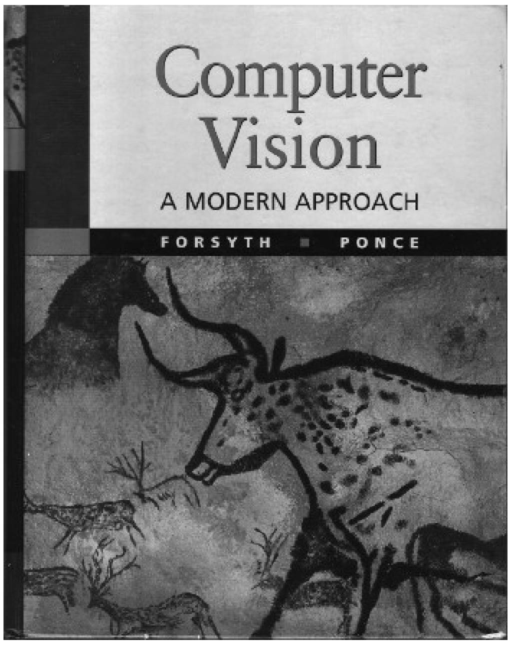
CV Assignment 2

Name 1: Abdelrahman Salem Mohamed
ID 1: 6309
Name 2: Reem Abdelhalim
ID 2: 6114

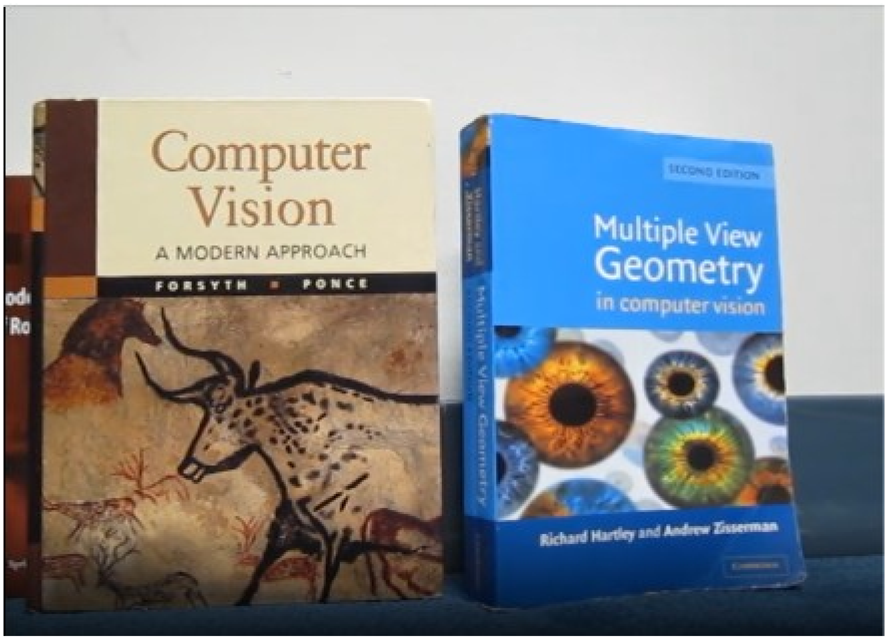
1- Part I: Augmented Reality with Planar Homographies

1.1 Preparations

We want to map video of a cartoon movie to another video containing book, the book cover we have here:



