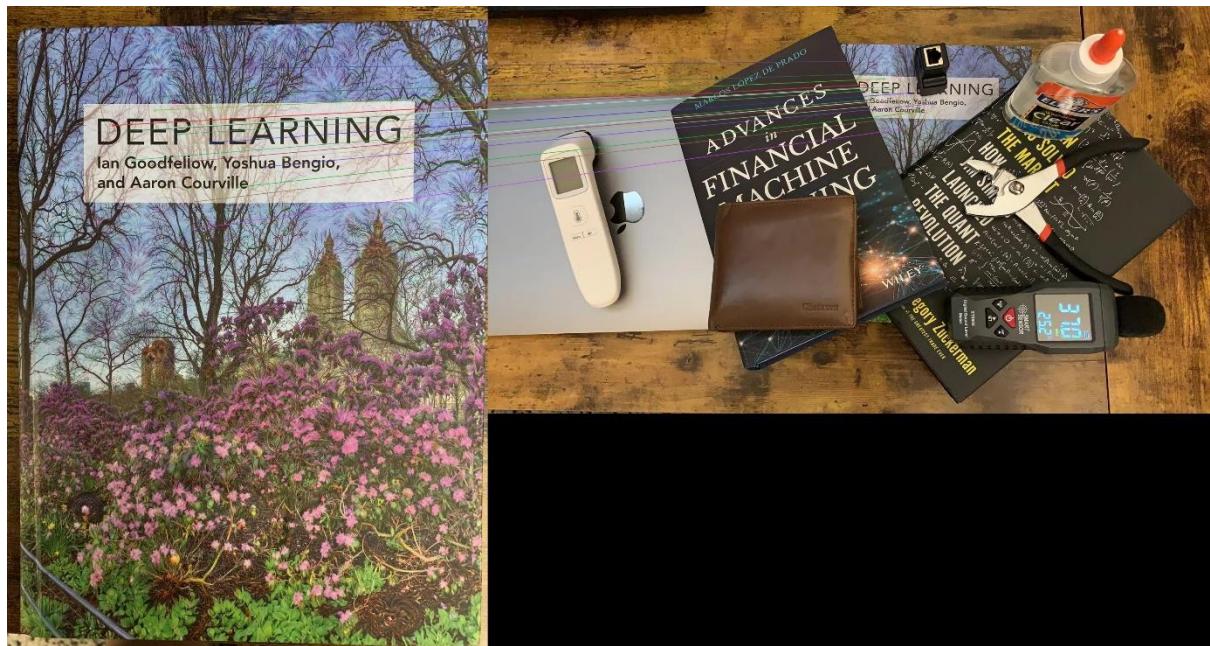
# THE MAN WHO SOLVED THE MARKET HOW JIM SIMONS LAUNCHED THE QUANT REVOLUTION

Gregory Zuckerman

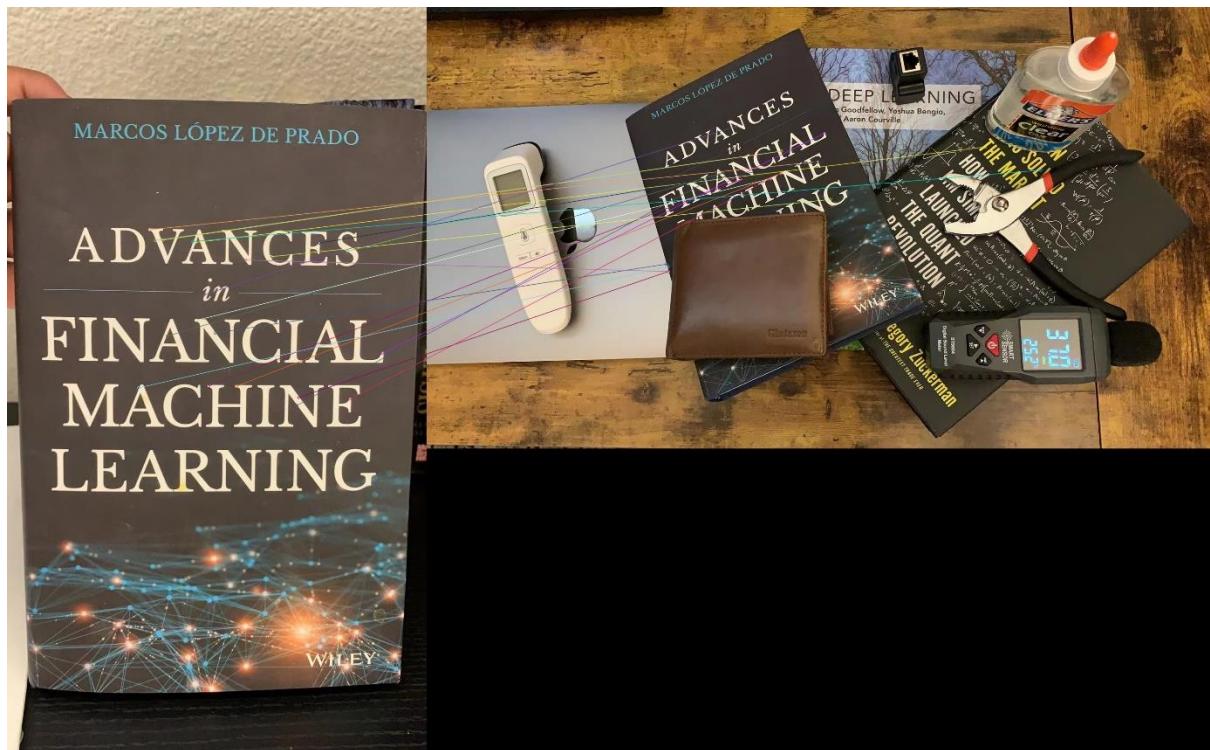
AUTHOR OF THE GREATEST TRADE EVER

(SIFT for market)

