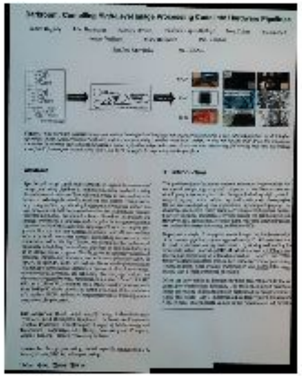
MOBILE PAGE SCANNER

Fig. 5. Multiple Input Images

https://stacks.stanford.edu/file/druid:bf950qp8995/Badlani_Akinola_Li.pdf



Final Output Image

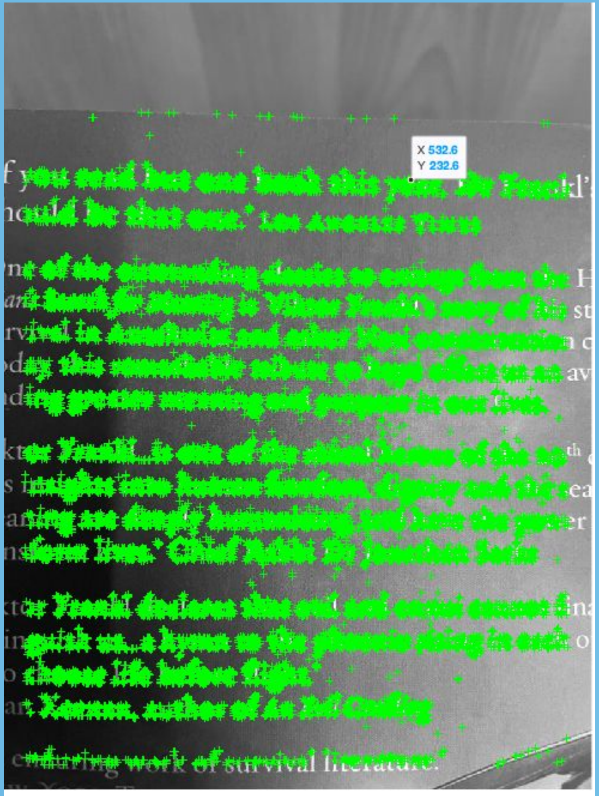
INTRODUCTION, WORK DISTRIBUTION

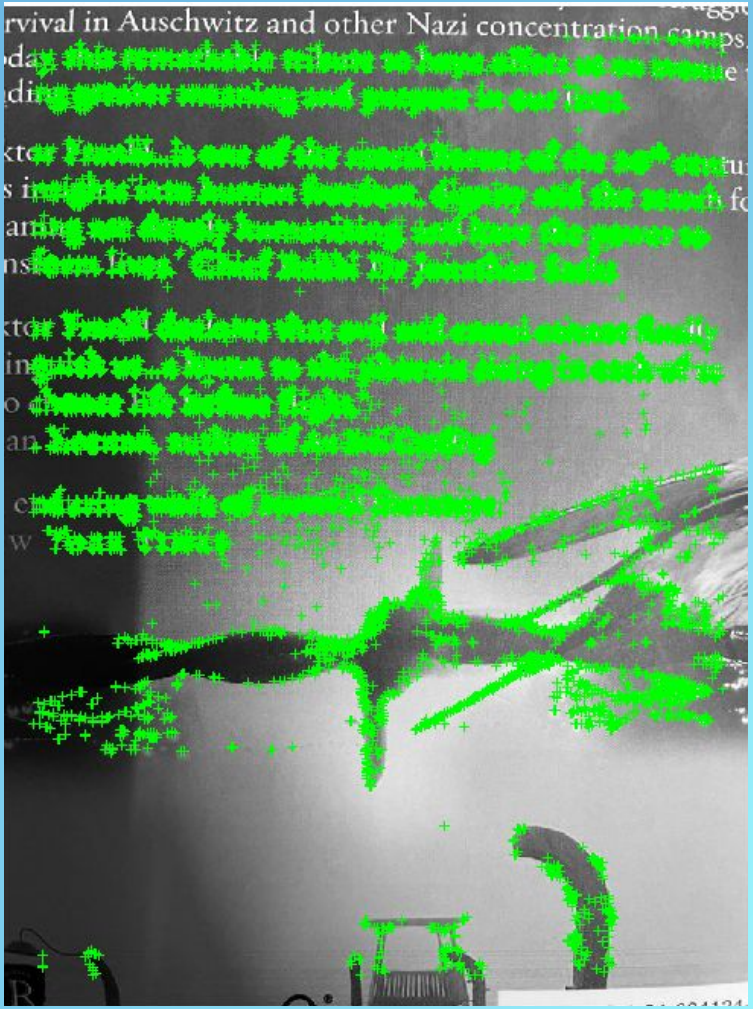
Rishab Khantwal- 180100095- Feature detection, Feature matching, calculation of homography matrix

