Computer Vision
Assignment 2

Name:	ID:
Manar Amr Mohammed	18011842
Nour Waled Ali	18012006
Youssef Mohamed Hussein	18015031

Part1: Augmented Reality with Planar Homographies

- *Getting Correspondences:*

  The correspondences between the book cover image and each frame of the video is found by applying the following steps:

  1. A SIFT descriptor is created to find key points of the book cover image and the book video frame.
  2. A Brute force matcher is used with the matching way as KNN with size 2 to get the corresponding key points between the 2 images.
  3. Ratio test is applied with ratio = 0.75 to filter the good correspondences.
  4. Correspondences are sorted ascendingly by their distances and the first 50 correspondences are returned.

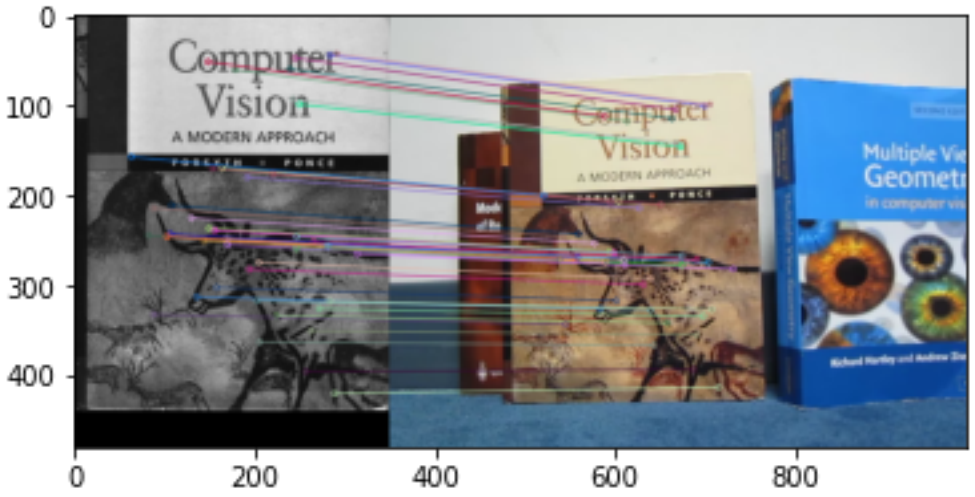


Image 1: Correspondences between the book cover and the book in the video.

- *Compute the Homography Parameters:*

  The correspondences obtained in the previous step are used to compute the homography matrix H. H is then used to project any point from the book cover image to the book video frame. The following steps are applied:

  1. The coordinates of matching correspondences are obtained as 2 lists of points
  2. The RANSAC algorithm is applied to get only inlier correspondences that can produce a good homography matrix. The algorithm uses the homography matrix calculated by solving a system of linear equations AH = 0.

$$\begin{bmatrix} -x & -y & -1 & 0 & 0 & 0 & xx' & yx' & x' \\ 0 & 0 & 0 & -x & -y & -1 & xy' & yy' & y' \\ \begin{bmatrix} -x & -y & -1 & 0 & 0 & 0 & xx' & yx' & x' \\ 0 & 0 & 0 & -x & -y & -1 & xy' & yy' & y' \end{bmatrix} \\ \vdots \\ \begin{bmatrix} -x & -y & -1 & 0 & 0 & 0 & xx' & yx' & x' \\ 0 & 0 & 0 & -x & -y & -1 & xy' & yy' & y' \end{bmatrix} \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \\ h_7 \\ h_8 \\ h_9 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

3. The correspondences returned from the RANSAC are then used to compute the final homography matrix, which represents the best model for these correspondences.

- *Calculate Book Coordinates:*
    1. The book cover image corners are calculated.
    2. The book corners in the video are computed using the homography matrix. The center is calculated as the average of corner points coordinates.
    3. Calculate panda frame corners using the projected book corners coordinates, and Calculate its center.

image 2: book corners and center.

