OUTDOOR LOCALIZATION OF A ROBOTIC PLATFORM USING PANAROMIC IMAGES

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

GÜLÇİN CEREN EKİNCİ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
MECHANICAL ENGINEERING

JULY 2021

Approval of the thesis:

OUTDOOR LOCALIZATION OF A ROBOTIC PLATFORM USING PANAROMIC IMAGES

submitted by **GÜLÇİN CEREN EKİNCİ** in partial fulfillment of the requirements for the degree of **Master of Science** i**n Mechanical Engineering, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** ________________

Prof. Dr. M. A. Sahir Arıkan
Head of the Department, **Mechanical Engineering** ________________

Assoc. Prof. Dr. A. Buğra Koku
Supervisor, **Mechanical Engineering** ________________

Examining Committee Members:

Prof. Dr. Melik Dölen
Mechanical Engineering, METU ________________

Assoc. Prof. Dr. A. Buğra Koku
Mechanical Engineering, METU ________________

Assist. Prof. Dr. Ali Emre Turgut
Mechanical Engineering, METU ________________

Assist. Prof. Dr. Kutluk Bilge Arıkan
Mechanical Engineering, TED Üni. ________________

Assoc. Prof. Dr. Gökhan Bayar
Mechanical Engineering, Bülent Ecevit Üni. ________________

Date: 08.07.2021

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name Last name : Gülçin Ceren Ekinci

Signature :

ABSTRACT

OUTDOOR LOCALIZATION OF A ROBOTIC PLATFORM USING PANAROMIC IMAGES

Ekinci, Gülçin Ceren
Master of Science, Mechanical Engineering
Supervisor : Assoc. Prof. Dr. A. Buğra Koku

July 2021, 87 pages

Advances in robotics liberated robots from factory floors by the end of 20th century. Use of robots in our daily lives is only expected to increase in time. Robots, while relieving us from the burden of tedious, hard and dangerous tasks, most of them are still expected to be territorial, i.e. they will operate in a predefined or rather bounded environment. For enhanced performance, a robot should be familiar with its territory. In this work, use of skylines extracted from panoramic images is studied in order to provide localization in a square region. In order to achieve localization in a specific territory, firstly, a map is formed in that area using skylines which are extracted from captured photographs. Then, different images are obtained in the territory and skyline signals are extracted for the purpose of matching with the ones of map. As a result of matching process, localization is performed in that certain territory. In order to observe performance of localization algorithm, photographs are captured in diferent environments, seasons and times of day. Also, artifical occlusions are added to skyline signals in order to observe failure point of algorithm.

Keywords: Outdoor Localization, Panoramic Images, Skyline, Spherical Camera

ÖZ

ROBOTİK PLATFORMUN PANAROMİK FOTOĞRAFLAR KULLANILARAK DIŞ MEKAN LOKALİZASYONUNUN YAPILMASI

Ekinci, Gülçin Ceren
Yüksek Lisans, Makina Mühendisliği
Tez Yöneticisi: Doç. Dr. A. Buğra Koku

Temmuz 2021, 87 sayfa

Yirminci yüzyılın sonlarına doğru robotik çalışmalarındaki gelişmeler robotların fabrikalardan çıkıp günlük hayattaki kullanımlarına imkan tanımıştır. Robotların çoğu insanlar için sıkıcı, zor ve tehlikeli işleri gerçekleştirirken belli bir bölgede çalışmaya devam edeceklerdir. Gelişmiş bir performans için, bu robotların çalıştıkları bölgeye hakim olmaları gereklidir. Bu çalışmada, panoramik fotoğraflardan elde edilen gökyüzü çizgilerinin kare bir bölgede lokalizasyon için nasıl kullanılabileceğine dair metotlar geliştirilmiş olup deneyler yapılmıştır. Buna göre, lokalizasyon için öncelikle belirlenen alanda çekilen panoramik fotoğraflardan gökyüzü çizgileri elde edilip bunlar harita olarak adlandırılmıştır. Sonrasında bu alanda farklı noktalarda çekilen fotoğraflardan elde edilen gökyüzü çizgileri haritanınkiler ile karşılaştırılıp lokalizasyon işlemi gerçekleştirilmiştir. Algoritmanın performansının incelenmesi için, fotoğraflar farklı çevrelerde, mevsimlerde ve günün farklı zamanlarında çekilmiştir. Ayrıca, algoritmanın verdiği sonuçların ne zaman güvenilir olmayacağını incelemek için gökyüzü çizgilerine yapay engeller konulup lokalizasyon denenmiştir.