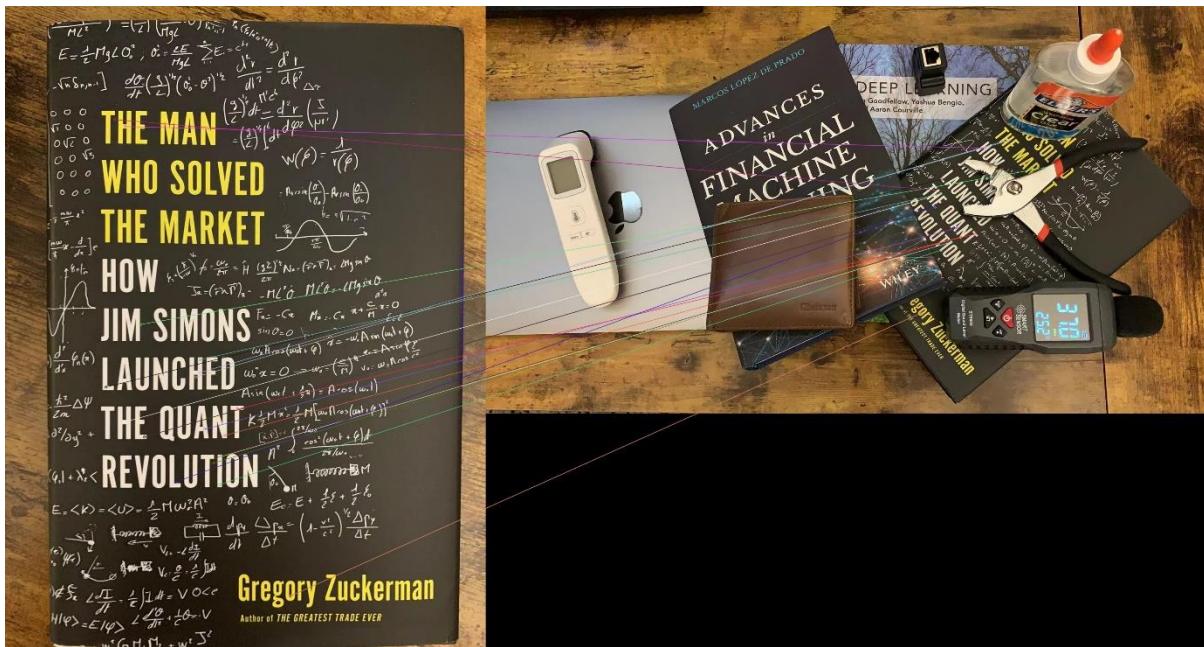
## Top 20 matches prior to homography



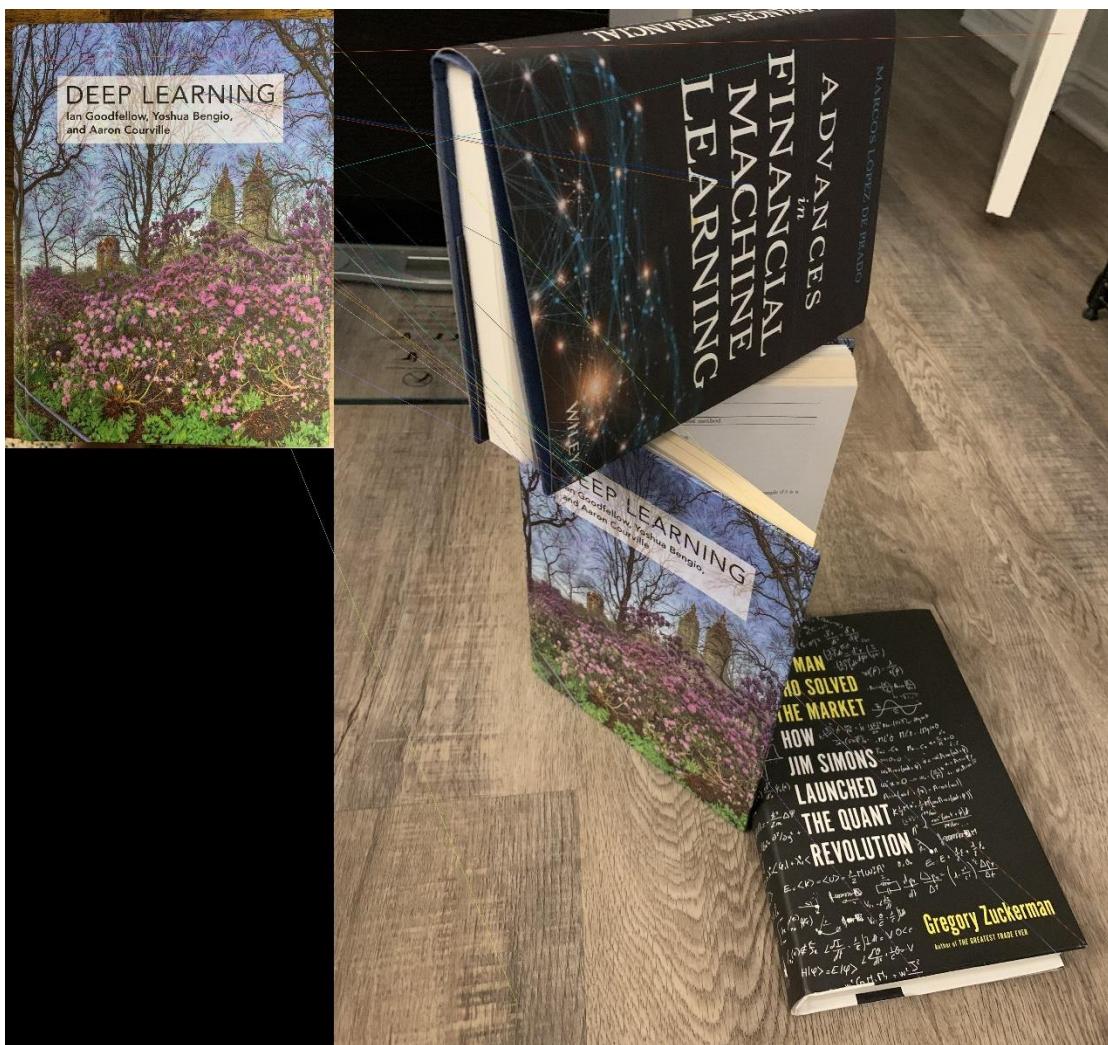
(deep-flat)