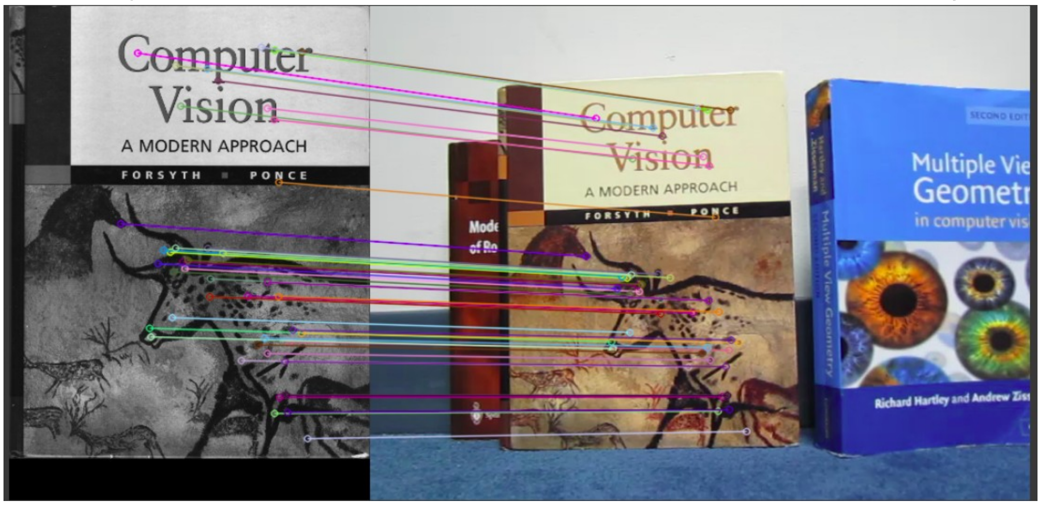
And the video containing the book:



So for every frame of the video we want to replace the cartoon video frames with this book in the video.

1.2 Getting Correspondences

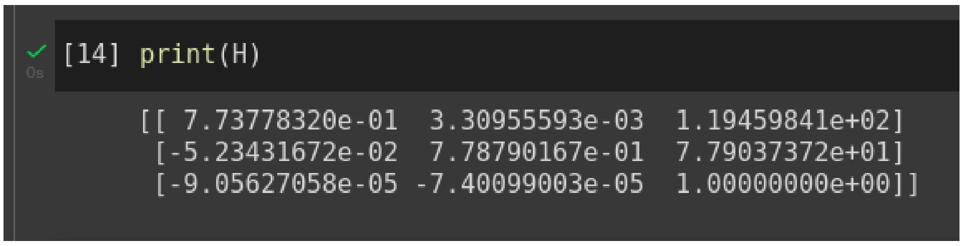
To match two images together first we need to get some features within each image and describe each feature so we can compare between each features from the two images to get the best matching, here we used the sift descriptor to find the keypoints (image features) then compare the features using KNN method to get the best 2 matches for each feature, we can get too many features so we selected the best k matches where k is user input



1.3 Compute the homography parameters

To represent the transformations of the pixel locations from one image to another we need to compute the homography matrix H where it is represented by 3x3 matrix containing 8 parameters to find, since **each correspondence match can be used to identify 2 parameters**, therefore we need **at least 4 different correspondences points**

The H matrix we computed:

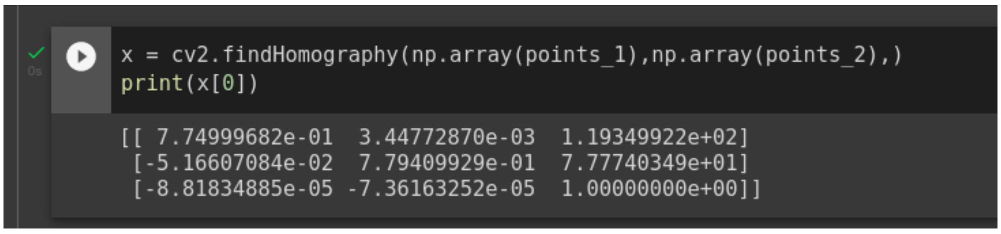


```
[14] print(H)

[[ 7.73778320e-01  3.30955593e-03  1.19459841e+02]
 [-5.23431672e-02  7.78790167e-01  7.79037372e+01]
 [-9.05627058e-05 -7.40099003e-05  1.00000000e+00]]
```

To verify we used the built in open cv function to compute the homography matrix

cv2.findHomography()

