RBE 549- Project 1:MyAutoPano

Ankit Mittal
Department of Robotics Engineering
Worcester Polytechnic Institute
Email: amittal@wpi.edu

Rutwik Kulkarni
Department of Robotics Engineering
Worcester Polytechnic Institute
Email: rkulkarni1@wpi.edu

***Abstract*—tilizing one late daytilizing one late dayuIn this project, we propose techniques for stitching images to generate seamless panoramic scenes. The process involves using pairs of images with overlapping regions. Our objective is to merge these images into a single panoramic image. Our approach encompasses both traditional computer vision methods and deep learning techniques. The classical method we've used is a feature-based technique that establishes a relationship between images by identifying and matching features. For the deep learning component, we employed both supervised and unsupervised learning strategies to estimate the homography, which is the transformation between a pair of images, to achieve the desired output.**

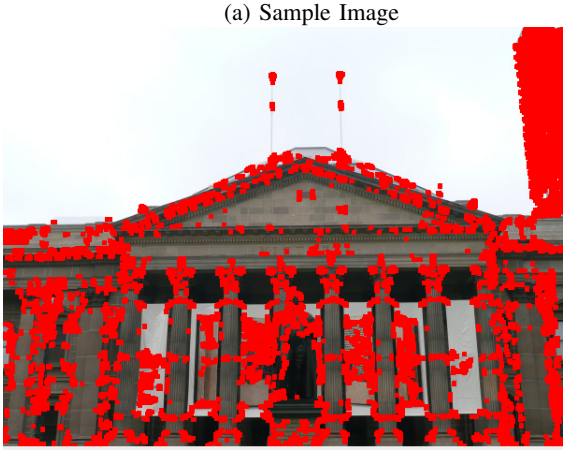
I. Phase 1: Classical feature based technique

Feature-based methods focus on identifying features present in the overlapping areas of images. These features, known as keypoints, are matched between the images. We use this matching to estimate the homography, which represents the transformation between these sets of points from the two images. This homography is then applied to warp the second image to align with the first. Following this alignment, the images can be seamlessly stitched together, using the first image as the reference. Each subsection of our documentation provides detailed explanations of the methodologies used and the resulting outputs of the images.

A. Corner Detection

The concept centers on establishing connections between images by identifying a set of features. Corners are particularly effective for this purpose as they are discernible from various viewpoints. By detecting as many corners as possible in a given image, we can compare these features between images. This comparison provides insights into the geometric relationship between the images. For corner detection, we utilized the Harris corner detection function available in OpenCV. We determined the following parameters to be optimal for corner detection: a kernel size of 3, a Harris K parameter of 0.04 Figure 1 illustrates the corners detected in the images using these parameters.

(a) Sample Image



(b) Harris Corners

Fig. 1: Image corner detection

B. Adaptive Non Maximal Suppression (ANMS)

Upon detecting corners in each image, the challenge is to identify the "best" corners, characterized by their distinctiveness among local peers and even distribution across the image for superior homography. This is addressed using Adaptive Non-Maximal Suppression (ANMS), which operates by first selecting local maxima among the corners, focusing on those farthest from stronger corners. The essence of ANMS is its effectiveness in evenly distributing corners, countering the tendency of corner detectors like the Harris method to identify clusters of corners, rather than individual ones. This is due to corners encompassing multiple pixels, particularly in high-resolution images. ANMS overcomes this by choosing points maximally distant from stronger corners, ensuring that when the top N best corners are selected, they represent individual points from each cluster. Figure 2 illustrates the transformation from clustered to well-distributed, optimal corners achieved through ANMS.

(a) Sample Image 1

(b) Sample Image 2

Fig. 2: Image corner After ANMS

C. Feature Descriptor

To effectively compare corners across images, we assign them unique identities using Feature Descriptors. In line with our approach outlined in the problem statement, we first select corners that fit well within a 40x40 dimension. These selected corners are then flattened into a 1D array. To create a representative sample, we pick pixels at every 25th index, essentially capturing every 5th pixel in a row-wise or column-wise manner. This process results in a patch of dimensions 8x8. We standardize and blur this patch to ensure a smooth variation, providing each corner with a distinct and comparable identity, encapsulated in its Feature Descriptor. This methodology allows for precise comparison and alignment of corners across different images.

