Cross-Domain Localization - Towards SLAM in Dynamic Conditions

Submitted By:
Vishal Pani (IIT2017066)
Arpit Mishra (IIT2017088)
Naman Tiwari(IIT2017034)

Project guide:
Dr. Rahul Kala

Allahabad, Uttar Pradesh, India { 211012}

Department of Information Technology

Indian Institute of Information Technology, Allahabad

Certificate

This is to certify that this is a bonafide record of the project presented by the student whose name is given below in partial fulfillment of the requirements for the degree of Bachelor of Technology in Information Technology.

Roll No	Names of Students
IIT2017034	Naman Tiwari
IIT2017066	Vishal Pani
IIT2017088	Arpit Mishra

Dr. Rahul Kala
(Project Guide)

Date - 20 May 2020

Content

1. Introduction

Visual recognition of places and place sequences in the field of robotic navigation has advanced rapidly in recent years. If a robot has visual data of a place in day lighting, the information is useless if the lighting conditions change. We tackle the problem specifically with navigation at night time by using a map of the place, made in day lighting.

2. Data set Collection

How we collected data :

1. **Camera used:** ZED Camera, 720p @15FPS
2. **Path:** CC-I -> CC-2 -> Main Gate -> CC-1
3. **Conditions:** Afternoon/Day and Night
4. **Image Count:** Around 6,000 LR Paired Images

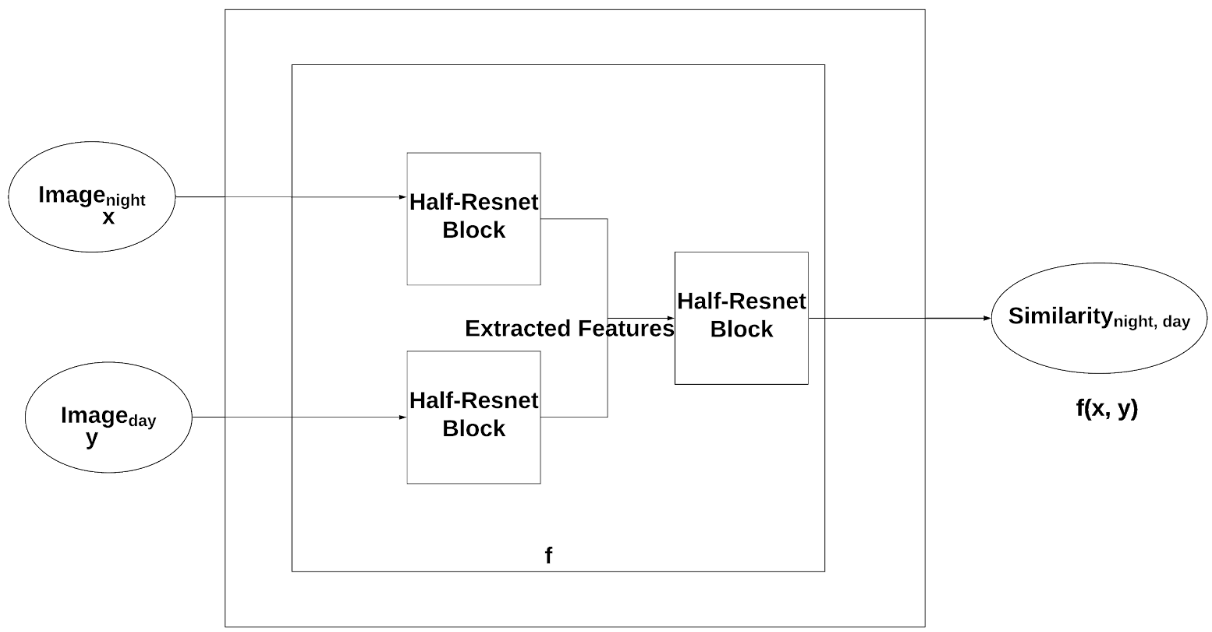
Sample Pairs:

3. YModel for Similarity Score

A ResNet model is divided into 2 halves. The architecture of the first half was copied. Now, the two first halves take one image each (Day and Night) and at the end their outputs are concatenated. These concatenated outputs are passed through the second half of the ResNet. We call this model **YModel**, owing to its shape. This model is used as a **binary classifier** between Night and Day images.

Block Diagram of YModel:



4. Base Algorithm

---

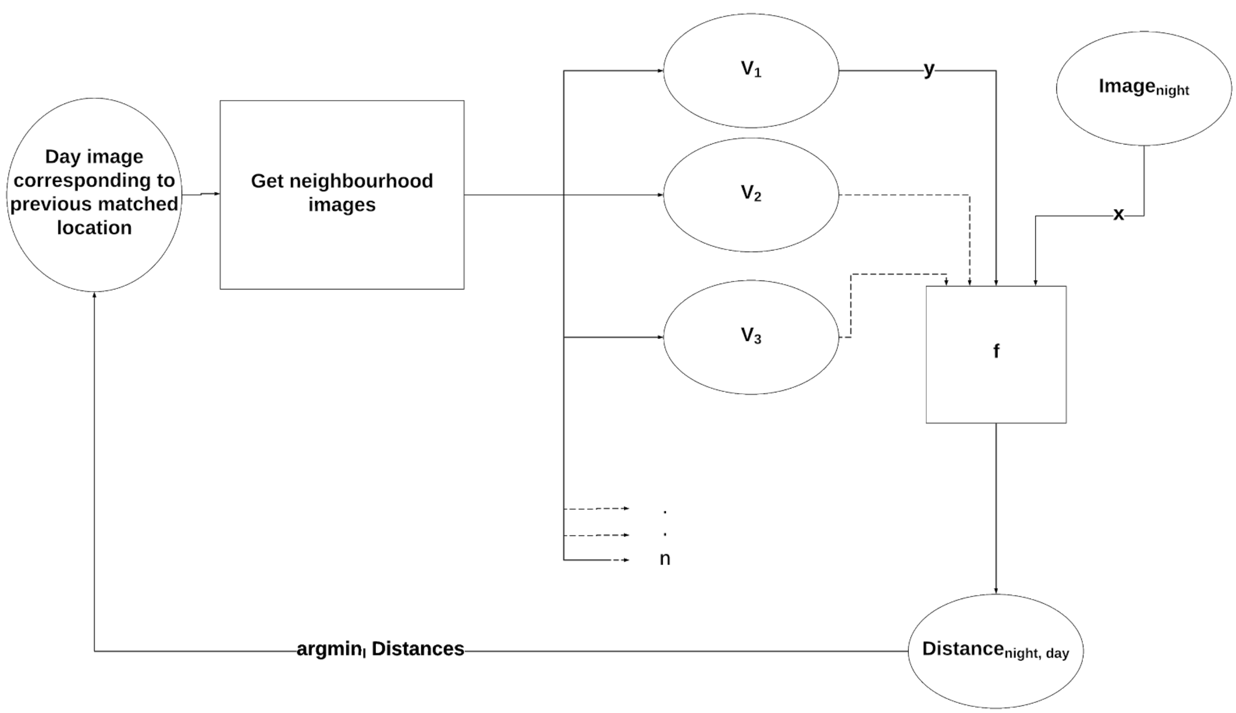
1. We take a long path in consideration. In our case, we took continuous images of the path in front of the academic buildings (CC1, Lecture theatre, CC2, and further).