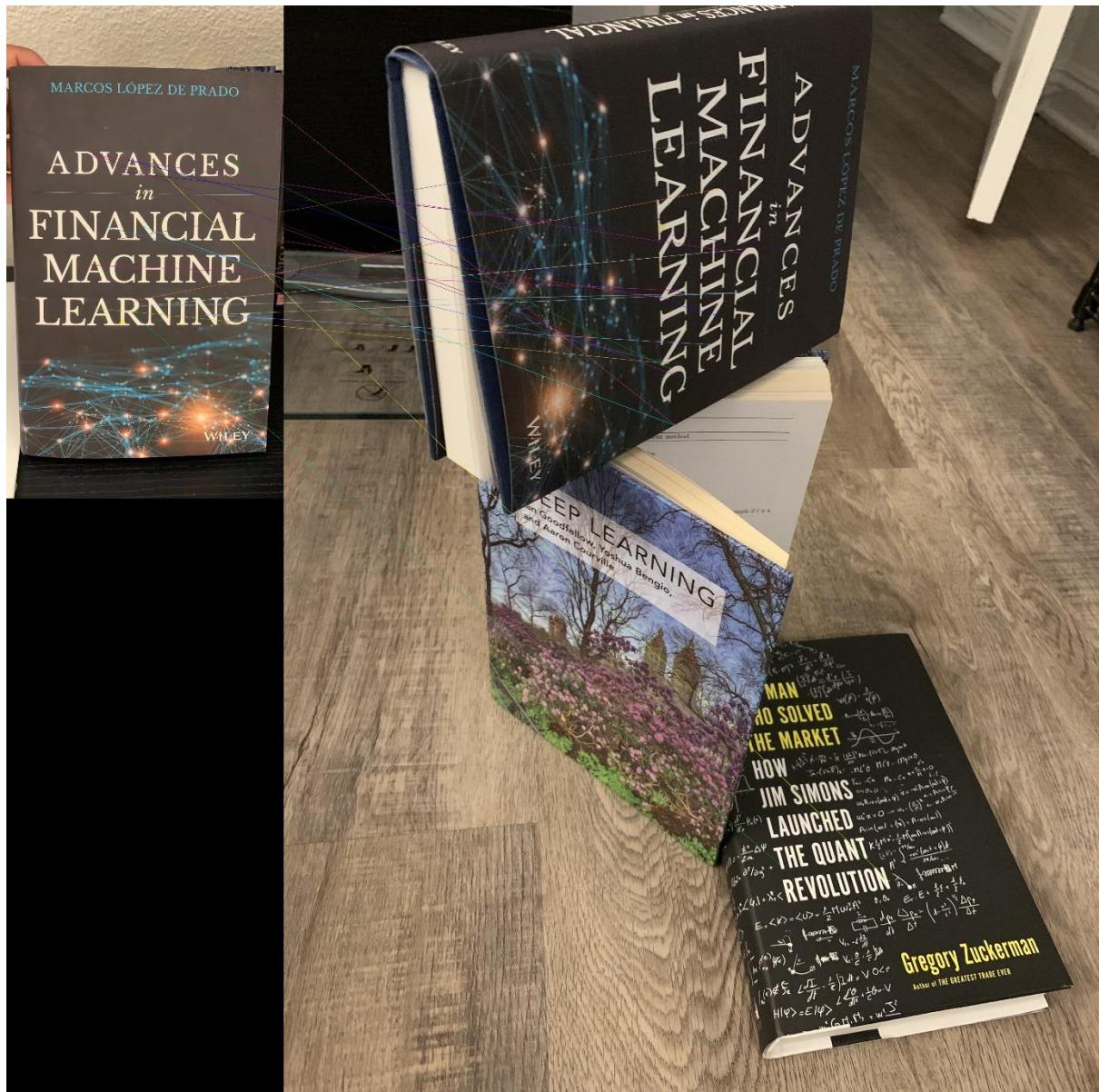
(finance-flat)



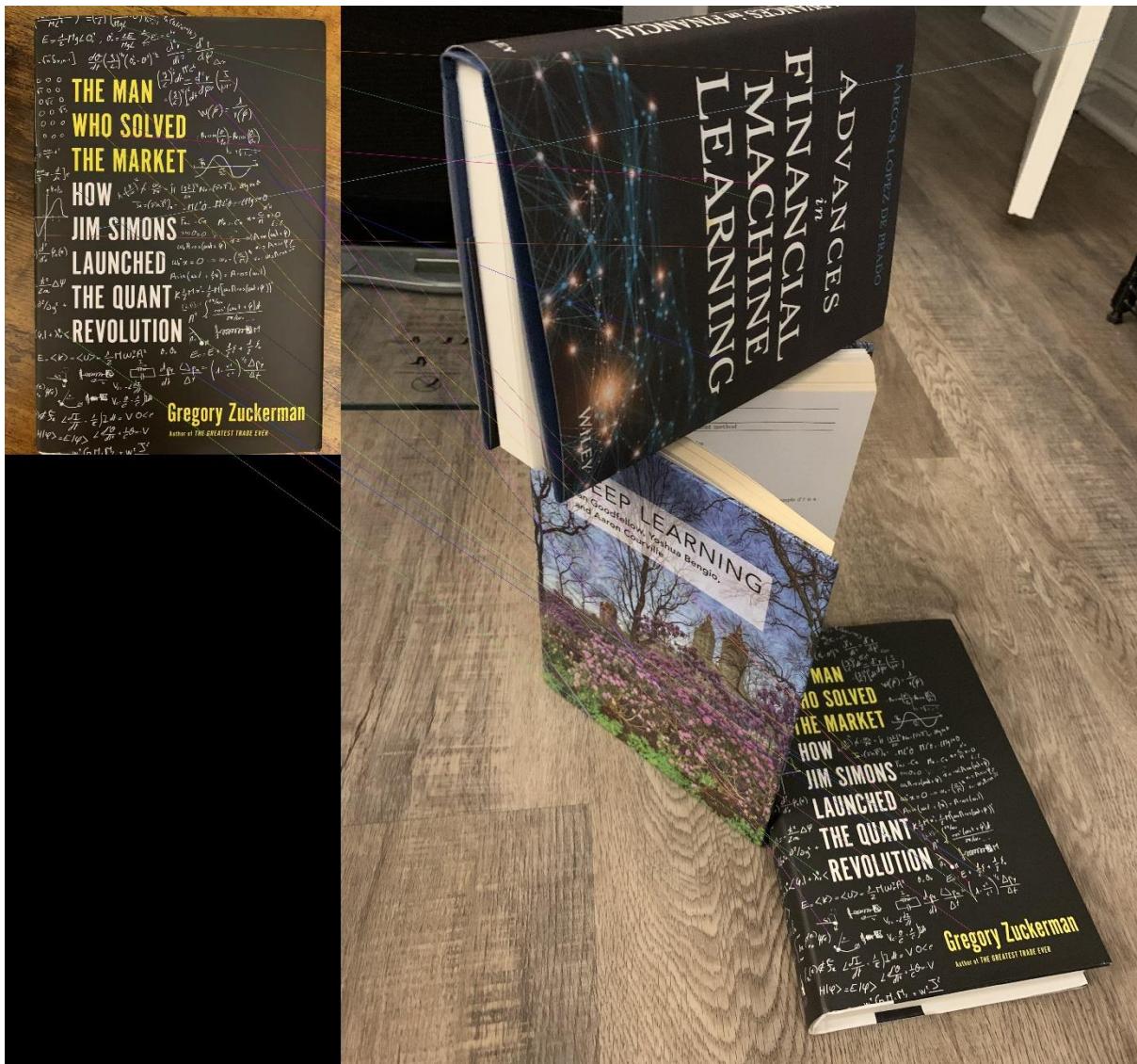
(market-flat)



(deep-tall)