Anahtar Kelimeler: Dış Mekan Lokalizasyonu, Panoramik Fotoğraflar, Gökyüzü Çizgisi, Küresel Kamera

Kızlarım Elif ve Ayşenur'a

ACKNOWLEDGMENTS

I would like to express my gratitude to my advisor Assoc. Prof. Dr. A. Buğra Koku for his guidance throughout the whole four and a half years work.

I am forever indebted to my parents Galip Sağıroğlu and Servet Akyar for giving me the opportunities and support for my education through years. I thank my husband Mustafa Ekinci for his endless support and belief.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

Autonomous robots working in specific territories are getting more and more indispensable each day since they are ideal for dangerous, tedious and repetitive works. Territorial autonomous robots have been useful to humanity in a large amount of applications. Some examples could include space exploration [1], military applications [2] or more recently domestic use [3], highway driving [4] and agriculture applications [5].

Since the beginning of autonomous robot applications, localization has been an important challenge. Localization is defined as the process of determining robot's position with respect to an environment which the robot might or might not have prior knowledge. One of the first solutions to localization problem with the knowledge of environment is sonar applications [6]. It is considered for indoor applications by matching straight segments in the sonar range data with the existing room model. Without having prior knowledge, robot has to form its own map which can be metric or topological. Metric maps are used in case of accurate position estimation such as vehicle localization or obstacle avoidance. They represent the environment in a more structured way with details such as distances or measures. However, this kind of information requires high computational needs. Therefore, in the case of abstract representation of an environment, topological maps are used. They are represented by graphs where nodes and arcs are defined in order to describe distinctive places in the environment and relations between them. [7]

Whether metric or topological maps are used, environment must be formed using a localization method. Different approaches regarding localization have been presented over the years. One common method global positioning system (GPS) [8], is used for outdoor applications. Even though accurate position estimations can be obtained using GPS, it is known that it can be distorted from time to time [9]. Therefore, more reliable choices have been searched throughout years. As technology evolves and cameras become affordable, vision-based methods are being widely used in both indoor and outdoor applications. Extracting global [10] [11] or local features [12] [13] from images are two techniques. For global feature methods, only one component is obtained and matched, which results in less storage and CPU needs. However, methods of local features, such as SURF, SIFT and FAST, require more storage and CPU needs due to matching of a large amount of features. In spite of advantages in storage needs, it is stated that solutions based on global descriptors are not able to cope well with situations such as partial occlusions or camera rotations [7]. Even though local feature extraction seems to be robust, they can fail due to extreme seasonal changes [14].

Considering advantages and disadvantages of several localization methods; in this study, a computationally efficient and reliable method is investigated. Regarding downsides of previous methods; the method is aimed to be operational in spite of seasonal changes, occlusions and camera rotations. For the scope of this thesis, skylines obtained from panoramic images are used in order to form map and perform localization. Even though whole image is used for evaluations, it is only for obtaining 1-D skyline signals. Since matching process is performed with 1-D signals, the method is computationally efficient. Camera rotations are also ineffective for skyline signals as they are obtained from 360° spherical cameras.