Akshay Iyer- 190070006- Finding affine transformations, image warping, bundle adjustment

Amrit Rao- 19D070008- histogram matching, multi band blending, combining all the parts together

FEATURE DETECTION
THE ORB DETECTOR



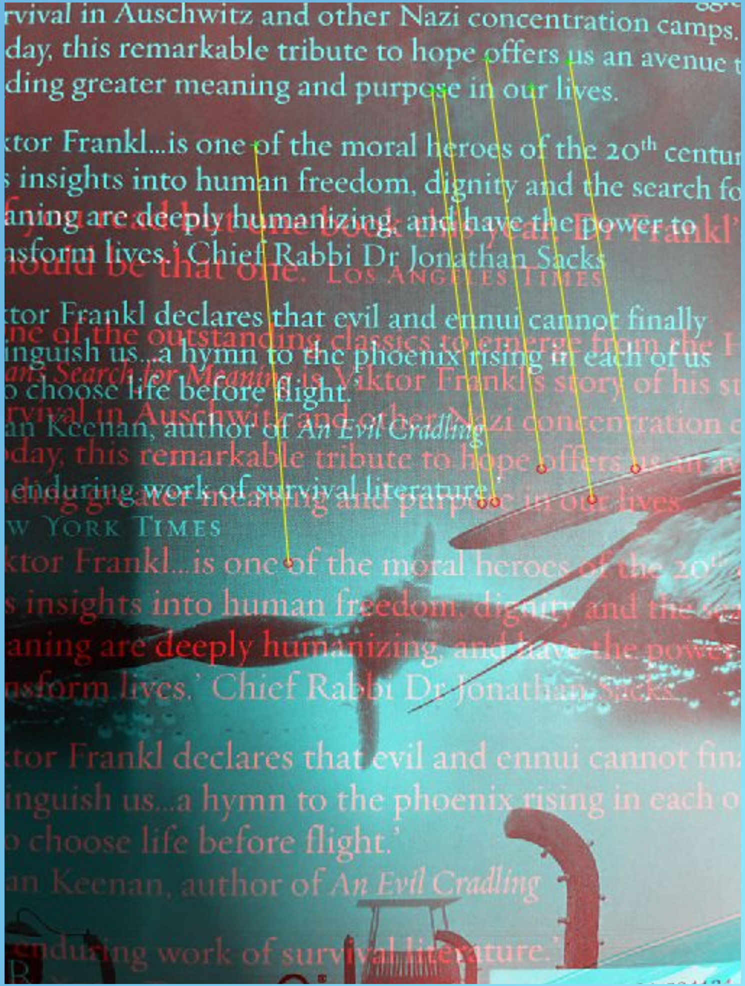


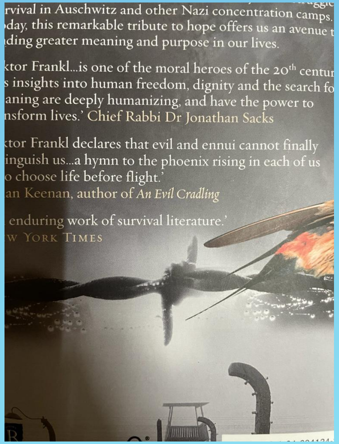
FEATURE MATCHING

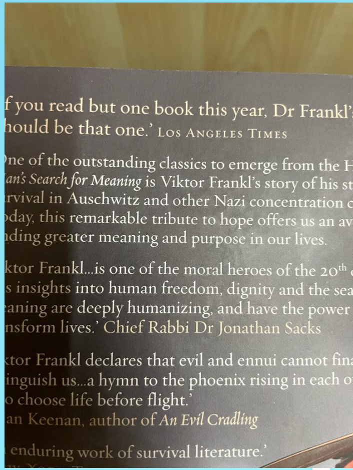
Once we have the ORB features for all the images, we can perform feature matching on each pair of images.