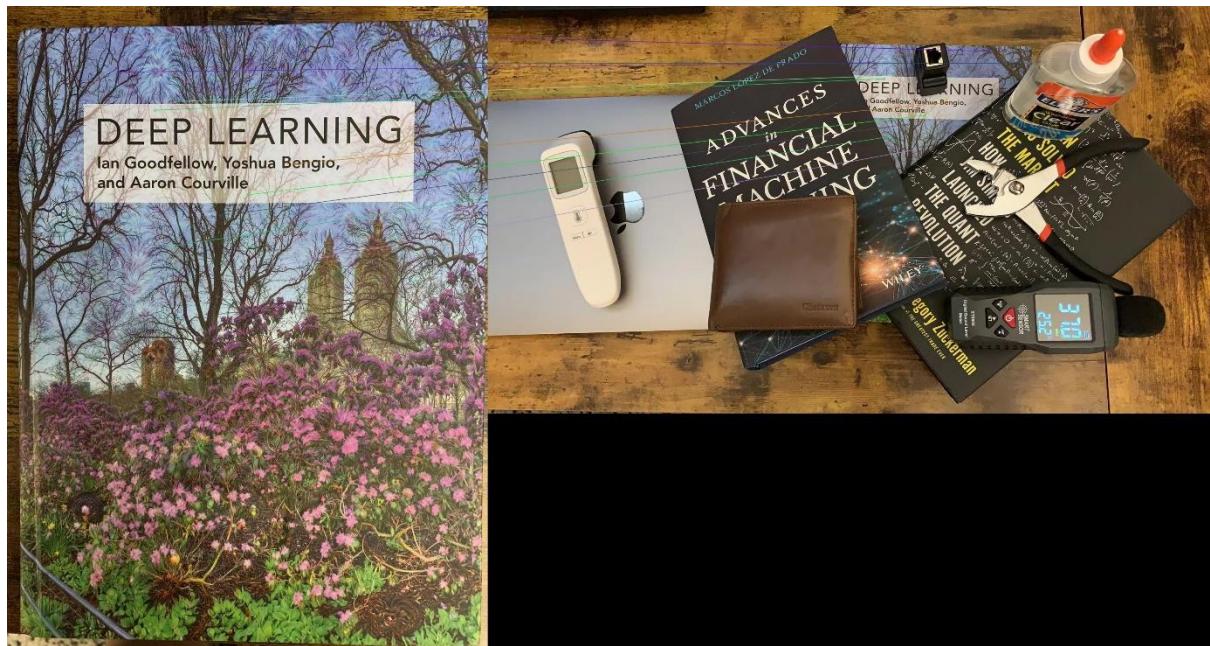
(finance-tall)



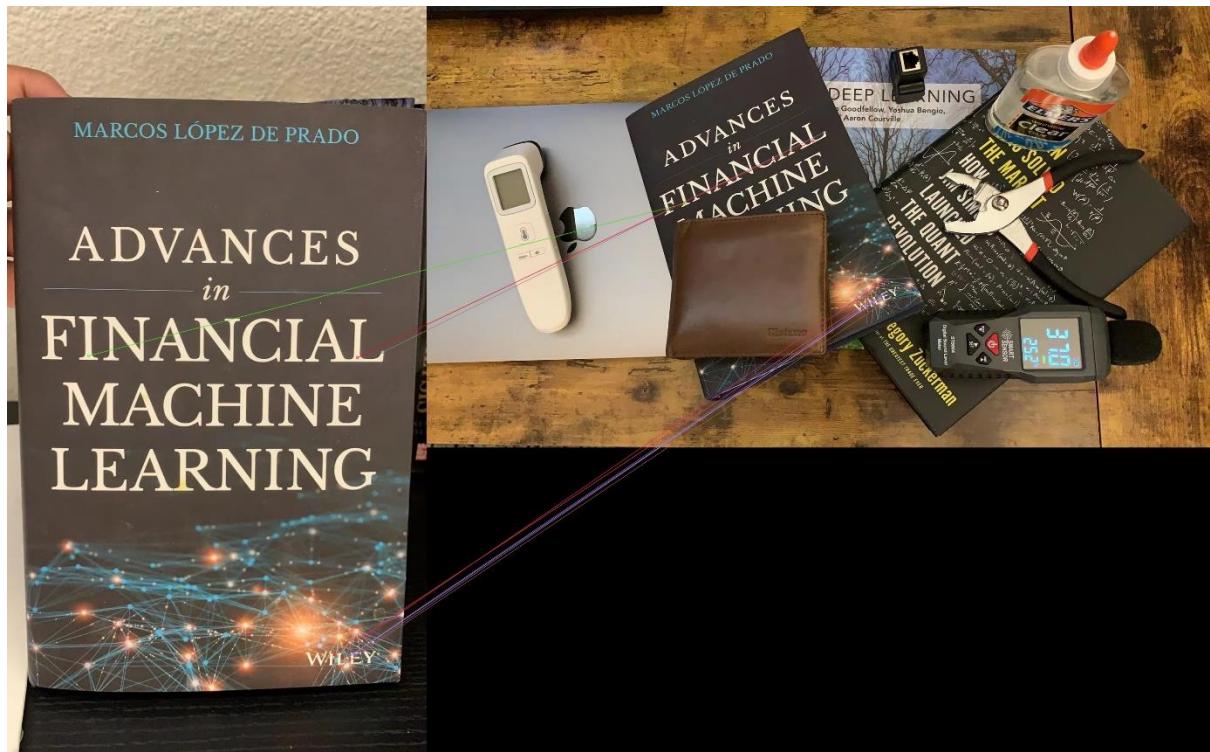
### (market-tall)

Note that if good matches were less than 20 then all good matches are displayed as top matches.