D. Feature Matching

The feature descriptors obtained through our process are then matched with descriptors from other images, a crucial step as it indicates the extent of overlap between the images. Ideally, photometric comparisons would provide the most accurate measure of overlap, but these methods are computationally expensive. Instead, we rely on geometric features like corners. Our comparison algorithm is based on David Lowe's ratio test.A simplistic approach would have been to minimize the distance between feature descriptor vectors, but this risks false positives leading to inaccurate matches. The ratio test circumvents this by comparing the distances of the first and second best feature descriptors. If the ratio of these distances falls below a certain threshold, the pair is considered a good match and is added to the feature match set. This process ensures the uniqueness and significance of each match. If the second-best match is comparably close to the first, indicating a lack of uniqueness, the pair is discarded. This method effectively screens for false positives, providing a straightforward and efficient way to match features across different data sets.

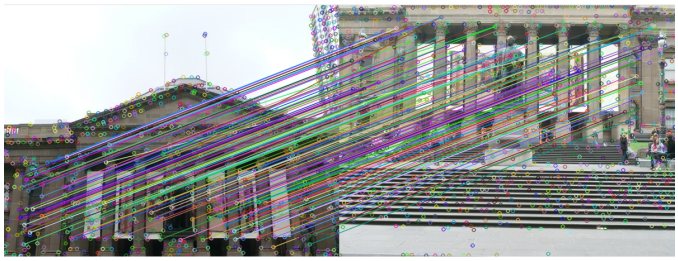


Fig. 3: Feature matching

E. RANSAC

The feature matches we've identified, like any data set, are susceptible to inaccuracies and outliers. To distinguish between valid data and outliers, it's necessary to employ a data model to fit and analyze the data. While manual pruning or using standard deviation and quartiles for data selection are possible methods, they require extensive tuning and may not generalize well across different data sets. Instead, we utilize Random Sample Consensus (RANSAC), a robust method that provides a probabilistic approach to creating a dataset free of outliers. RANSAC helps in identifying the minimal subset of data that best fits our model and generalizes effectively to the rest of the data points. In our context, the model in question is the homography estimation between a pair of images. Applying the RANSAC algorithm, we are able to refine the feature matches.

F. Blending Images(Warping and Stitching)

In our project, we developed a process to stitch two images together using a homography matrix. This process begins by assessing the dimensions of each image and identifying their respective corner points. Subsequently, we apply a perspective transformation to one of the images using the derived homography matrix. The next step involves creating a composite canvas, sized to accommodate both the original and the transformed images. This step is crucial as it involves translating the images to ensure all coordinates are positive, thus preventing any part of the images from being lost. Once the canvas is prepared, the function warps the second image in accordance with the pre-calculated transformation and integrates it onto the canvas. The final and critical step

in this process is the overlaying of the first image onto the warped version of the second image. This results in a seamless and coherent panoramic image. Our method is particularly effective in merging images that have overlapping regions, thereby producing a single, expansive image. This approach has proven to be an efficient solution for creating panoramic views from multiple images.

Fig. 4: Image 1 & 2 Blending

G. Results

In our test set, we encountered some images that either contained irrelevant content ('garbage') or had very minimal overlap with other images, resulting in few or no feature matches. To address this issue, we implemented a threshold criterion for feature matching. We established that if the number of feature matches between a pair of images fell below 10, we classified the image as unsuitable for stitching. Consequently, such images were excluded from the stitching process. This approach ensured that only images with sufficient overlap and relevant content were considered, thereby enhancing the overall quality and coherence of the stitched panorama. Then we also tried the graph approach where tried to find the best matching image(having maximum overlap). Computing this on normal image size is computationally expensive and for the best matching image for stiching we rsized all the images in the lower resolution. To further refine our image stitching process, we explored a graph-based approach aimed at identifying the best matching image, defined as the one with the maximum overlap. Recognizing that computing matches on full-sized images is computationally intensive, we adopted a strategy of resizing all images to a lower resolution. This resizing step significantly reduced the computational load, enabling a more efficient determination of the most suitable image for stitching. By focusing on lower-resolution images, we were able to swiftly and effectively identify the image with the highest degree of overlap, which is critical for achieving a seamless and high-quality stitched panorama.