2. For the day image graph, we took all the images of the path, divided them equally into successive sets of images, and selected a representative sequence of images for each vertex of the graph. (**Note:** sequence size < bin size)

3. Then we run a loop through all the night images. Initially, we know the day image vertex (and in effect, the real position) corresponding to the first night image.

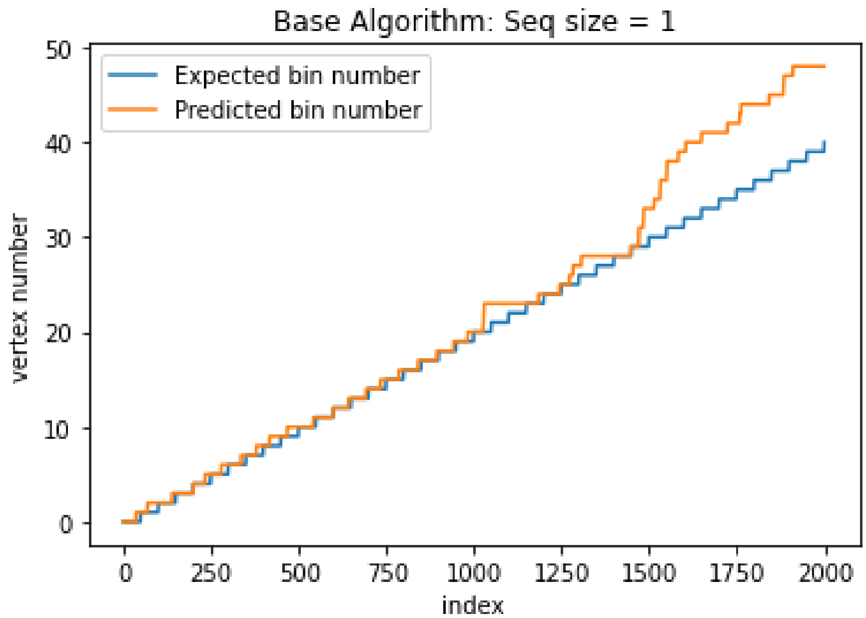
4. For every night image, we consider the previously matched day image vertex and the next vertices adjacent to it. If we get a higher similarity score with a new vertex, we update the current position.

5. If the real position differs from the predicted position, we add 1 to the total error.

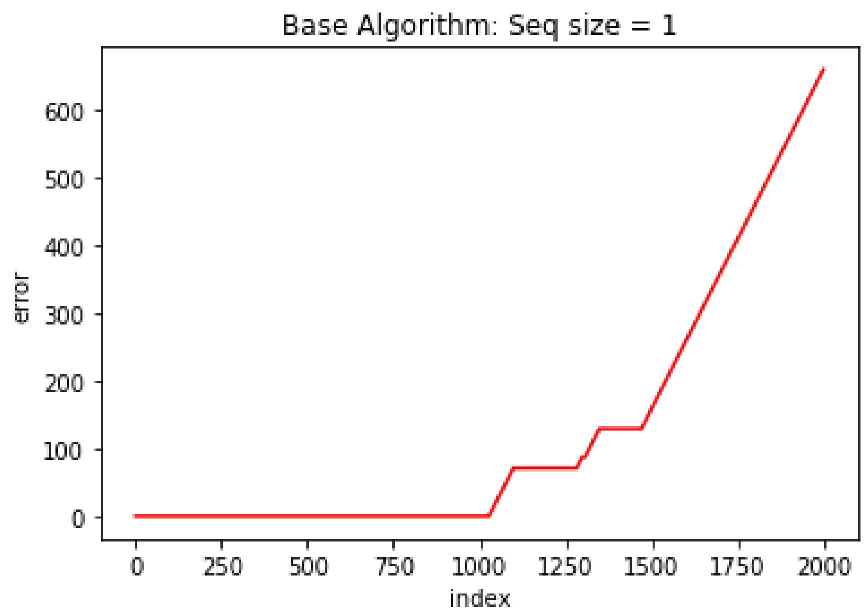


Flow Chart of the Algorithm:

4.1 Initial Results



Trajectory Graph with Sequence Size = 1



Error Graph with Sequence Size = 1

5. Sequence Similarity

The problem with using only a single pair for similarity is that it is very prone to error. This may occur due to various factors like introduction of variable elements into the image like cars, flashlights, etc. Therefore using a sequence of consecutive images will be more robust to such random elements. Thus we consider using sequences in our further study.