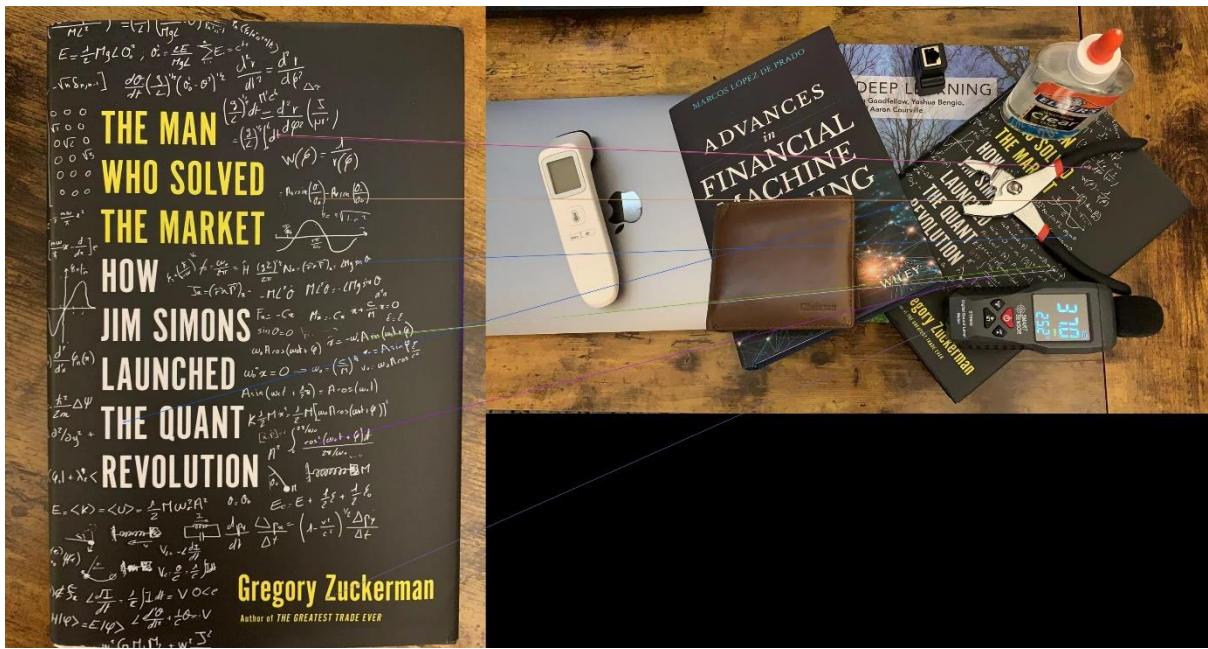
## Top 10 inlier matches post homography



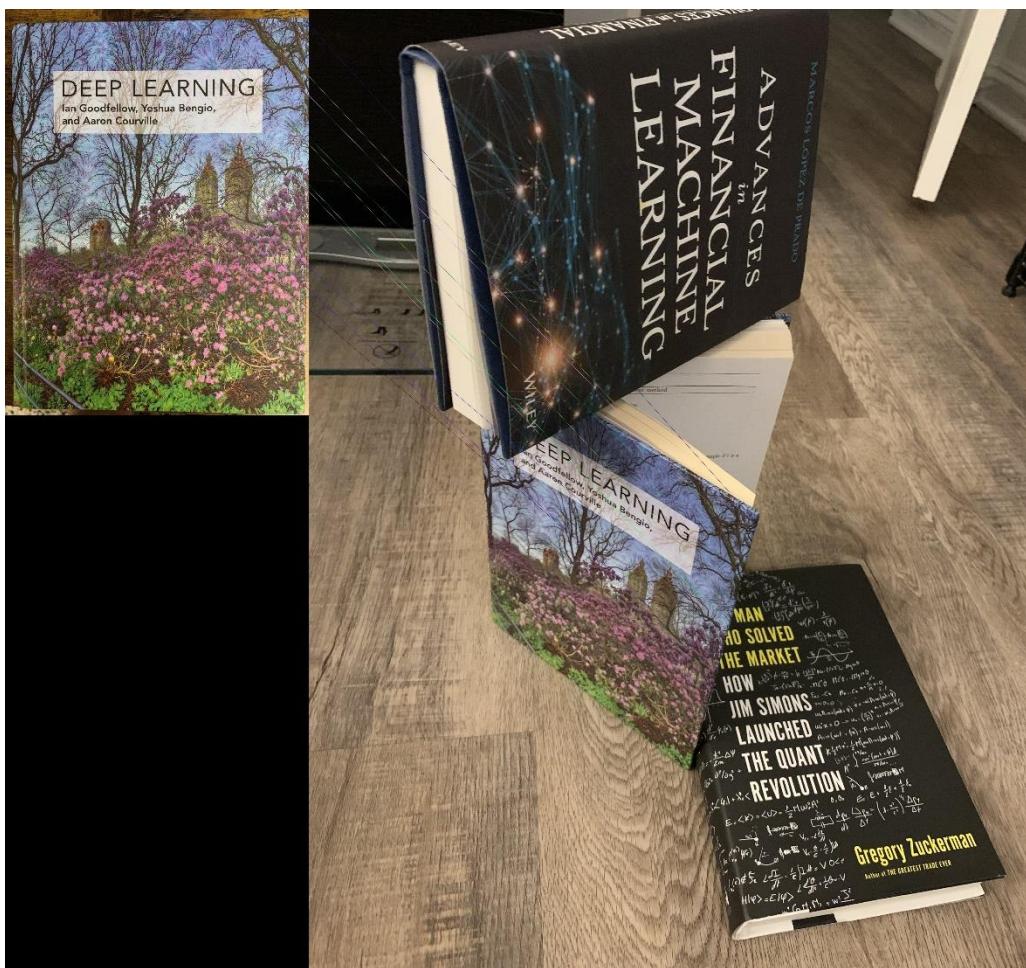
(deep-flat)



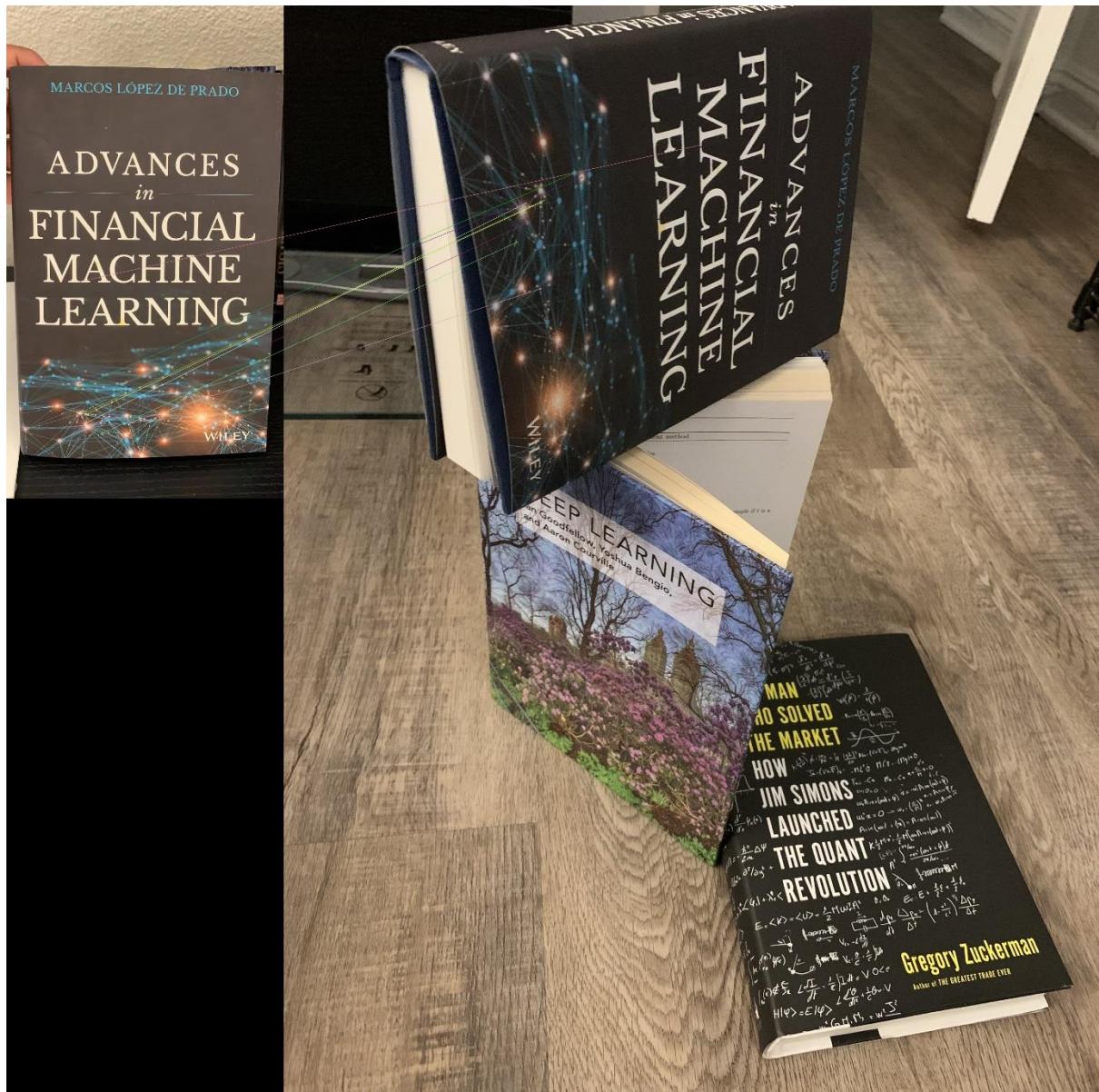
(finance-flat)