Fig. 5: Set 1 Blending

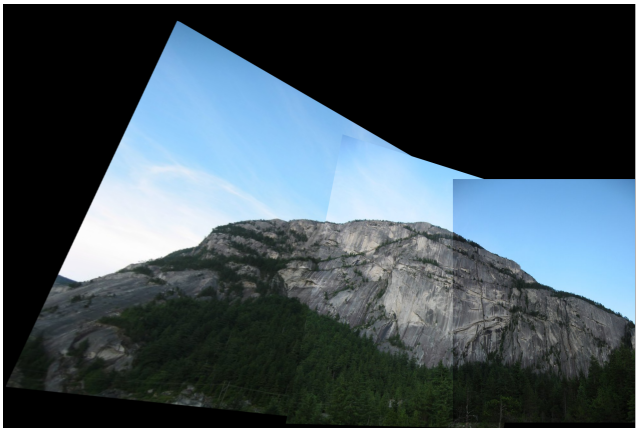


Fig. 6: Set 2 Blending

Fig. 7: Set 3 Blending

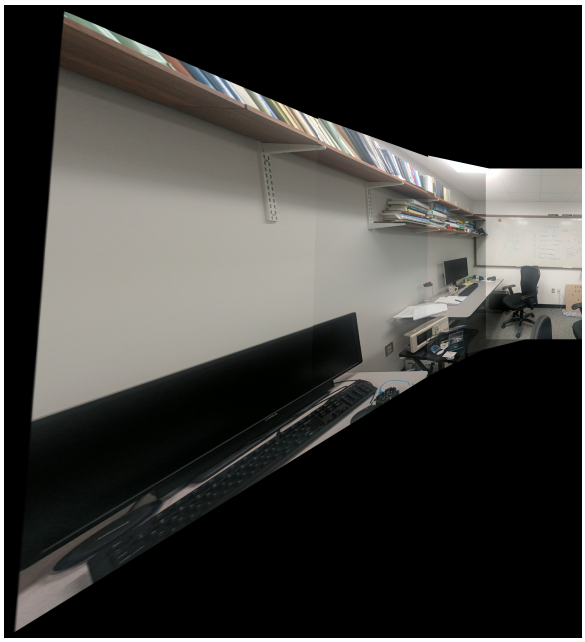


Fig. 9: TestSet 2 Blending

Fig. 8: TestSet 1 Blending

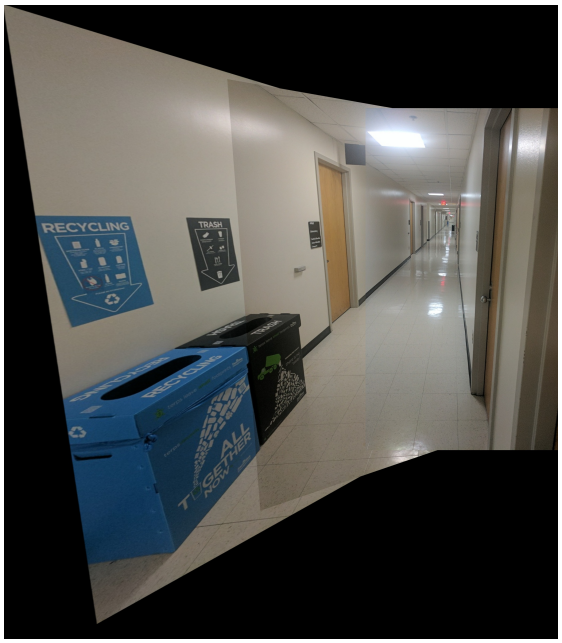


Fig. 10: TestSet 3 Blending

II. Deep Learning for Homography Estimation

A. *Data Generation*

In this approach, we utilize a subset of 10,000 images from the Microsoft COCO dataset to generate pairs of images (I_A, I_B) that simulate consecutive frames in a video sequence, focusing on small perspective and positional changes. Each image pair, representing different perspectives of the same scene, includes an original patch (denoted by a white box) and a perturbed patch (denoted by a black box) in the respective images. The perturbations are parameterized by ρ for perspective and γ for translation, both ranging from -16 to +16, to mimic the slight variations typical in video frames. The homography transformation, which accounts for these changes, is mathematically expressed as a function of ρ and γ, and is used to warp the original image. The process ensures that in the warped image, the perturbed patch aligns back to the shape of the original patch.