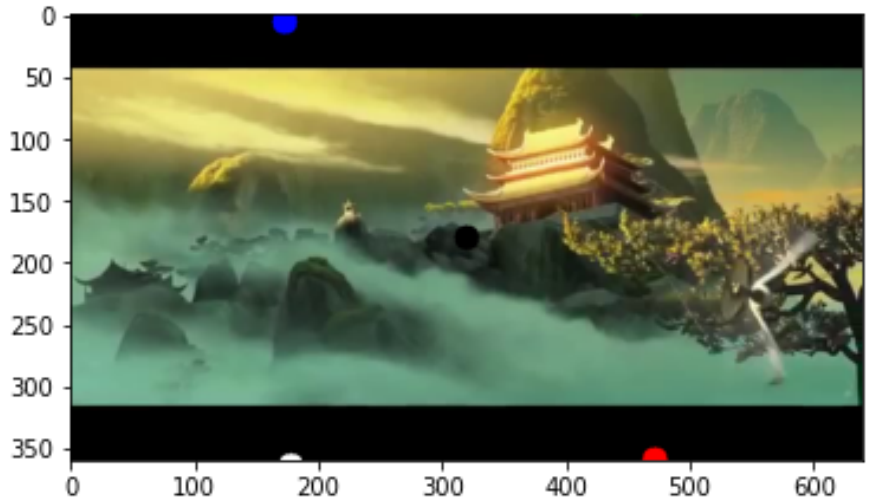


image 3: panda frame corners and center coordinates.

- *Crop AR Video Frames:*

  The panda video frames are cropped to fit onto the book cover using the following steps:

  1. Padding of black pixels is added to the panda frame in case this frame is smaller than the book cover.

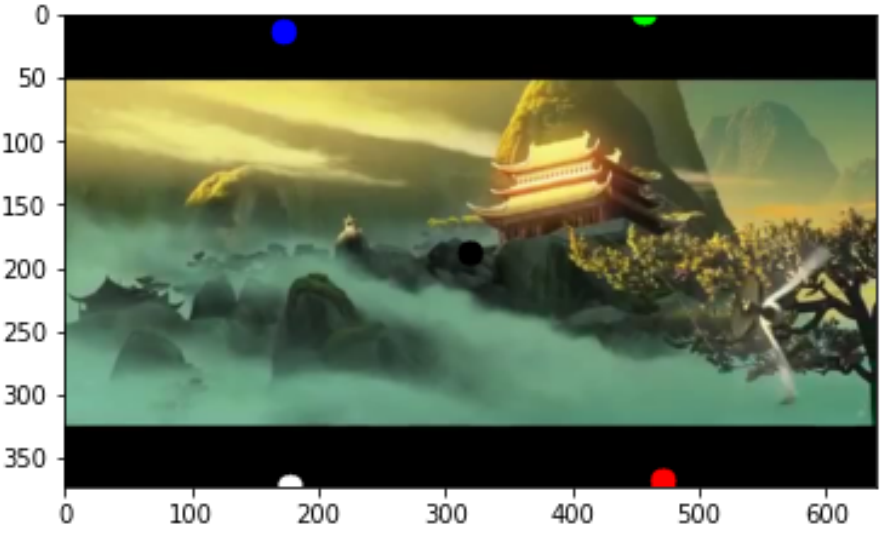


image 4: panda frame after adding some padding

2. A mask array is created with the same size as the book in the video, and then applied on the panda frame to crop these frames.