For a particular feature in image A, we can find its nearest and second nearest neighbor in image B. If its nearest neighbor is much closer than its second nearest neighbor, then the feature in image A and its nearest neighbor in image B is a good match.

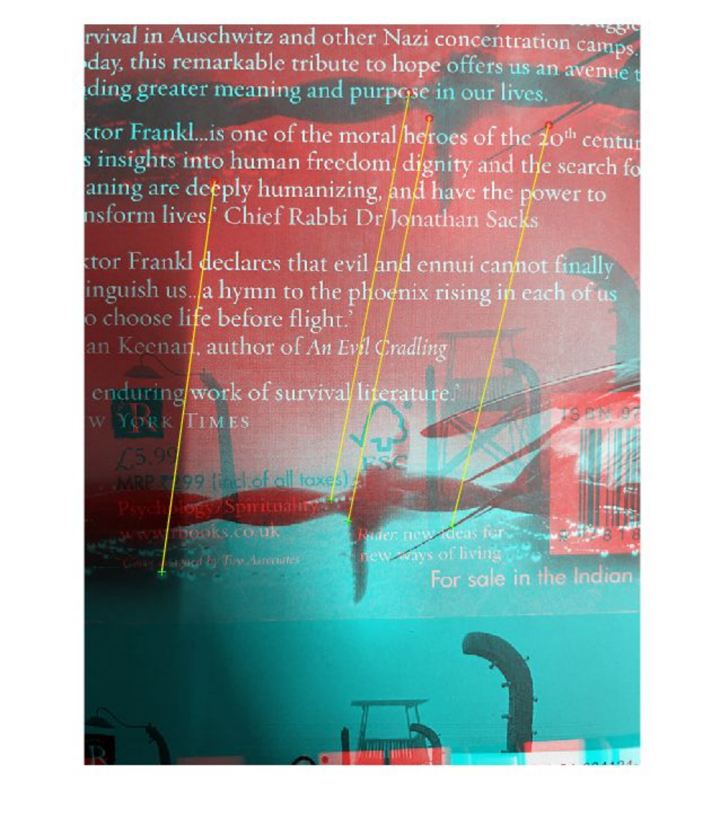
In this way, we can get all pairs of matched features between each pair of two images.







Homography estimation



1. For pair of images for which we find the matching features more than 4. We find the Homography matrix

2. To estimate the homography matrix we need 4 pairs of matching features, but actually we have much more.

3. Each group of 4 matched features corresponds to a homography matrix and we need to use RANSAC (random sample consensus) algorithm to pick out the best

FINDING AFFINE TRANSFORMATIONS

Now even after we have established which matrices are compatible with each other, we need to find out what the affine transformation is between features of matched images in order to make them match perfectly.

This is by minimizing the error between corresponding points.

Now in order to keep the images as seen only from one angle, we successively keep transforming the images back to the position of the previous image.

BUNDLE ADJUSTMENT

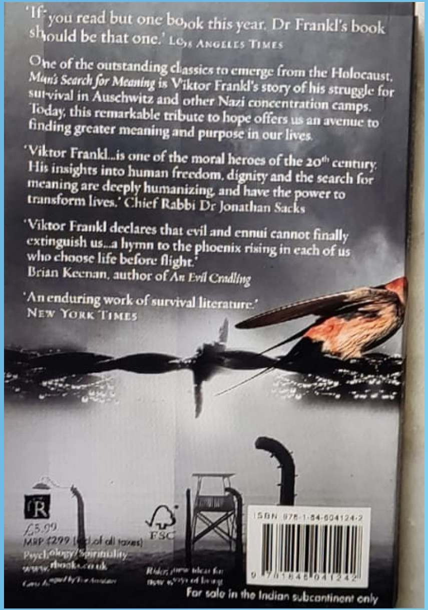
We tried to implement bundle adjustment too.