(market-flat)



(deep-tall)



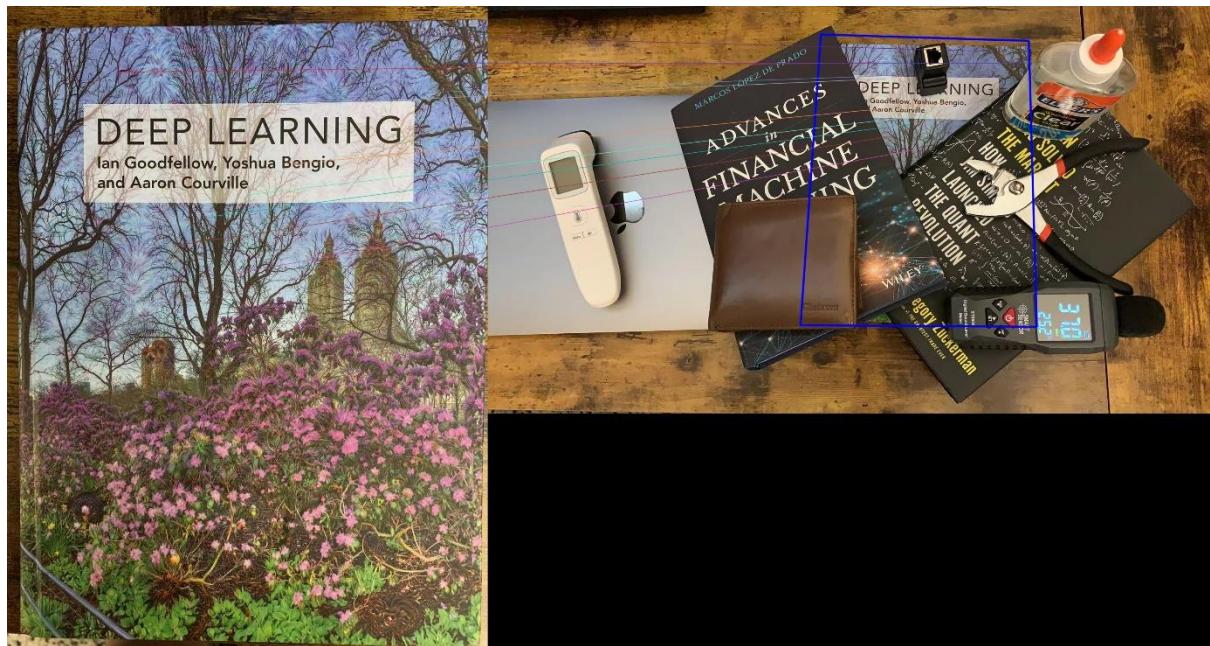
(finance-tall)



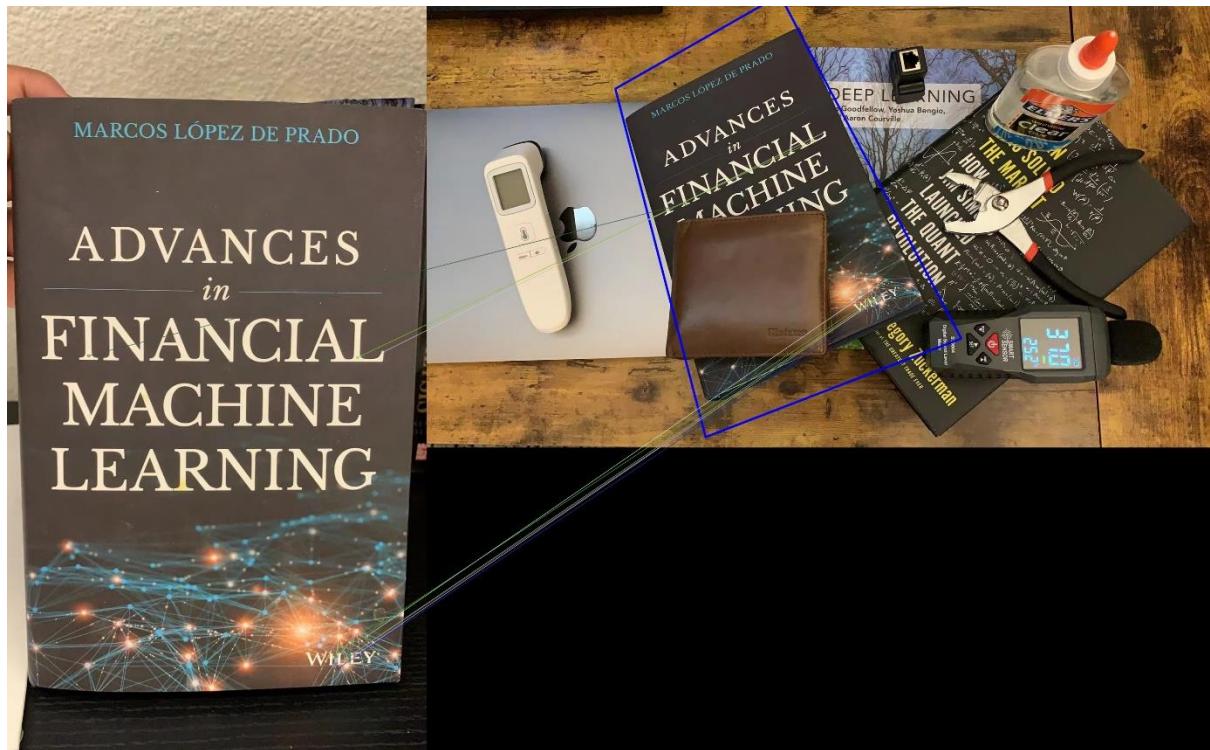
### (market-tall)

Note that if total inlier matches are less than 10 then all inlier matches have been displayed.