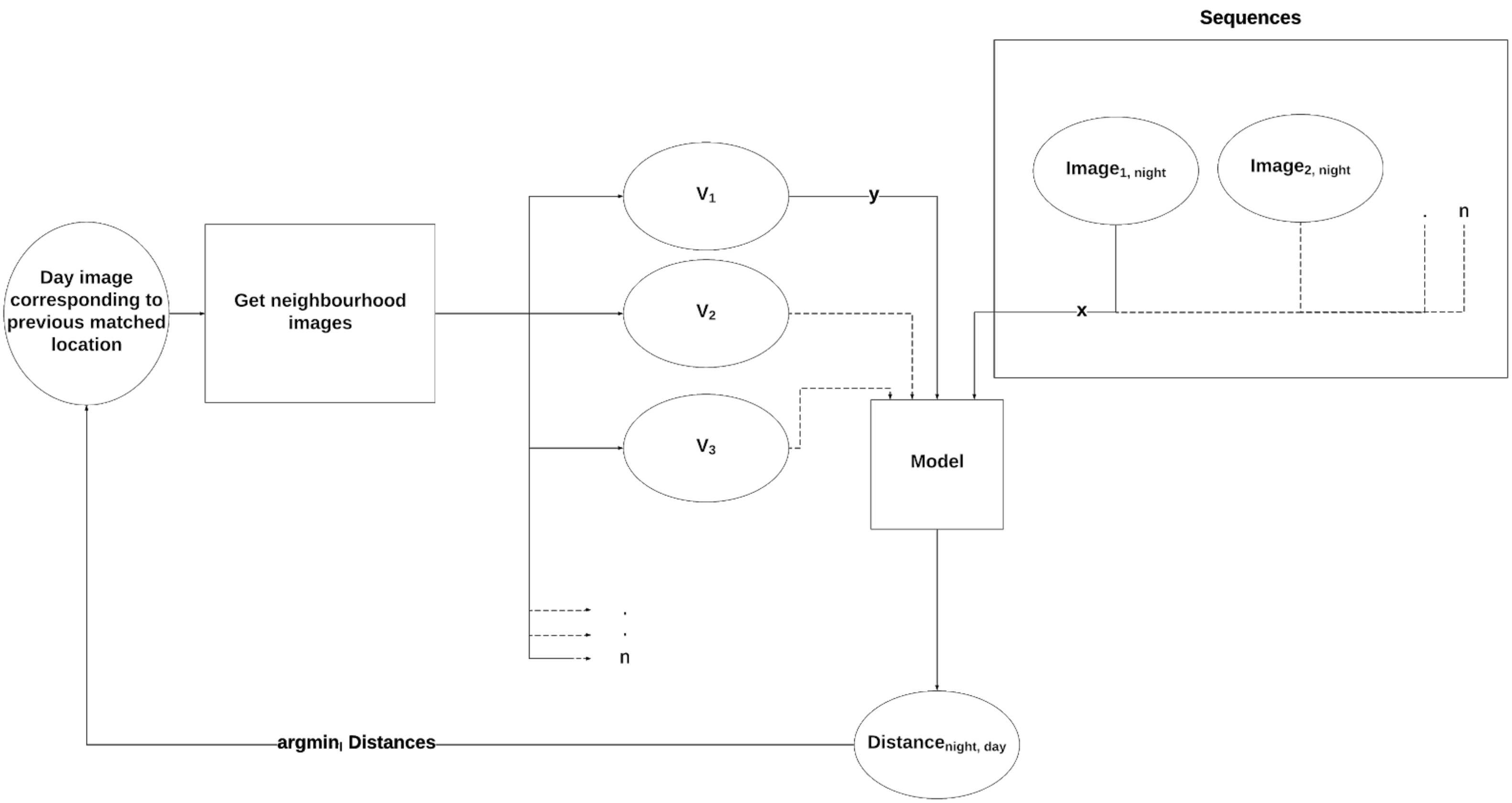
5.1 Sequence Similarity

$$\mathbf{Seq\_Similarity_n = \Sigma\ f(x_i,y_i)}$$

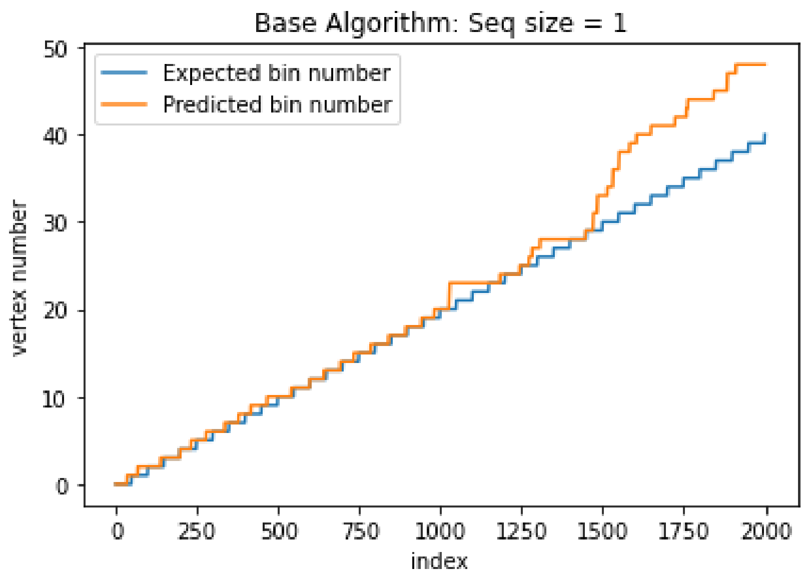
Where,

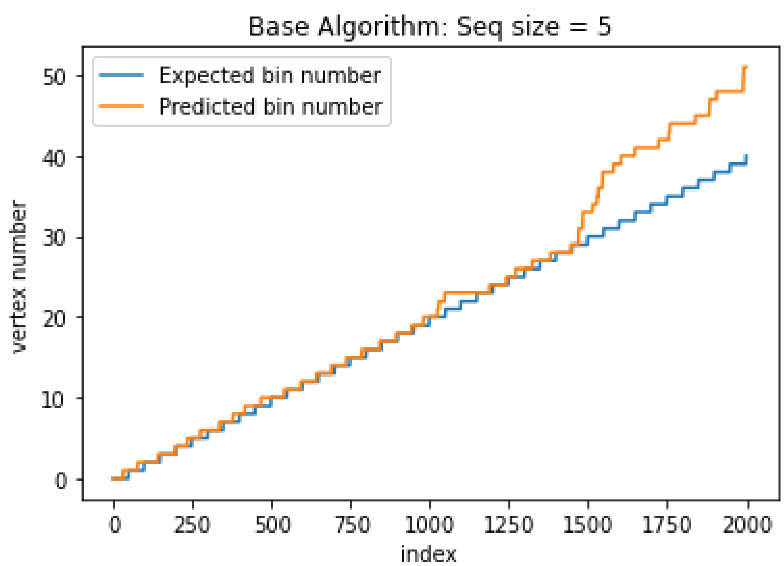
$\mathbf{x_i}$ is the i^{th} night image
$\mathbf{y_i}$ is the corresponding noon image