CHAPTER 2

SCOPE OF THESIS

It is crucial for mobile robots to locate themselves within a territory in order to accomplish their mission. Accuracy of localization depends on requirements of operation and abilities of the robot. For example, a border tracking robot does not necessarily need localization with millimetric accuracy if it has obstacle avoidance and local target tracking algorithms which provide a smooth interaction with its close neighborhood. As an extension of this observation, this study focuses on robotic applications where the accuracy of localization is less important than whether a robot gets close enough to targets where local behaviors can complete the task at hand. Therefore, this work is similar to topological and qualitative navigation problems rather than metric map applications [24].

In this study, as explained before, skylines extracted from panoramic images are used for localization purposes. Skyline is defined as the line which separates sky and everything else. The reasons why skylines are used rely on the observation that skylines are very similar at points that are close to each other and skylines become less and less similar as reference points drift apart as shown in Figure 2.1.

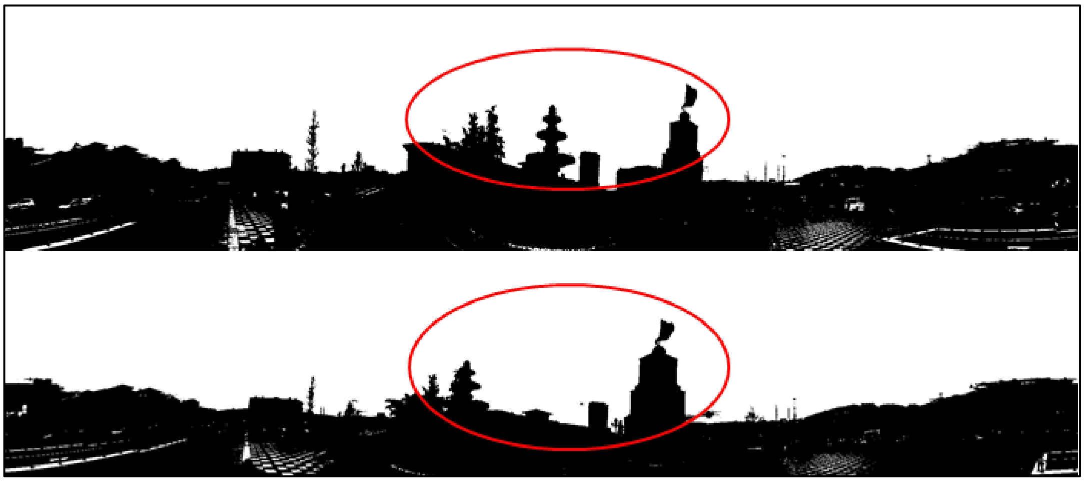


Figure 2.1. Change of Skyline as Real Location of Robot Changes

Skylines come less similar as the distance between the points they are extracted from increase. Therefore, skylines can be used as landmarks to tell the points that are close to each other apart from those that are not. Also it is worth noting that the skyline (which indeed is a 1D signal) depends on the extraction algorithm; however, this work does not focus on optimizing skyline extraction. The main purpose of this study is to continue on the work by Koku [23] where the author showed that skylines can be used as a landmark to encode points on straight or curved paths. This work extends the work of Koku by offering a localization method within polygonal regions. A collection of skylines extracted from images systematically taken within this region (a.k.a. the territory of the robot) is used as a map where skylines of the images taken at a later time are compared to the ones in this map to localize the robot. Therefore, localization in our perspective is not absolute like GPS; it is rather a relative localization with respect to a map which is formed beforehand.