It is a function which implements the transformations between images, in the form of a global matrix, so that all the observed images look like they are being viewed from a single location.

It uses the levenberg marquardt algorithm in order to do so.

IMAGE WARPING

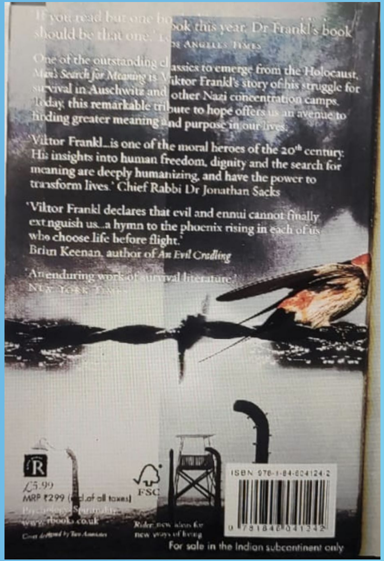
We then merged all of these images into a single panorama image, by using the affine transformations we calculated in the previous steps.



HISTOGRAM MATCHING

We also carried out histogram matching of the individual images we combined, with the central image we have.

This was as, without this step, contrast exposure etc of different images may be different.
Hence by doing this we have made the individual images more similar. This is how our image looks with Histogram matching.



MULTI BAND BENDING

We also tried to implement multi band bending.

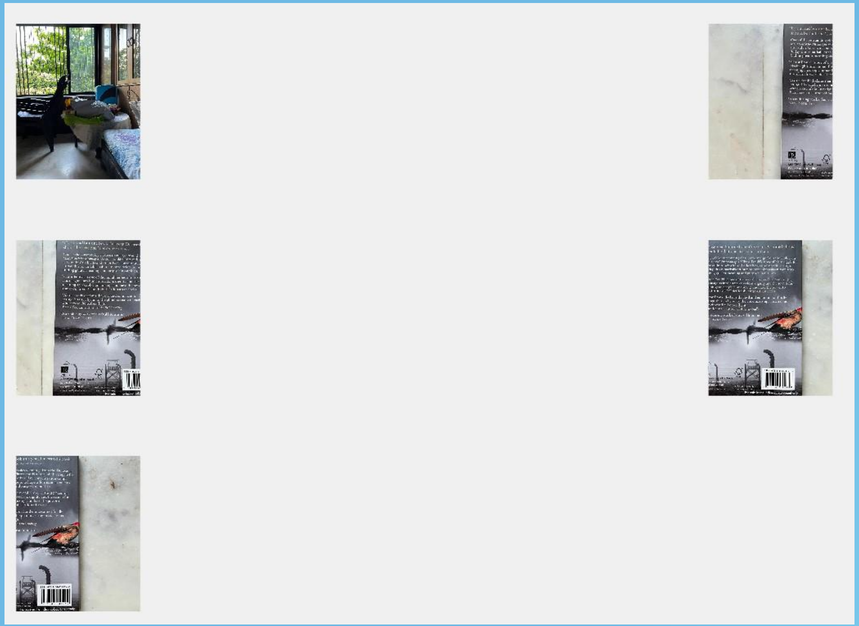
Even after histogram matching, it is possible that the edges of the individual images in our final warped image have different intensities on both sides.

This might give rise to us seeing edges inside the net panorama image, that separates sections inside it.
We also cannot linearly add all the images and then take their average, as this will lead to a loss of high frequency content in the original image. Therefore, we want to smooth over a large range of low frequencies, and a short range of high frequencies. For this, we can use multi band blending.