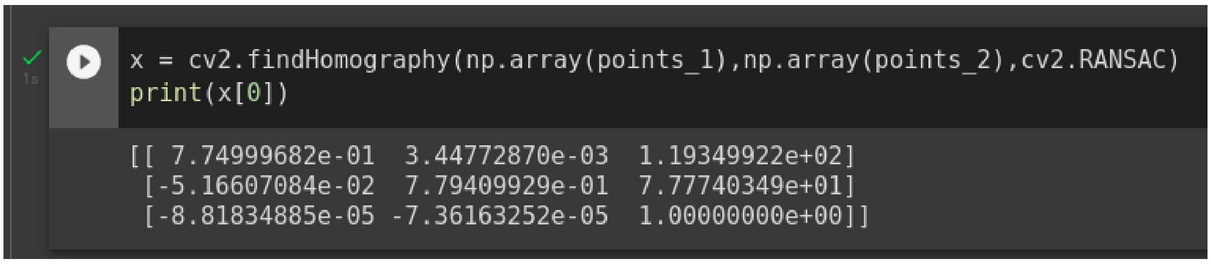


```
x = cv2.findHomography(np.array(points_1),np.array(points_2),)
print(x[0])

[[ 7.74999682e-01  3.44772870e-03  1.19349922e+02]
 [-5.16607084e-02  7.79409929e-01  7.77740349e+01]
 [-8.81834885e-05 -7.36163252e-05  1.00000000e+00]]
```

Also open cv function have option to **get more accurate matrix** by remove the outliers (noise) by applying **RANSAC Algorithm**

For the first frame with the cover image, the H matrix **almost the same** with and without using RANSAC



```
x = cv2.findHomography(np.array(points_1),np.array(points_2),cv2.RANSAC)
print(x[0])
```

```
[[ 7.74999682e-01  3.44772870e-03  1.19349922e+02]
 [-5.16607084e-02  7.79409929e-01  7.77740349e+01]
 [-8.81834885e-05 -7.36163252e-05  1.00000000e+00]]
```

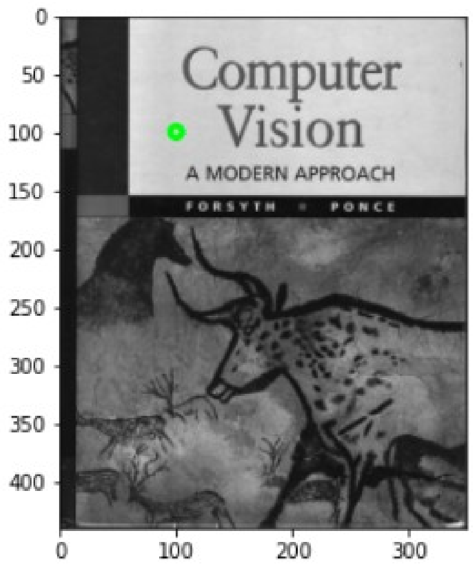
1.4 Testing the homography matrix

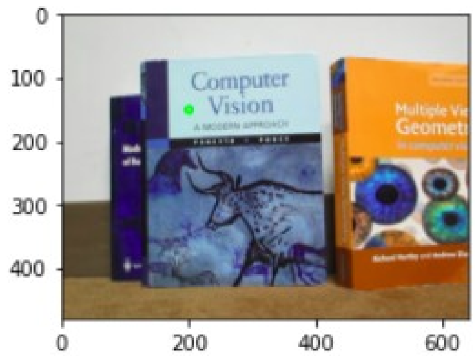
To apply the transformation on points (x,y) from one image to get the mapping into another image we need to compute the dot product