i ranges from **n** to **n - seq_size**
n is the current query night image
$\mathbf{f(x_i,y_i)}$ is the out of the YModel

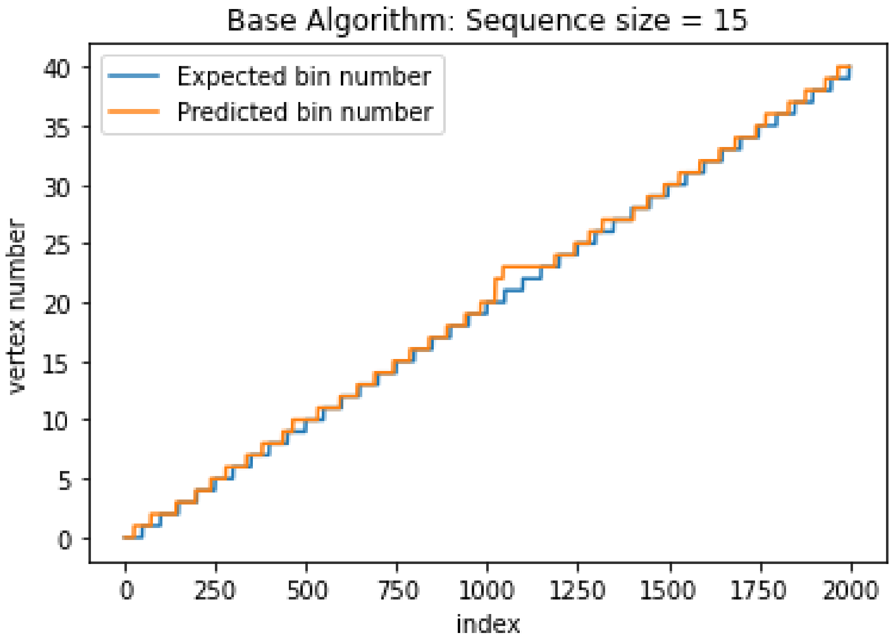
5.2 Flow Chart for Sequence Similarity Algorithm



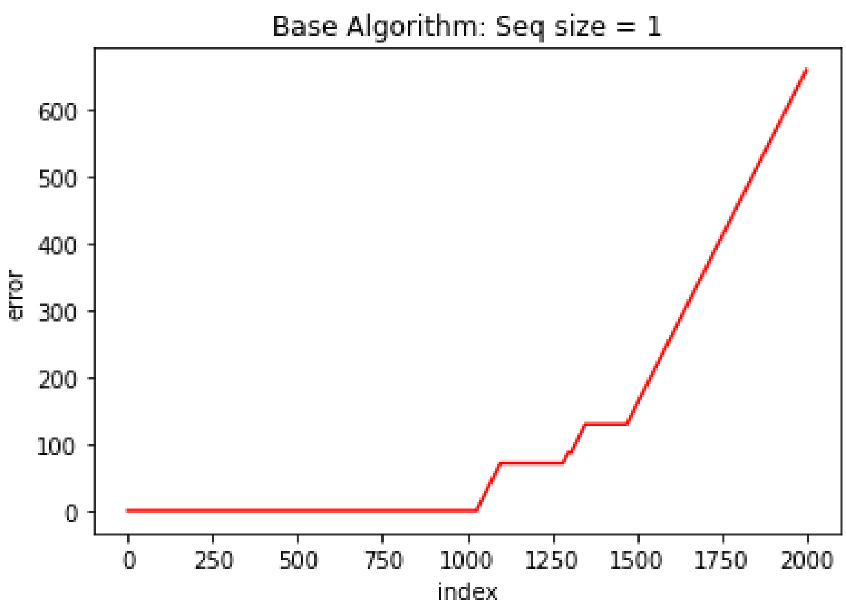
5.3 Trajectory Graphs with Sequence Size 1, 5, and 15