CHAPTER 3

LITERATURE REVIEW

Several sky segmentation and matching methods have been studied for the purpose of localization. The first known article about skyline localization is [15]. Stein et al focuses on drop-off problem in which a robot is left at an unknown location and localization is performed using topographic map of an area which is also provided to the robot. Digital elevation models are used as topographic maps. For this study, panoramic horizon contours are obtained and these are computed offline from the map. Stein et al also would like to address occlusion problem by stating that portions of skyline are used for localization and this would allow algorithm to tolerate occlusions. However, no test data with occlusion is given. The algorithm does not perform correctly when the robot is put in a location where drastic changes occur in horizon line even with small changes of the viewpoint. Stein et al concluded that the system gives good performance as the view is further away. Therefore, this method might be usable for only mountainous area as similar to [16]. Saurer et al also represent the visible skyline by a set of contour words and use digital elevation models. They assume no camera roll. For large scale location recognition problem, success rates of 88% and 76% are achieved on two different sets. Those datasets are collected at different seasons, landscapes and altitudes.

Studies conducted only in urban areas are also present. In [17], Johns et al do not use panoramic cameras. The goal is to match a test view with other images taken from known positions. The challenge would be to identify specific buildings in the test view. Johns et al work on skyline contours which are described by a continuous piece-wise linear function. In order to obtain those, edge-detection is used. Matching process of skylines is performed by using segment similarity in the length and angle dimension contours. During this process, an amount of ambiguity might occur due to vertices since they can occur because of different reasons or noise. Also, rotation of camera is not considered during this study. To conclude, Johns et al use very few dataset of 30 images, all captured at the same time. Out of 30, 17 are used for test purposes and 2 of them are incorrect matches.

As another example for urban localization, in [18] Ramalingam et al use 300 fisheye images captured with lenses 183° field of view in a skyscraper-rich environment. The goal is to perform localization by matching skylines obtained from omni-directional images to skyline features from coarse 3D city models. Skylines are extracted from images using graph-cut method and then fisheye synthesis is performed. As a result of experiments, Ramalingam et al concluded that it is possible to outperform GPS based localization in urban areas. However, there might be possible problems due to insufficient 3D models and loss of skylines due to sun positions and weather conditions.

A different approach for skyline matching is represented in [19]. In this paper, experiments are also conducted in an urban place. However for this study, panoramic skylines are obtained from laser scans instead of images. Possible advantages of lasers could be obtaining skylines which are not affected from weather conditions and different sun positions. The beam of a laser scanner is reflected by the buildings and forms skyline. However, invalid measurements could also occur due to max-range measurements. In order to match skylines, longest common subsequence problem is considered. The solution is to test all possible subsequences from one skyline string, and search for correct matches in other strings. Therefore, matching would not be affected by rotation of skyline. The two experiments are conducted roughly half a year in between. Then, scans from those two different times are tried to match. Although scans are performed in different times, since there are no trees in the area, seasonal changes would not affect the matching procedure. However, changes due to construction occur in those different times, which seem to have minor effects on skyline matching.

In [20], Stone et al try a different approach for obtaining skyline. They use panoramic UV images, getting inspired by ant navigation. It is shown that UV images show better performance for skyline segmentation than visible images by studying datasets over a range of different locations and conditions. Stone et al did not consider seasonal changes or occlusions in the scope of this study. One example in which visible images give bad results is white minaret with white clouds. It is stated that UV based skyline is robust to changing light and weather conditions as similar to skylines obtained from laser scans in [19]. However, there are certain cases where UV images give bad performance. For example, when ground objects have particular kinds of metal roofing which has high UV reflectivity, it adversely affects sky segmentation. Nevertheless, Stone et al state that for the purpose of navigation, it is less important to find the skyline exactly than it is to find it consistently. The navigation potential is studied in an urban area of 2 km route on two different days with different weather conditions. Matching of skylines obtained from UV images from one day against other shows successful localization.

Another paper using UV images is [21]. For this study, Stone et al use UV sensitive fisheye lens camera mounted on top of a rough terrain platform. Experiments are conducted at three places; urban, forested and industrial areas. Aside from studying variable conditions, Stone et al also make tilt evaluation of camera by stating that when a camera is mounted on a platform performing on a rough terrain, it is most likely to experience yaw, roll and pitch change. In order to overcome changes due to tilt, spherical harmonic amplitudes are used for matching UV images.