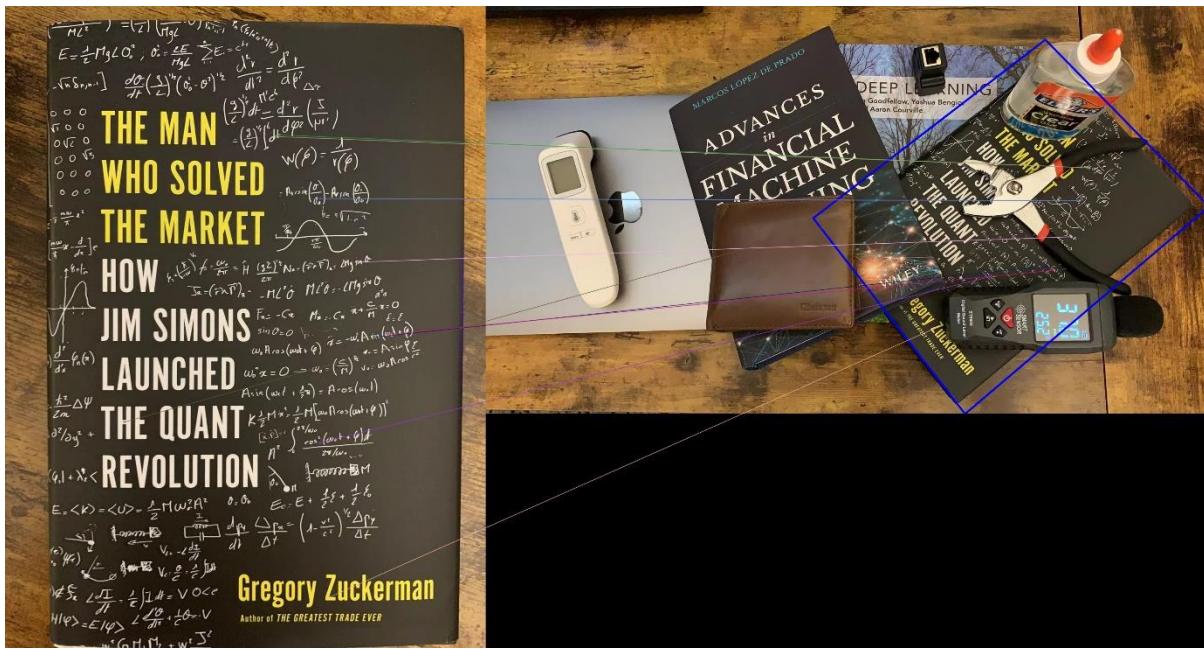
## Projected Source Bounding Box in Destination (found object visualization)



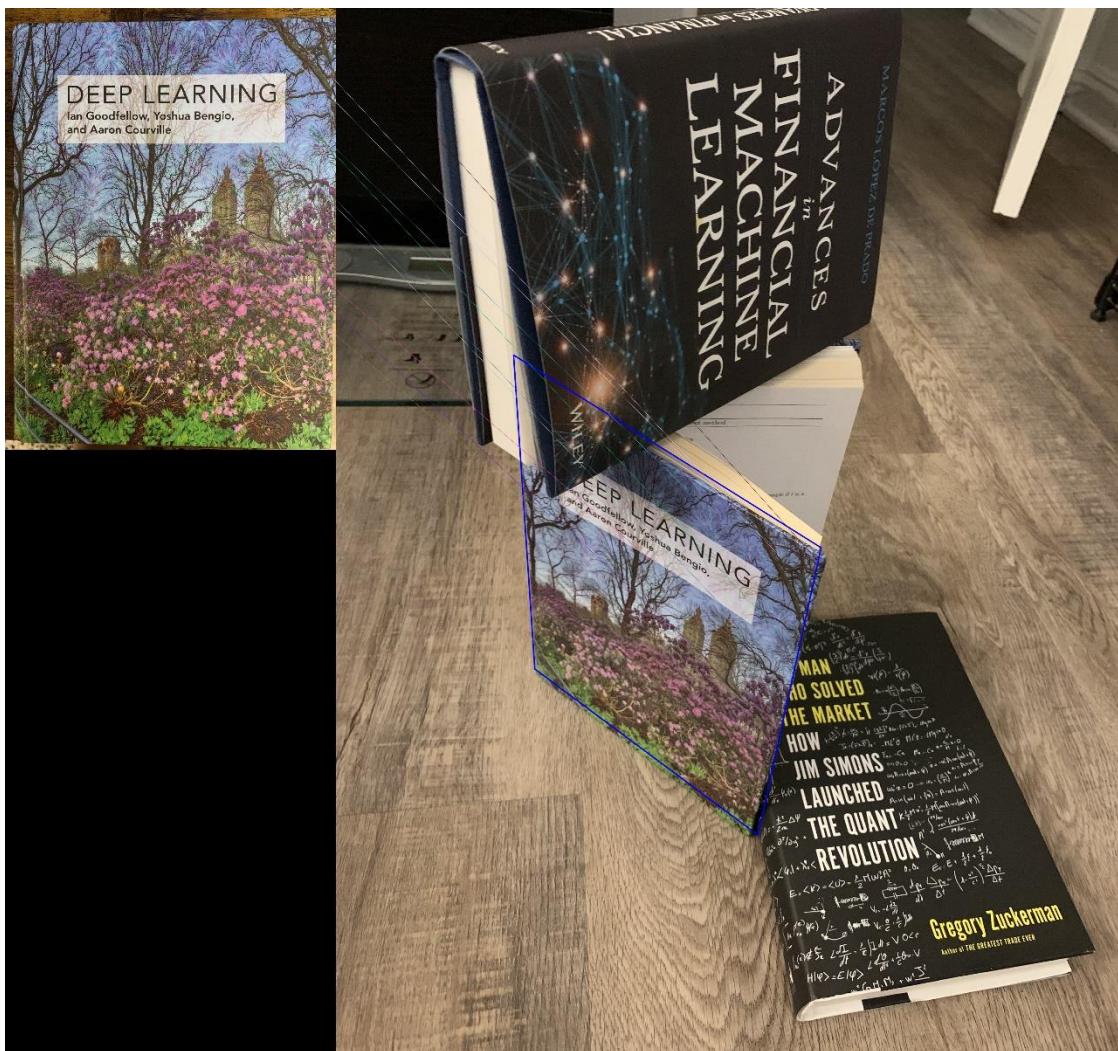
(deep-flat)