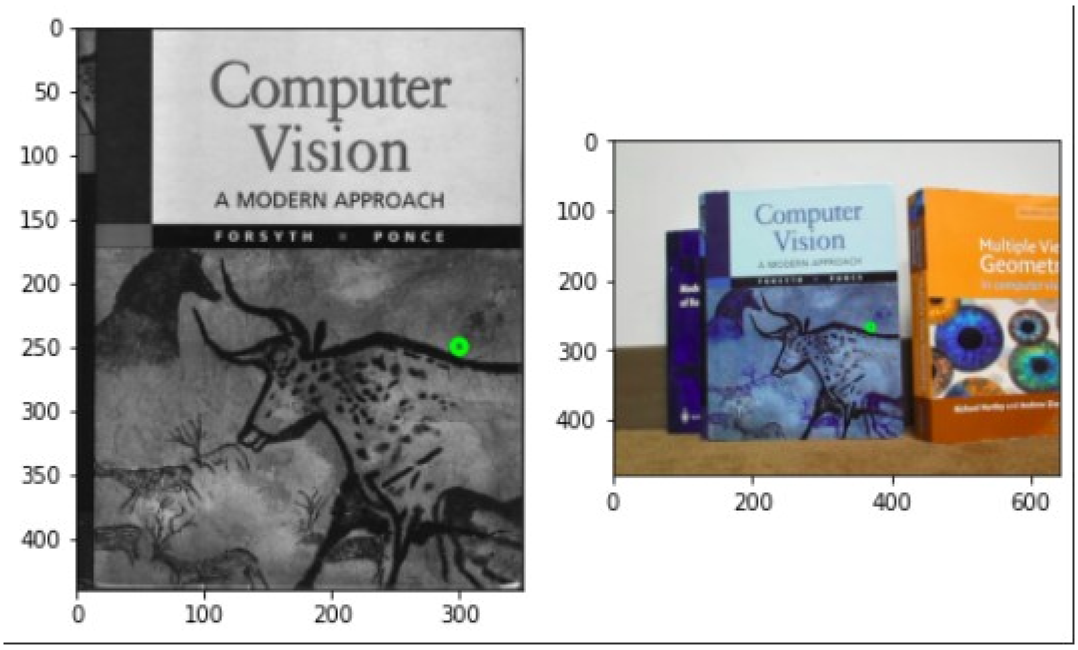
$$P' = H \cdot P$$

$$P' = \begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} \Rightarrow p' = \begin{bmatrix} x'/_{w'} \\ y'/_{w'} \end{bmatrix}$$

And here some test points we chosen randomly to see the result

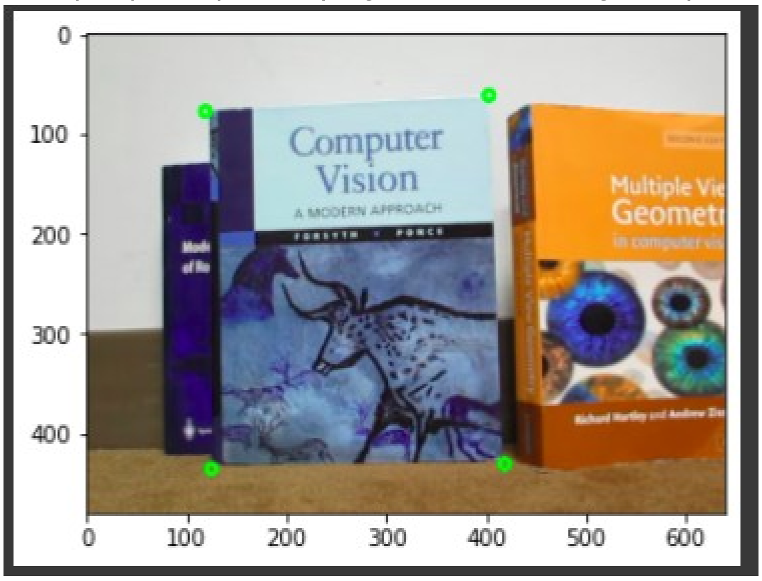






1.5 Calculate Book Coordinates

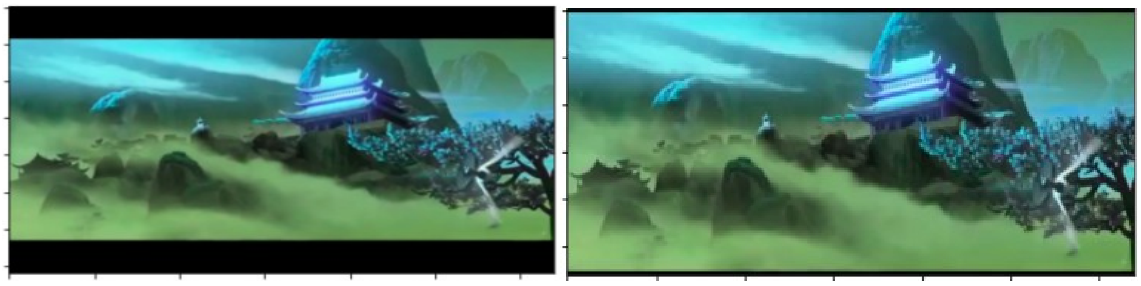
To fit the aspect between the mapping between two images it is required to identify the corners of the book.
We compute the coordinates of the book by apply homography for points
(0,0), (0,W), (L,0) and (L,W) for top left, top right, bot left , bot right respectively