The ground truth labels for each image pair are derived based on the computed homography transformation. These labels represent the necessary parameters to align the perturbed patch in I_B with the original patch in I_A, effectively capturing the homography between the two images. This dataset generation method is specifically designed to provide a challenging and realistic training environment for models focused on homography estimation and image stitching in video sequences, offering a practical approach to simulate real-world scenarios in image stitching tasks.

Fig. 11: Original Image

Fig. 12: Warped Image

Supervised Approach

A. *Model Architecture: Modified VGG for Homography Estimation*

This Convolutional Neural Network, created for estimating the hpoint4 parameter crucial in image stitching, draws inspiration from the VGGNet architecture but includes specific adaptations for its specialized task. The network starts with several convolutional layers, each featuring a 3x3 kernel size, followed by batch normalization and ReLU activation. These layers are organized in pairs, with a max-pooling layer of 2x2 window size after each pair. The design follows the VGGNet approach of increasing complexity through stacked convolutional layers, but with an adjustment in filter depth, increasing from 64 in the initial layers to 128 in the later layers.

The structure then shifts to fully connected layers. The output from the convolutional layers is first flattened into a one-dimensional vector, then passed through a dense layer of 1024 units. A dropout layer with a 50 percent rate follows, aimed at reducing overfitting. The final output layer has 8 units, specifically for computing the hpoint4 parameter, which represents the difference between the original and transformed corners of an image patch. This parameter is essential for constructing the homography matrix using Direct Linear Transformation (DLT), a key step in the image stitching process. The network's loss is calculated using a square root mean squared error function, focused on accurate hpoint4 estimation. This model can be seen as a modified version of VGGNet, adapted to fit the unique requirements of homography-based image stitching tasks.

Fig. 13: Supervised Homography Net Model Architecture

B. Training and Validation

During the training process, a notable trend was observed in the behavior of the validation loss. Initially, the validation loss showed a decrease from 21 to 12 throughout the first 6 epochs. However, around the 6th epoch, a divergence between the training and validation losses became apparent. While the training loss continued to decrease, the validation loss ceased to show further reduction. This pattern of the training loss decreasing without a corresponding decrease in the validation loss persisted beyond the 6th epoch, indicating a potential onset of overfitting in the model.

Sr. No.	Hyperparameter	Value
1	Epochs	80
2	Learning Rate	1e-4
3	Batch Size	64
4	Optimizer	AdamW