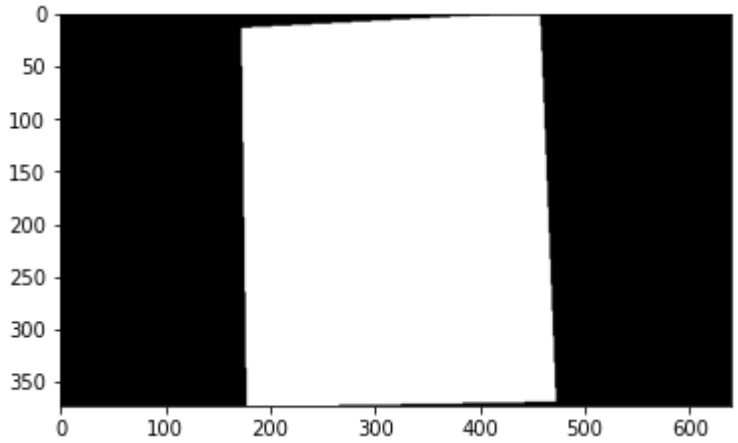


image 5: panda frame mask

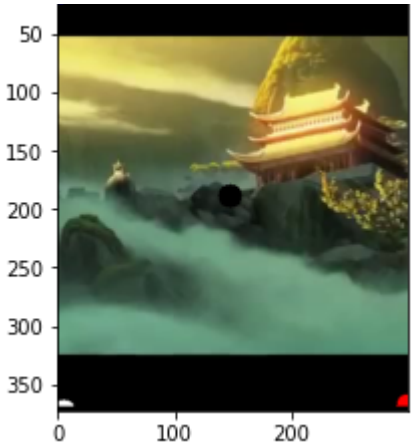


image 6: panda frame cropped (rectangle crop)

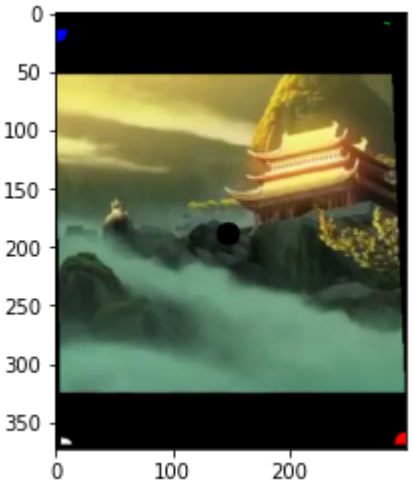


image 7: panda frame after padding cropped (rectangle crop)

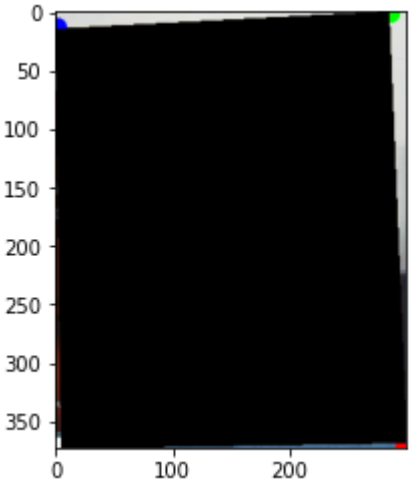


image 8: book masked with the inverse mask in book frame.

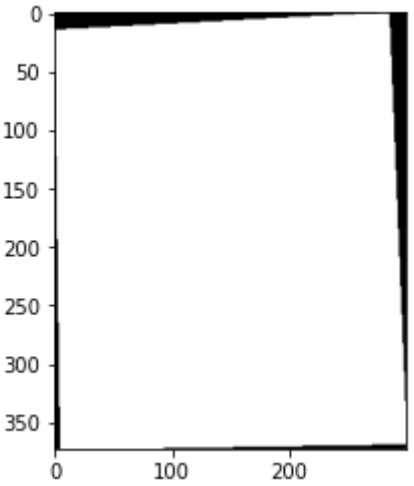


image 9: cropped mask image.

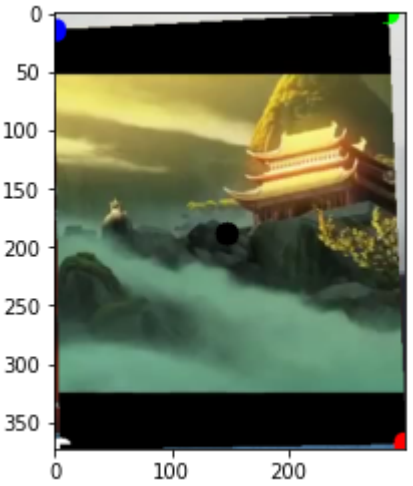


image 10: overlaying panda frame on book frame.

- *Overlay the First Frame of the Two Videos:* The book cover in the first video frame is replaced with the cropped panda frames from the panda video.