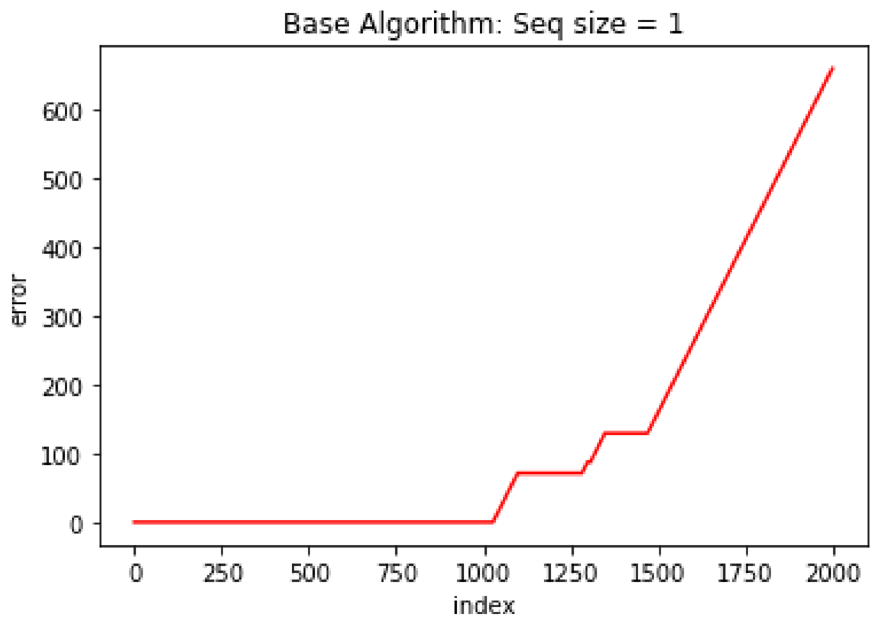


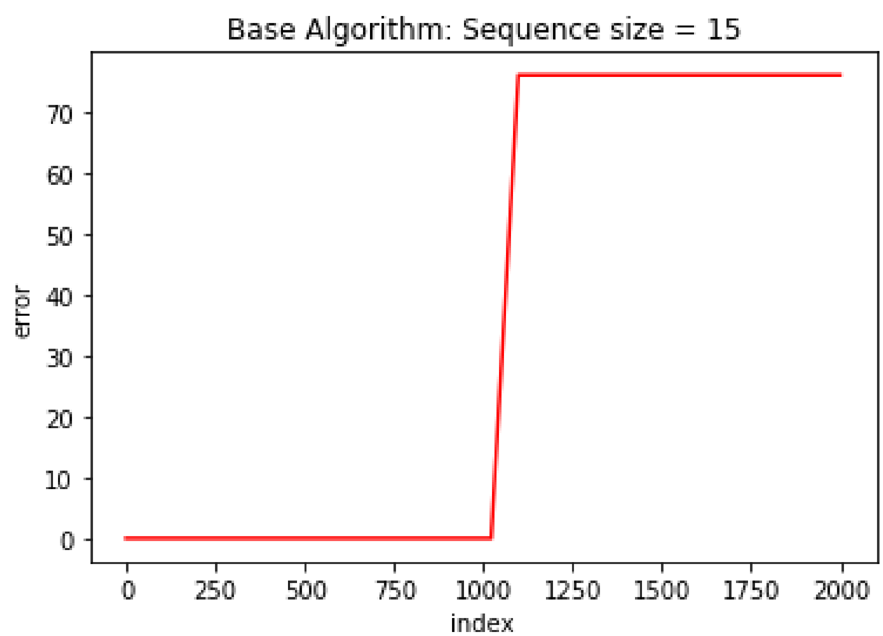




5.4 Error Graphs with Sequence Size 1, 5, and 15

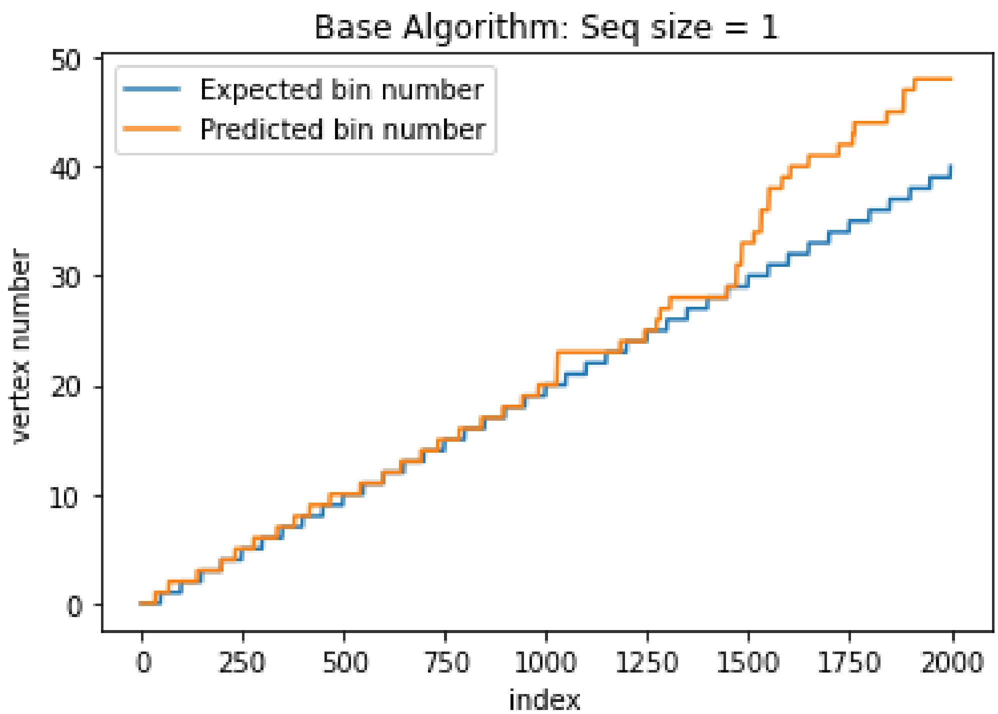



Base Algorithm: Seq size = 1
error
index
0
100
200
300
400
500
600
0
250
500
750
1000
1250
1500
1750
2000


Base Algorithm: Sequence size = 15
error
index
0
10
20
30
40
50
60
70
0
250
500
750
1000
1250
1500
1750
2000

6. Landmarks and Global Threshold

The idea is to allow re-localization in case the algorithm has gone too far off the correct vertex. The arrow in the diagram (on the right) marks the point where the predicted location (vertex) starts to move away from the correct location (vertex)



Landmarks are chosen by finding visually the most distinctive locations (maximum difference from the other vertices). Then for any night time image sequence we also compare it with all the landmarks and if it matches with the landmark (if sequence similarity of current night image sequence with landmark is above a threshold) then local search is done around the landmark. The **Structural Similarity Index (SSIM)** is used to find distinctive locations.

The Structural Similarity Index is given by the formula:

$$\mathrm{SSIM}(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

The SSIM method attempts to model the perceived change in the structural information of the image. There is a subtle difference between SSIM and MSE, but the results are dramatic.