TABLE I: Hyperparameter Settings for Supervised Homography Net

Model	Number of Parameters
Supervised Homography	67,497,034

TABLE II: Number of parameters in Homography Net

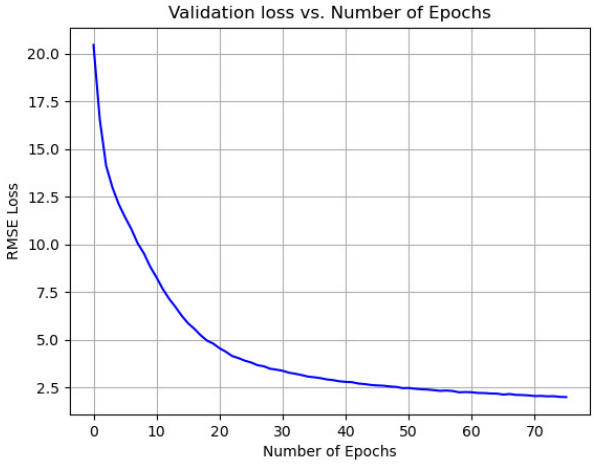


Fig. 14: Training Loss per Epoch for Supervised Model

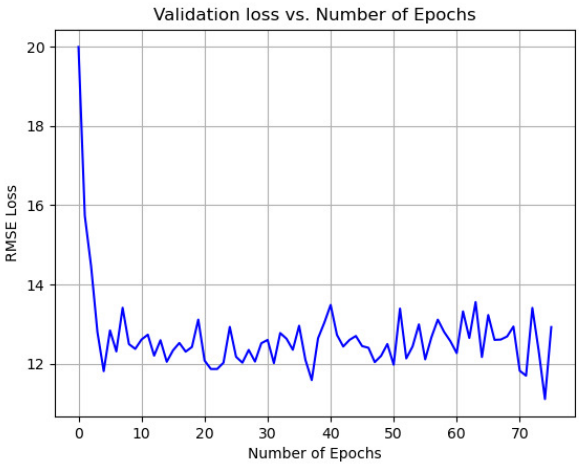


Fig. 15: Validation Loss per Epoch for Supervised Model

UNSUPERVISED APPROACH

A. Model Architecture

The architecture consists of two main components: a convolutional neural network (CNN) and a fully connected network (FCN). It is designed to take two input images and learn a transformation represented by a 4x2 matrix δ to align or match the two images in some way. The network appears to be used in a task where aligning image patches is crucial, possibly for image registration or another computer vision application. Here's a breakdown of the network architecture:

1. Input Layer: The network takes two input images. These images are concatenated along the channel dimension to form a single input tensor.

2 . Convolutional Neural Network (CNN): The CNN component consists of multiple blocks, and each block is designed to extract features from the input images. Each block consists of two convolutional layers with ReLU activation functions, and optionally, batch normalization. Max-pooling is applied after each block if the "pool" parameter is set to True when creating the block. This helps in spatial feature reduction.

3 . Fully Connected Network (FCN): After passing through the CNN layers, the output is flattened using the "Flatten" layer and then passed through a fully connected network. The FCN consists of two fully connected layers with ReLU activation functions and dropout regularization to prevent overfitting. The final output layer has 8 neurons, which corresponds to a 4x2 matrix representing the transformation "δ."

4. Output: The output of the FCN is reshaped to obtain the "δ" matrix, which represents the transformation to be applied to one of the images to match it with the other.

For calculating the loss during training we have used kornia module.

B. Training and Validation

Sr. No.	Hyperparameter	Value
1	Epochs	250
2	Learning Rate	1e-4
3	Batch Size	64
4	Optimizer	Adam