A useful potential application of skyline localization is given in [22] where Ho et al represent the results of experiments conducted on freeways. Eleven video datasets are recorded via a single forward facing camera –not panoramic– mounted on the dash of a vehicle on the same route of 24.6 km. The conditions during data collection are varying in terms of weather, seasons and occlusions. However, seasonal changes are not expected to cause a big difference since videos are recorded during winter and autumn which means the trees are most likely in the same condition, without leaves. Another challenge is yaw, pitch, roll rotations. In order to obtain skyline, Ho et al make use of the fact that sky is always of a lighter intensity than the ground and as the image is scanned from top to bottom, the first pixel that has a blue intensity value higher than a threshold value is marked as the skyline pixel. This is also the method that is used in this thesis. As skylines are obtained, in order to match the lines, mean-sum-of-absolute-difference (MSAD) with shift correction is used. Again, for the purpose of this thesis, both MSAD and shift correction are used; however, the procedure is quite different than the authors'. As for the results of experiments, when the skylines obtained from ten dataset are compared with the reference, which is the first dataset, the best match is 82%. The weather and seasonal conditions are the same in both dataset. The worst is 27% because of truck occlusions on the freeway.

The predecessor of this thesis is [23]. Panoramic skylines are obtained using the same logic as explained for [22]. Then Koku checks for the similarity of skyline signals obtained in the same location during different times of the day and with different weather conditions. Before starting matching process, a crucial step which is shift correction is applied using the concept of cross-correlation. Then, MSAD is used for finding similarity values between two skyline signals. The experiments are conducted on a line basis. Koku forms the map by capturing photographs on a line and tries to match test photographs obtained from two sides of the line with map photographs. Despite getting good results from this study, Koku does not work with seasonal changes or occlusions and he only obtains data from two different environments. As an extent of his study, in this thesis, different environments, weather conditions, seasonal changes and occlusions are investigated. Also, the localization is performed on a 2-D area, not a line, resulting in prominent differences for the method of matching skyline signals.

CHAPTER 4

METHOD

Localization process starts by mapping of an area, meaning that capturing photographs at previously decided map points. In this study, different maps are formed at different locations and all of them are created as 10 meters x 10 meters square. The distance between map points is decided as 1 meter which gives 121 map points in one map. Test points are located between map points, that is, inside of 1m x 1m areas as shown in Figure 4.1.

Once all the images are obtained, the first thing to do is to extract skylines from those images. Then, a set of similarity values between the skyline obtained from the test image and all skylines in the map are calculated. These similarity values range from 100 to 0. As the value decreases, similarity between two skyline signals also decreases. Finally, the map point with the highest similarity value is found as the center of 2 meters x 2 meters localization square. It is stated that the test point is inside of 2 meters x 2 meters square with a center of “map point X.” For example, as illustrated in Figure 4.1, true location of test point is shown with blue circle. If the highest similarity value of that test point with the map is at point shown with green circle, then the localization is achieved. However, if the highest similarity value is at one of the map points shown with red circles, then localization is not achieved since 2 meters x 2 meters area around those map points does not include the test point.