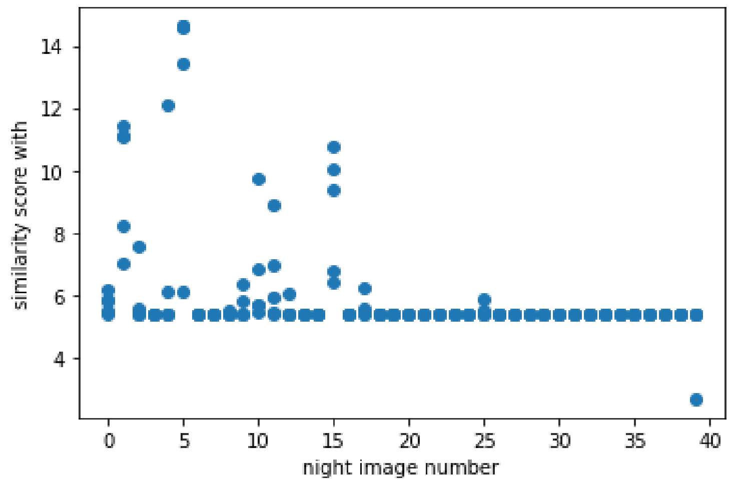
SSIM is used to compare two windows (i.e. small sub-samples) rather than the entire image as in MSE. Doing this leads to a more robust approach that is able to account for changes in the structure of the image, rather than just the perceived change.

The parameters to the equation include the (x, y) location of the N x N window in each image, the mean of the pixel intensities in the x and y direction, the variance of intensities in the x and y direction, along with the covariance.

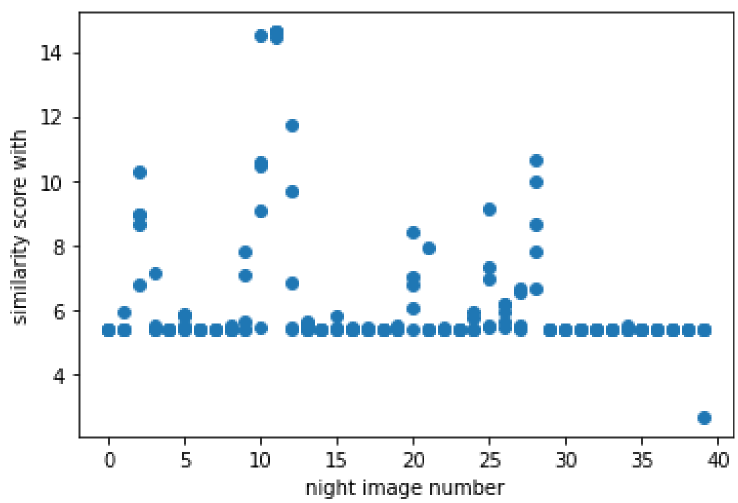
6.1 Finding the Threshold

The threshold for the landmark search was chosen to be 0.7 by analyzing the similarity score of each landmark with all the other bins. For bins corresponding to the landmark, the similarity score was greater than 0.7 and less than 0.7 in all other places. Therefore the max of similarity 1, similarity 2, and similarity from the global search were used to decide the current bin. Right now, global search just resets the bin number to the landmark bin, we can do a narrow search with this region (+/- 5 bins) to get a better restart location.