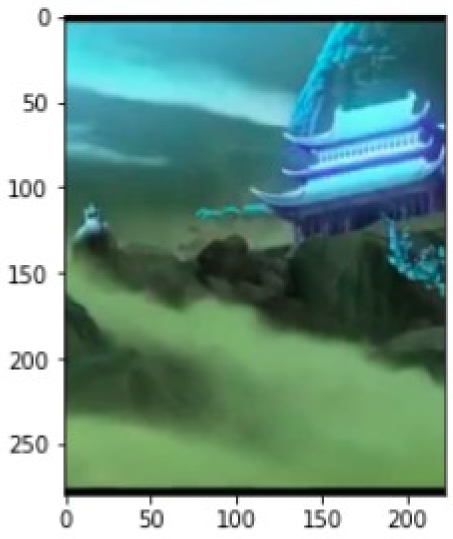
The results seems so promising **although there is some gaps** and the highlight circle has large thickness value **but for video frames it will be accepted**

1.6 Crop AR Video Frames

The original video contains some black margins from top and bottom so we decided to crop it 40 row pixels from top and bot

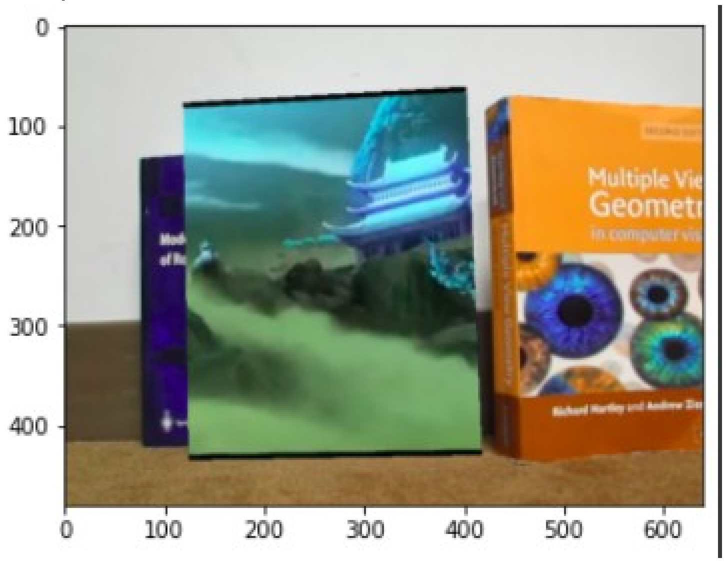


Then to keep the aspect ratio equally to crop the around the center of the frame then resize to return dimensions as the original frame as shown below



1.7 Overlay The first two frames of the two videos

We apply homography to every pixels from the first video frame to the another one to get the mapping then copy the pixel value to that location and the results was