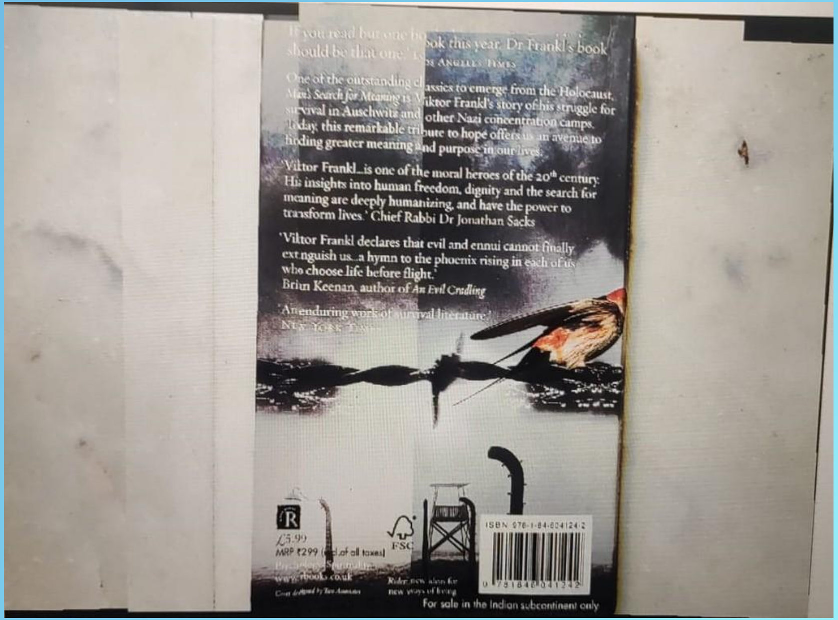
OVERALL ALGORITHM

1. Detect the features in the given images
2. Compare the images and find matching features
3. Use RANSAC algorithm to find which images can be stitched into panorama
4. Histogram match all the images with respect to the central image.
5. For each image, compute the affine transform matrix T relative to the previous image
6. Combine all Transformation matrices with respect to one single image
7. Warp all the images into a single panorama using above transformations.

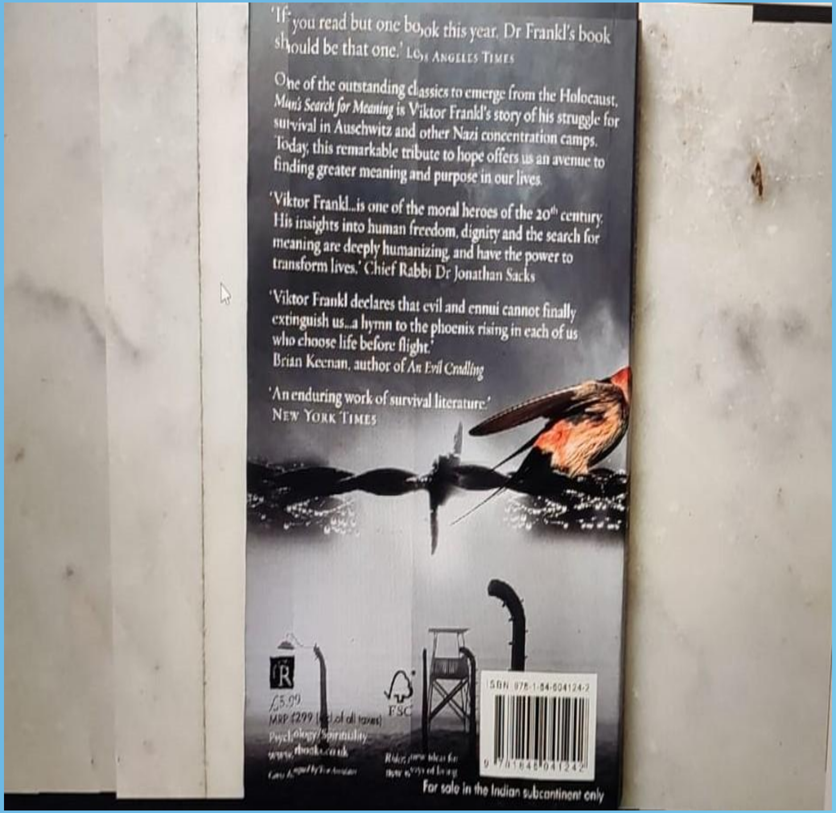
RESULTS



Initial set



Final Restored Image



Without Histogram Matching

With Histogram Matching