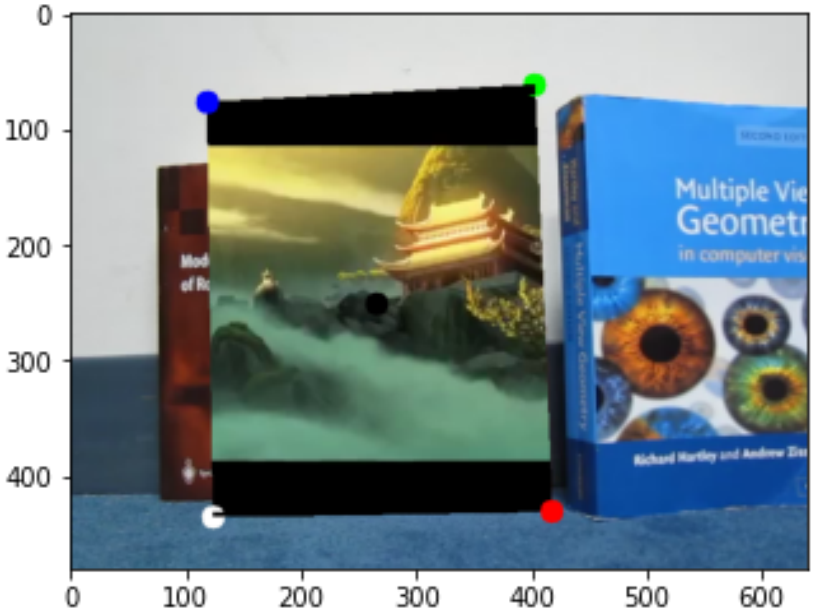


image 11: Result After Overlay

- *Creating AR Application:* All previous steps are repeated for each frame to create the video frames.
  Video: [link]

Part2: Image Mosaics

- *Getting Correspondences and calculating homography matrix:* Used the previously implemented functions (Read Part 1)

- *Forward Warping:*

  1. Corner points of the source image are used to find the bounding box of the destination image.
  2. 3 new arrays of zeros representing the 3 channels of the transformed image are initialized.
  3. Each point from the source image is transformed to the destination image using the homography matrix.
  4. All 3 channels are merged.

- *Backward Warping:*

  1. Calculate new homography matrix (Inverse matrix)
  2. Transforming every black pixel in the destination image back to its position in the source image.
  3. Use bilinear interpolation to approximate missing values

- *Output:*

Bonus

- *Image 1:*

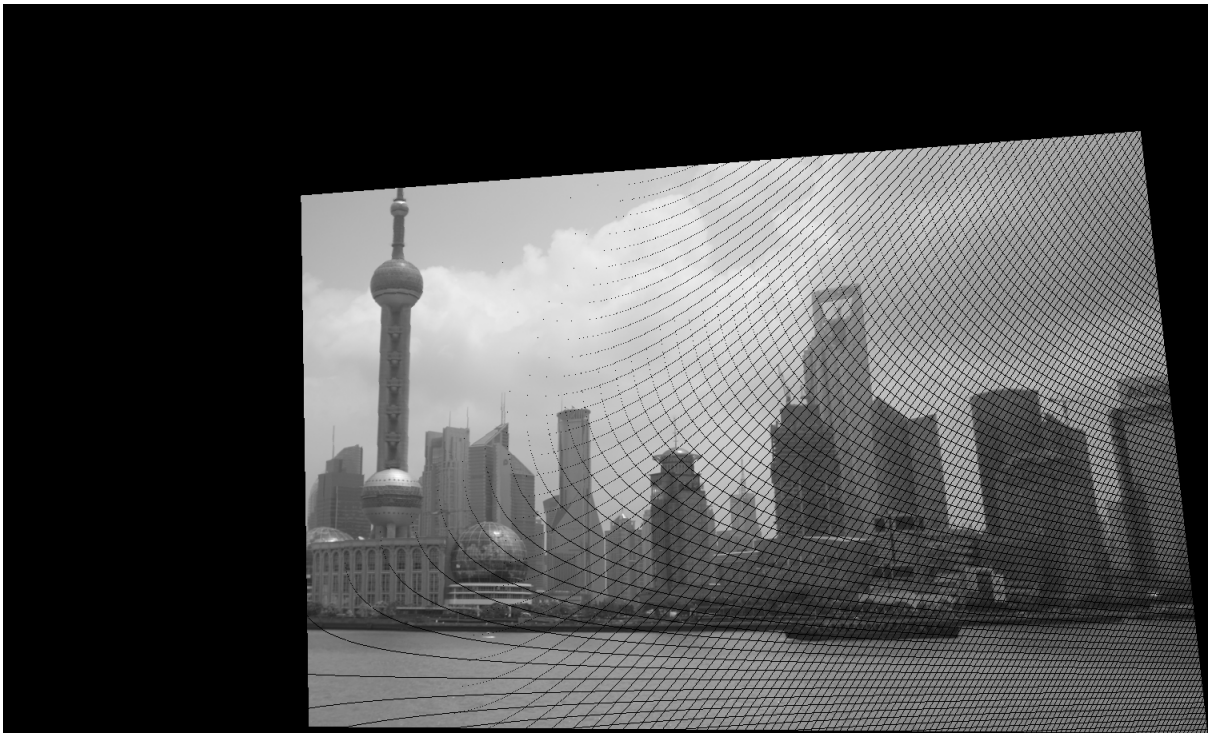
  Forward warping:

  Backward Warping:

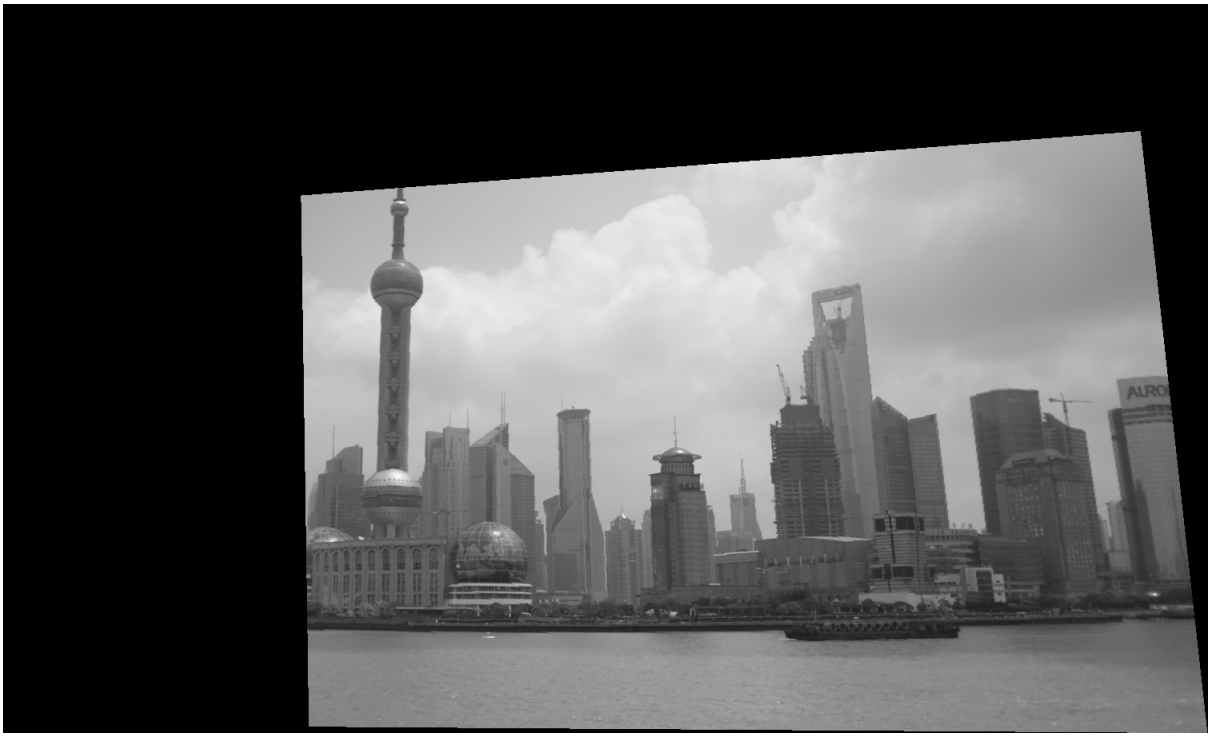
Combine with image 2:

- *Image 3 :*

  Forward warping :



Backward warping :



- *Final Output:*

References:

https://docs.opencv.org/3.4/da/df5/tutorial_py_sift_intro.html

https://docs.opencv.org/4.x/dc/dc3/tutorial_py_matcher.html

Colab Notebook: part 1

part 2