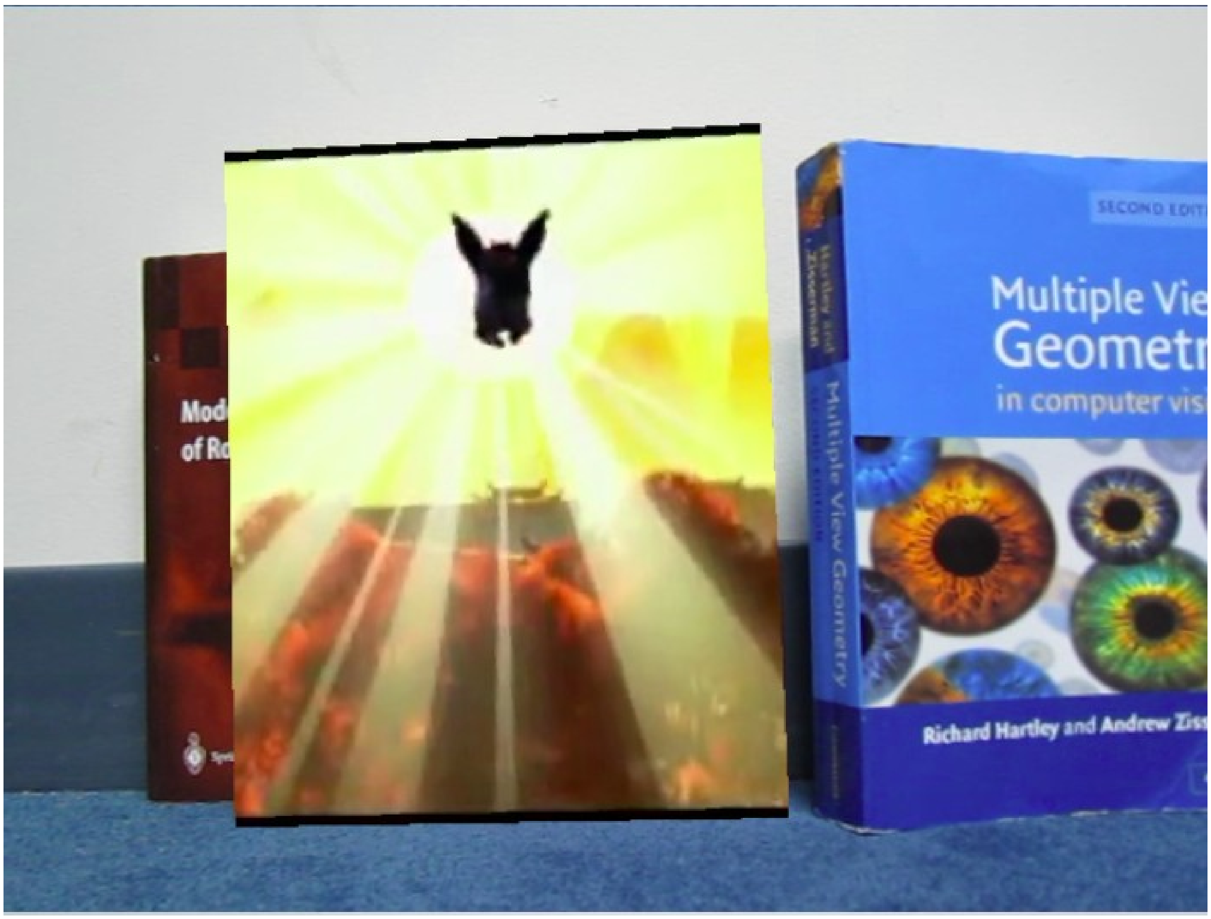


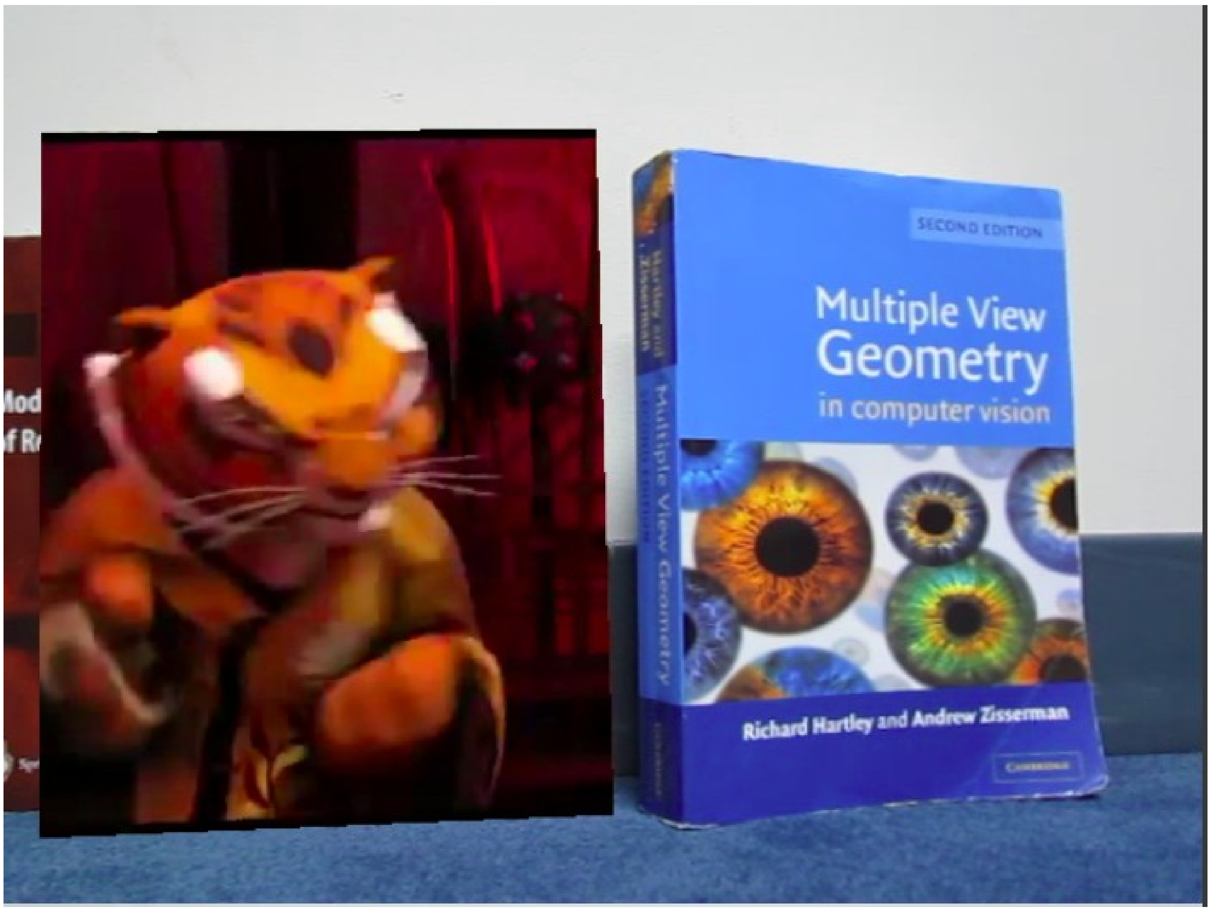
1.8 Creating AR Application

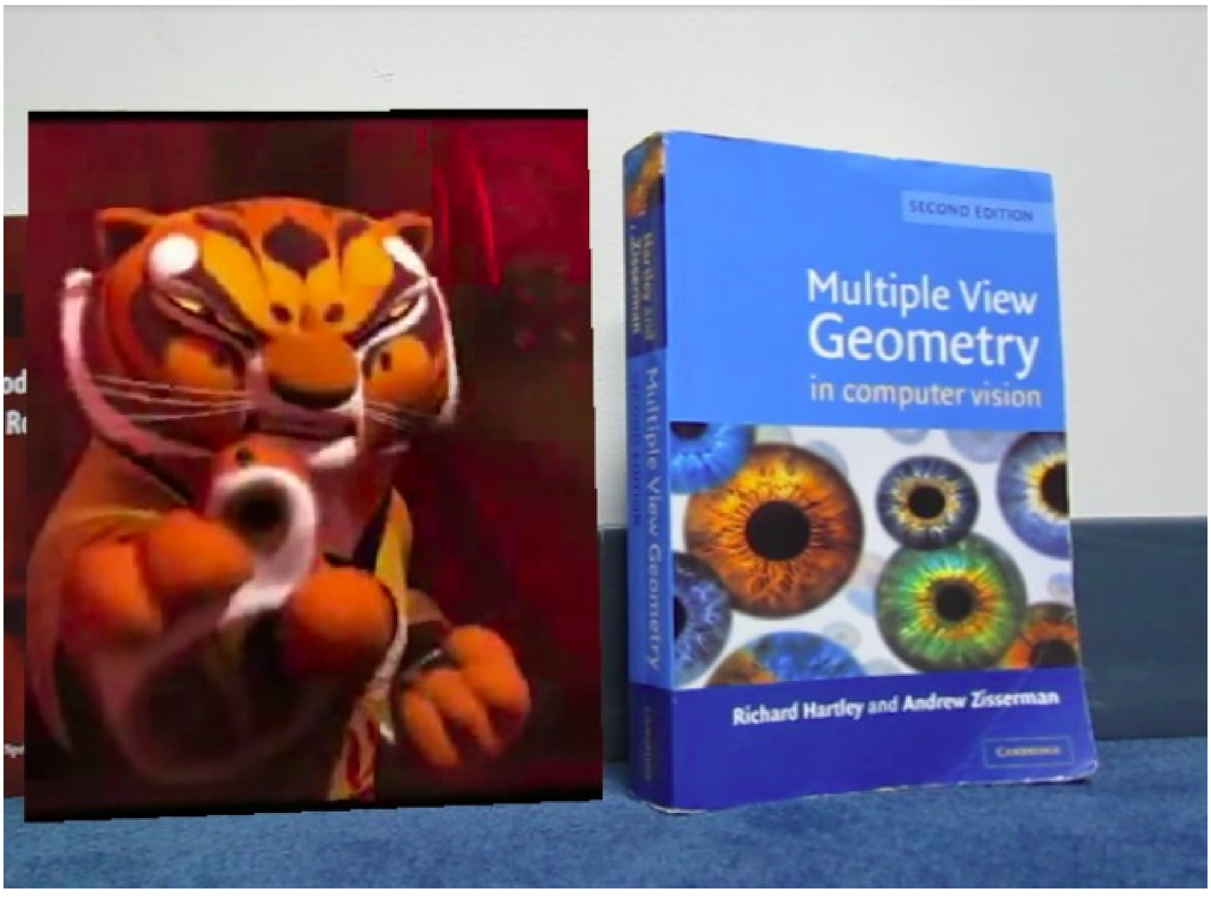
Till now we were using only the first frame of the cartoon video, to create the video we need to compute homography with each frame and perform the homography and overlaying the images together, so the steps was