(finance-flat)



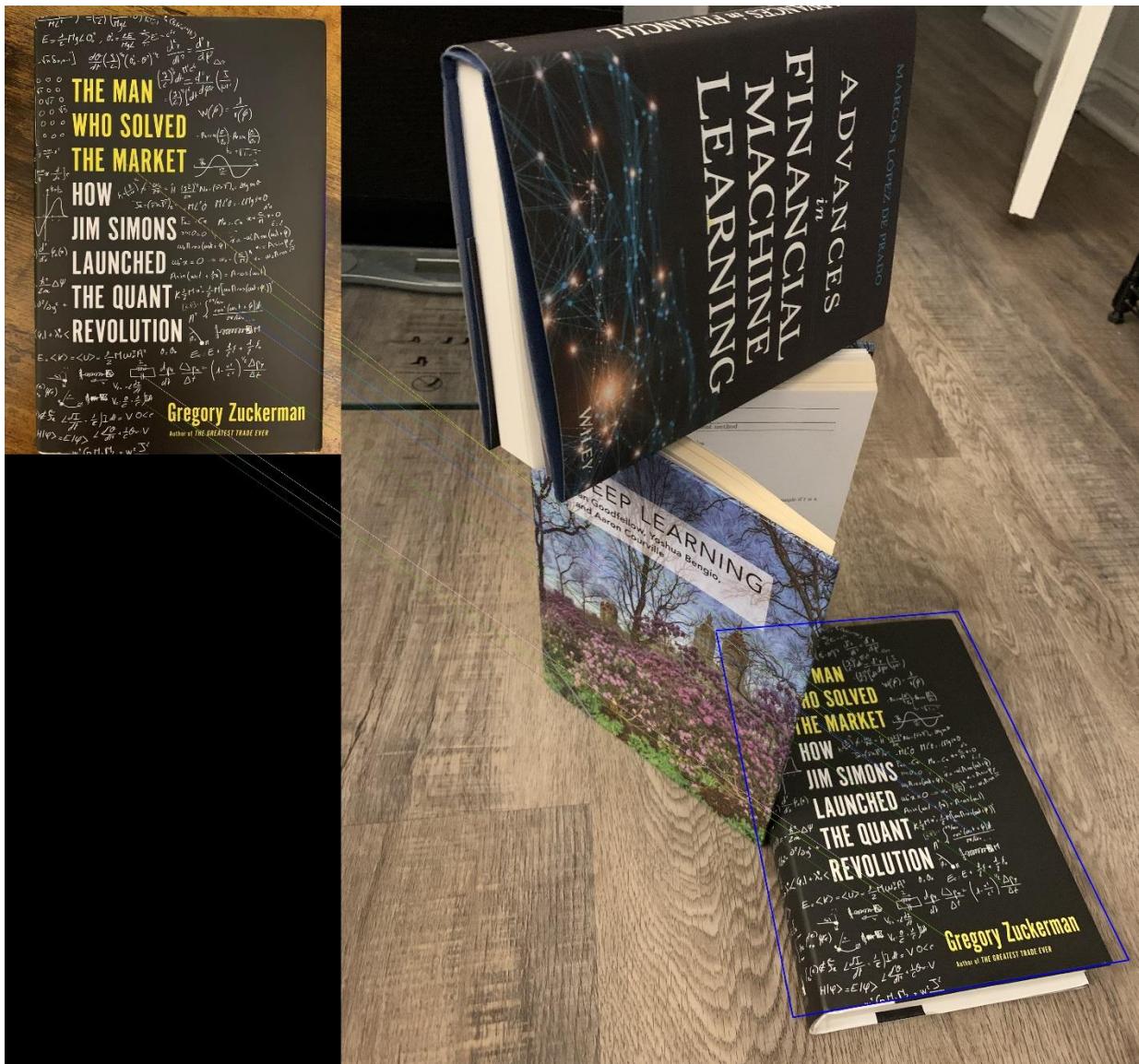
(market-flat)



(deep-tall)



(finance-tall)



(market-tall)

### Result Statistics (from .txt files)

Image	Total SIFT Features
Flat (destination)	10625
Tall (destination)	49979
Finance (source)	5259
Market (source)	13004
Deep (source)	42457

Image Pair	Total Matches	Total Inlier Matches
Deep + Flat	606	426
Deep + Tall	155	23
Finance + Flat	529	244
Finance + Tall	219	43
Market + Flat	1047	401
Market + Tall	1241	679

Additional statistics have been described in each .txt files.

## **Homography Matrix**

Homography matrices are outputted in the Notebook submitted as well as available in the .txt files in each image pair folder – to reduce clutter I will not re-display them here.

## **Result Comments**

It can be clearly seen from the projected boxes that the method works really well in obtaining the object region in the destination images. The projections are good estimates for where the objects are in the original image – notably even when the objects are occluded.

Source images whose edges have been transformed provide inconsequential bounding results when the source images contain uncropped exteriors/spillages – such as in the **finance** book where the image hasn't been cropped well so a chunk that isn't in the destination has been transformed for representation incorrectly. This can be avoided by cropping the objects appropriately or by perspectively transforming only that subsection of the object image that is ‘valid’ – this valid region can be determined relatively easily (manually or via segmenting the objects into truer boxes).

The inlier matches for the **tall** destination are low for two source images – the **book** face laying open in them is a possible reason for that – these open books are absent from the object images that only contain the image covers. As such, the covers are identified relatively accurately despite that – however this obviously categorically defeats the planar assumption in homography – something that is much stronger in the **flat** destination category.

**Market + Tall** have a lot of matches – tall and market have large number of SIFT features and are actually representing a closely planar object – which is a possible reason for the large inliers.

The results are quite accurate in terms of the bounding boxes irrespective of occlusion or the depth image nature – the covers are pretty much planar and thus the cover portions of the book are valid candidates for planar homography match – something that is evident from these results.

Using a more rigorous matching approach (to and fro matching from both source and destination images) could make the method more accurate in terms of homography estimation however they could also lead to fewer match points available with RANSAC to compute H.

Again, note that H is displayed in the .txt files and the Notebook.