TABLE III: Hyperparameter Settings for Unsupervised Homography Net

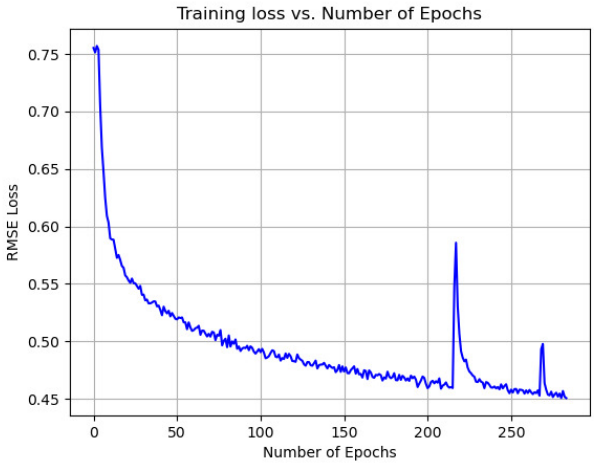


Fig. 16: Training Loss per Epoch for UnSupervised Model

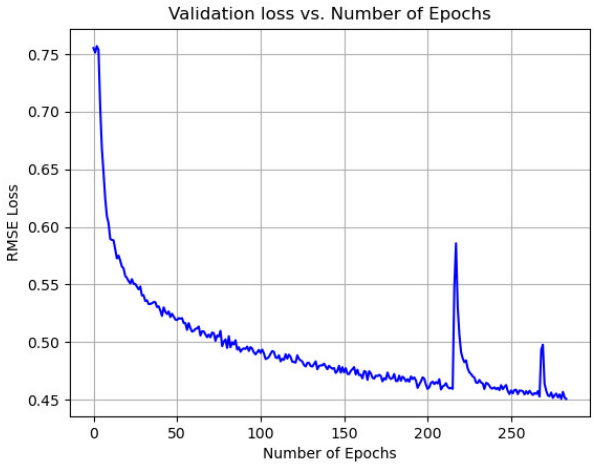


Fig. 17: Validation Loss per Epoch for Supervised Model

III. Results

A. For Supervised Model

Fig. 18: Test Set 1

Fig. 19: Test Set 2

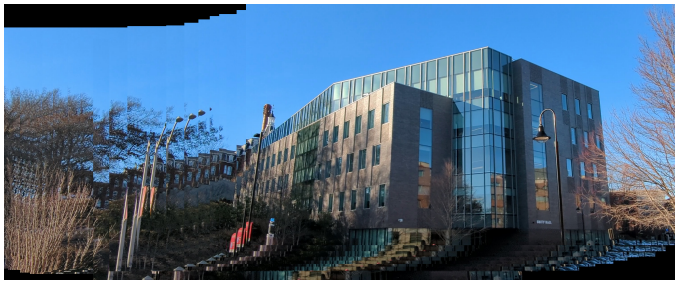
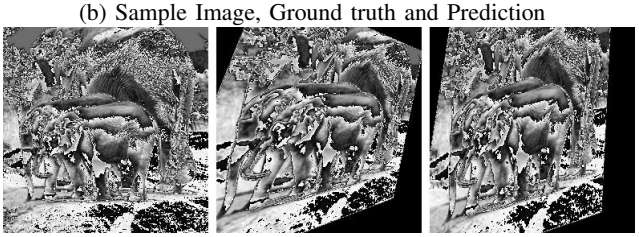


Fig. 20: Test Set 3

B. For Un-Supervised Model

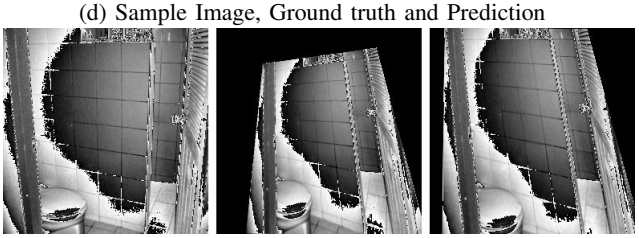
(a) Sample Image, Ground truth and Prediction

(b) Sample Image, Ground truth and Prediction



(c) Sample Image, Ground truth and Prediction

(d) Sample Image, Ground truth and Prediction



(e) Sample Image, Ground truth and Prediction

Fig. 21: Unsupervised Learning results

IV. Observation

A. For Supervised Model

- One of the primary observations from the training of the supervised Convolutional Neural Network (CNN) is its difficulty in effectively learning translation information. This challenge is inherent in the architecture of CNNs, which are typically more attuned to extracting hierarchical and spatial features from images rather than capturing translation variances.
- Another critical aspect observed is the need for enhanced training methods. Despite the progress made, the current accuracy levels of the model indicate that there is significant room for improvement. This suggests a potential exploration of advanced training techniques, hyperparameter tuning, or data augmentation strategies to achieve higher accuracy and reliability in the model's performance.

B. For Un-Supervised Model

- It has been observed that the unsupervised model underperforms, particularly in the context of image stitching tasks. The model, in its current form, struggles to effectively stitch multiple images together. This limitation is evident in the reduced accuracy and the inability to seamlessly merge multiple images into a single cohesive panorama.
- The unsupervised approach shows notable limitations when dealing with complex stitching scenarios that involve multiple images. These challenges may stem from the inherent complexities of unsupervised learning paradigms, where the model lacks explicit guidance or labels to learn from. The performance gap in handling multiple images suggests a need for further research and development, possibly exploring hybrid models that incorporate elements of supervised learning or more advanced unsupervised techniques to enhance the model's stitching capabilities.

V. Acknowledgment

The author would like to thank Prof. Nitin Sanket and the TA of this course RBE549- Computer Vision.

References

[1] RBE549 - Computer Vision Website**Link Link**