1. Getting correspondences between the cover book and the book video
2. Compute the homography matrix using the computed correspondences
3. Perform the cropping for the cartoon video
4. Overlay the two frames together
5. Repeat

And here some frames from the output video

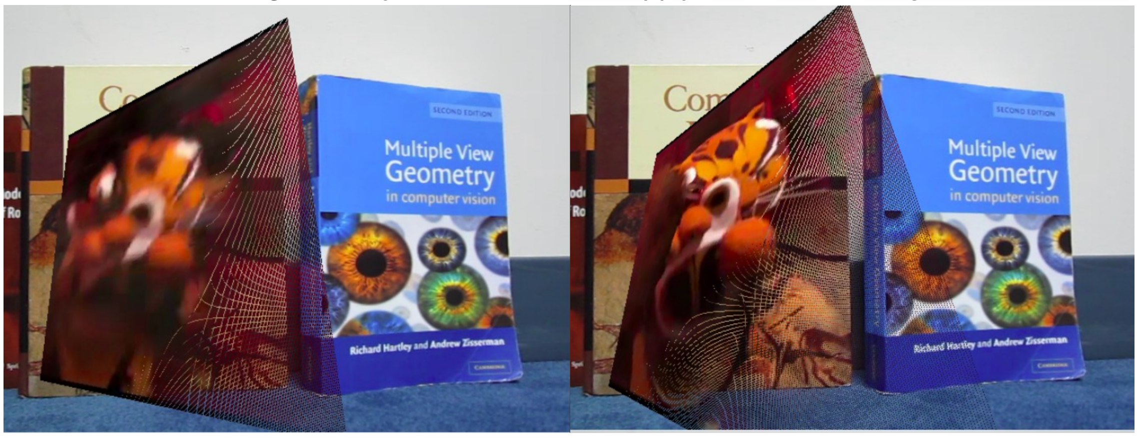



SECOND EDITION
Multiple View Geometry in computer vision
Richard Hartley and Andrew Zisserman
CAMBRIDGE


SECOND EDITION
Multiple View Geometry in computer vision
Richard Hartley and Andrew Zisserman
CAMBRIDGE

1.9 Some Observations

The video contains two frames corrupted, and the reasons was that the **key points generated by the sift was noisy and contains a lot of outliers** so the homography matrix **wasn't accurate enough** and by default we didn't apply the RANSAC algorithm


SECOND EDITION
Multiple View Geometry in computer vision
Richard Hartley and Andrew Zisserman
SECOND EDITION
Multiple View Geometry in computer vision
Richard Hartley and Andrew Zisserman

1.10 Links

- Video: https://drive.google.com/file/d/1CcOewIX3FFAKNfz9KAztspZmfJpTLWsf/view
- Notebook: https://colab.research.google.com/drive/1QHUcFyRf_kGMODH5Sf9CQ5MXypJn0NgX?usp=sharing