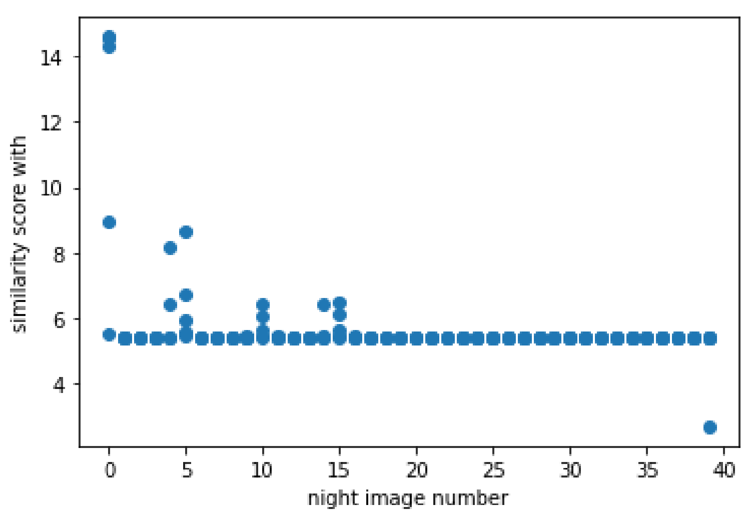
Landmark 5



Landmark 11

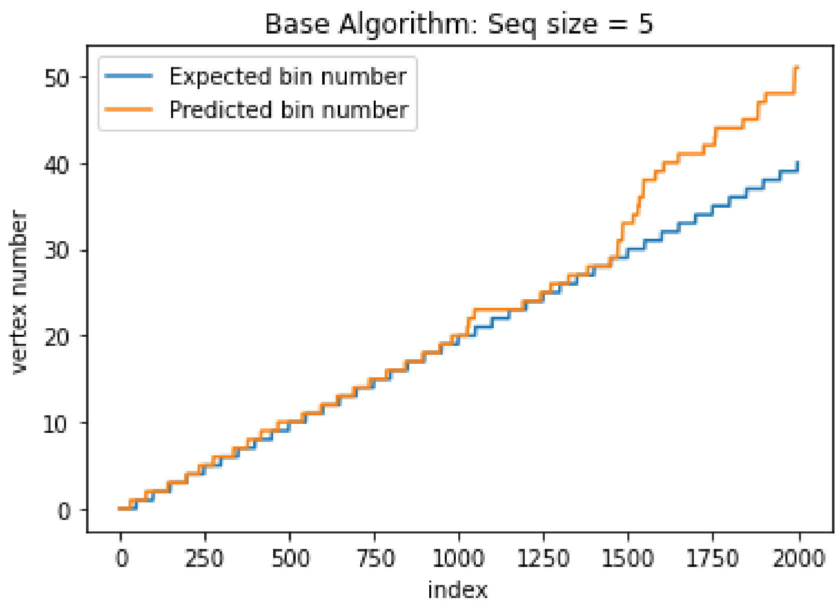


Landmark 0



6.2 Algorithm without Global Threshold

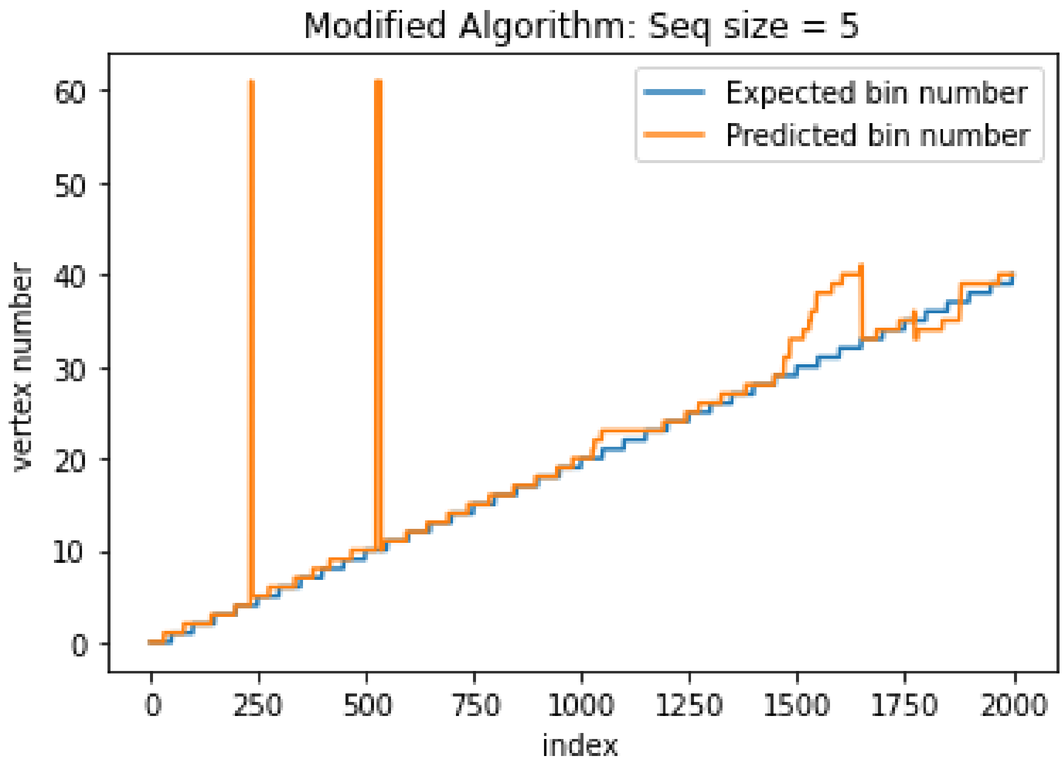
We ran the test algorithm with sequence size 5 without doing a landmark search. It is seen that the robot is lost at around the 1500 mark and never predicts the correct position again.



6.3 Algorithm with Global Threshold

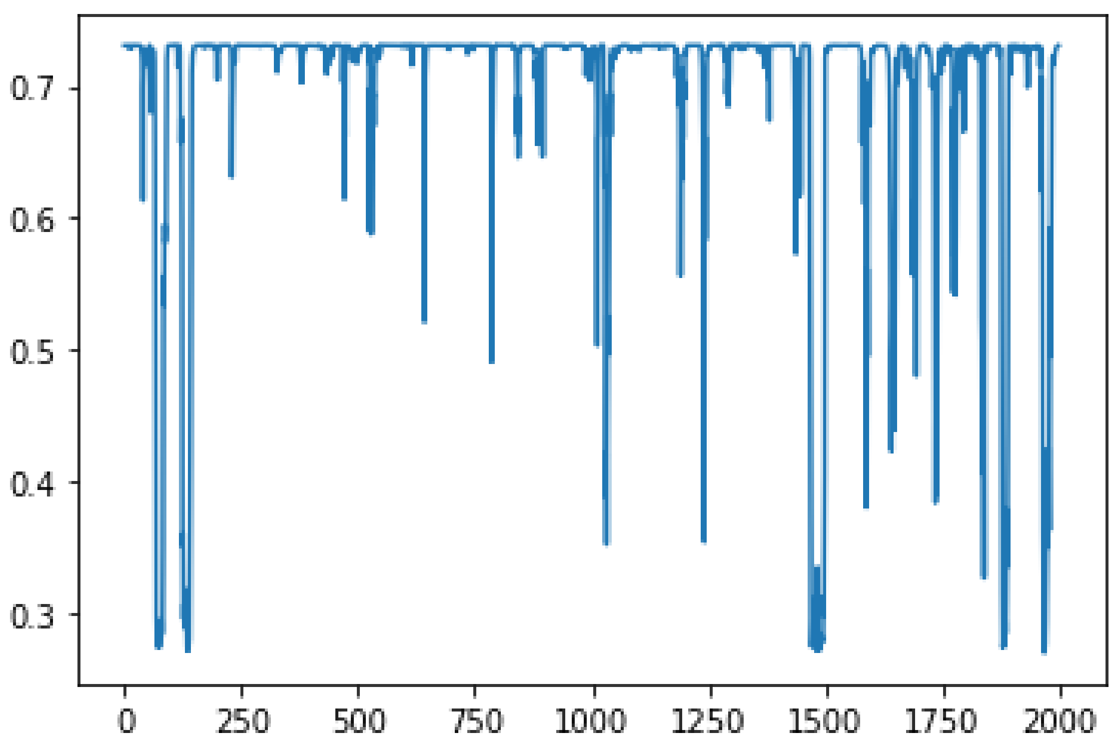
We ran the test algorithm with sequence size 5. It can be seen that after applying a global search through the landmarks, the robot which got lost priorly found the correct position at around the 1600 mark.

Note*: At around the time value of 250 and 500, global search resulted in the robot losing its correct position but it found its footing pretty soon after that, again due to global search. A larger sequence size solves this problem.*

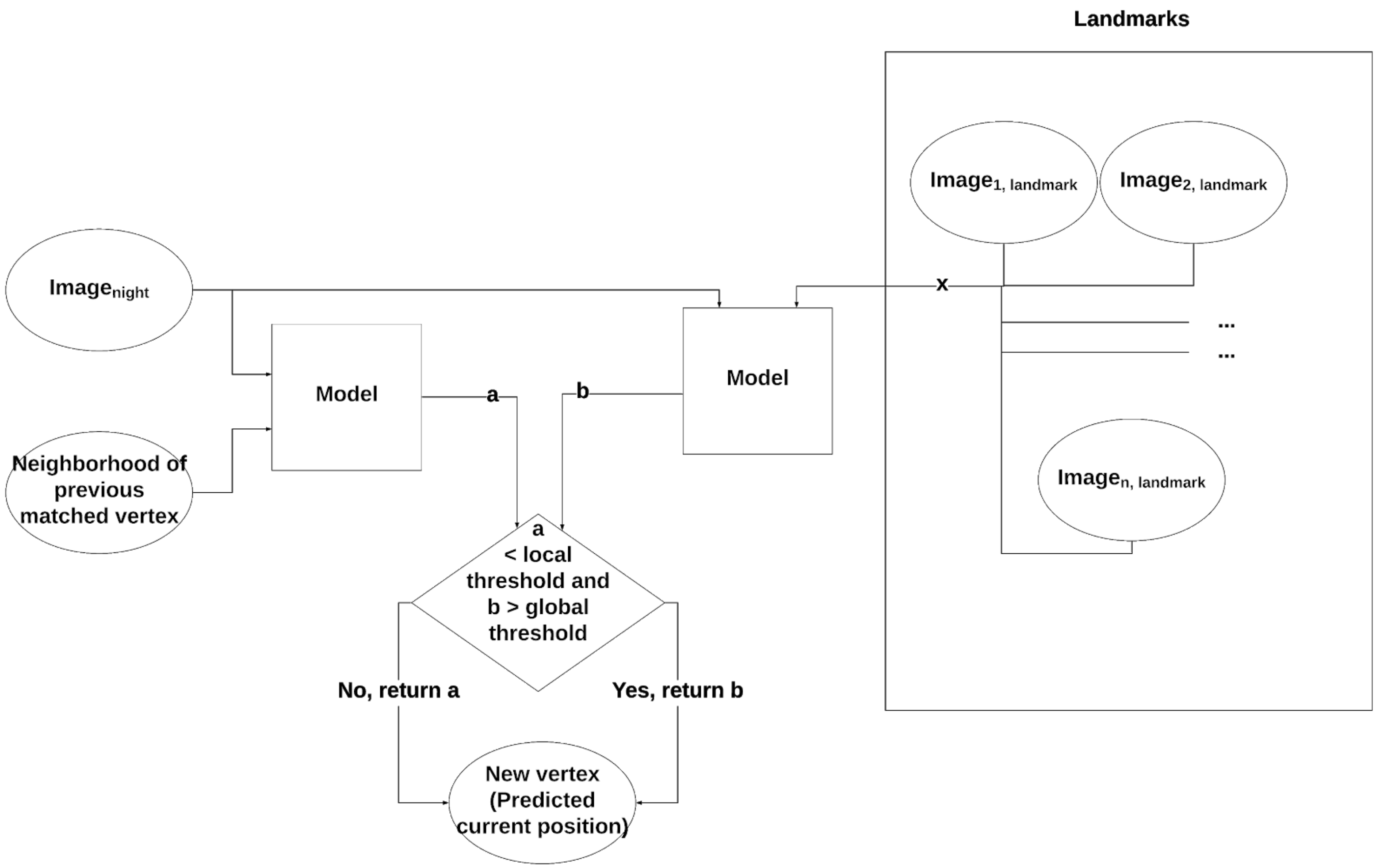

Modified Algorithm: Seq size = 5
Expected bin number
Predicted bin number
vertex number
index
0
10
20
30
40
50
60
0
250
500
750
1000
1250
1500
1750
2000

7. Local Threshold

1. For every night image and its corresponding vertex (the day image it is matched to), we plot the similarity scores.
2. The majority of night images show a similarity score of about 0.75. Hence we set 0.7 as a local threshold.
3. Now, only in case the average similarity score of a sequence of night images with a day image vertex falls short of this threshold do we then search globally for landmarks.
4. After applying the said local threshold, the total time of execution for all the test algorithms decreased to 1.5 hours from 6 hours.



8. Final Algorithm



9. Results

The error metric that is used is the number of erroneous hits.

9.1 Tables

Base Algorithm vs Modified Algorithm

SEQUENCE SIZE	BASE ALGORITHM	MODIFIED ALGORITHM (LOCAL + GLOBAL THRESHOLD)
1	658	685
5	599	**355**
15	76	76

The modified algorithm performs very well in the mid ranges of sequence size, but once the sequence size gets large there is hardly any difference. It should be noted that increasing the sequence size drastically increases the time of completion of the algorithm.

Base Algorithm vs Modified Algorithm (Seq Size = 15)

DATASET	BASE ALGORITHM	MODIFIED ALGORITHM (LOCAL + GLOBAL THRESHOLD)
Normal	76	76
Black Patch	72	72
Gaussian Noise	111	420

It can be seen that, while the modified algorithm is robust to modifications like adding black patches to the image, it is more sensitive to gaussian noise.

10. References

1. Capece, Nicola & Erra, Ugo & Scolamiero, Raffaele. (2017). Converting Night-Time Images to Day-Time Images through a Deep Learning Approach. 324-331. 10.1109/iV.2017.16.
2. M. J. Milford and G. F. Wyeth, "SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights," 2012 IEEE International Conference on Robotics and Automation, Saint Paul, MN, 2012, pp. 1643-1649, doi: 10.1109/ICRA.2012.6224623.
3. Zhou Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," in IEEE Transactions on Image Processing, vol. 13, no. 4, pp. 600-612, April 2004, doi: 10.1109/TIP.2003.819861.
4. https://www.pyimagesearch.com/2014/09/15/python-compare-two-images/
5. arXiv:1512.03385