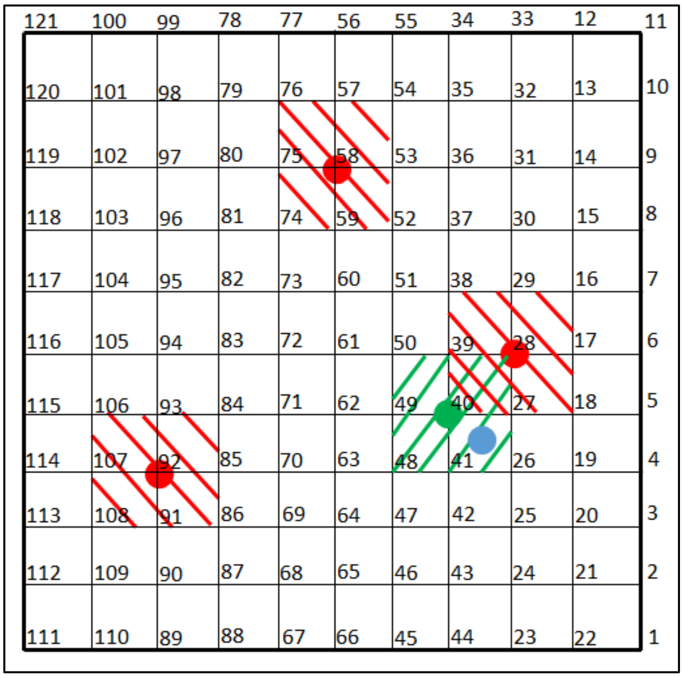


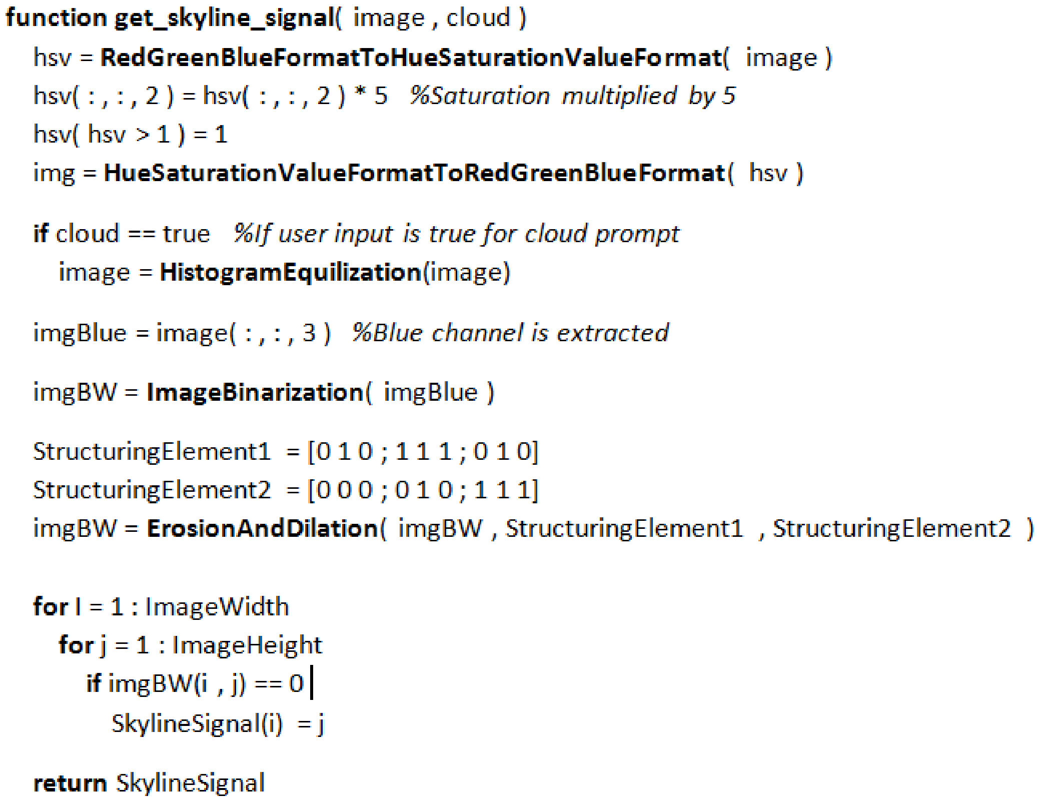
Figure 4.1. Map and Test Points With Correct and Incorrect Localization Points

Next subsections explain the process of extracting skylines in order to form map and test signals and matching of skylines extracted from new images taken at different times and weather conditions with the ones on the map.

4.1 Extraction of Skylines

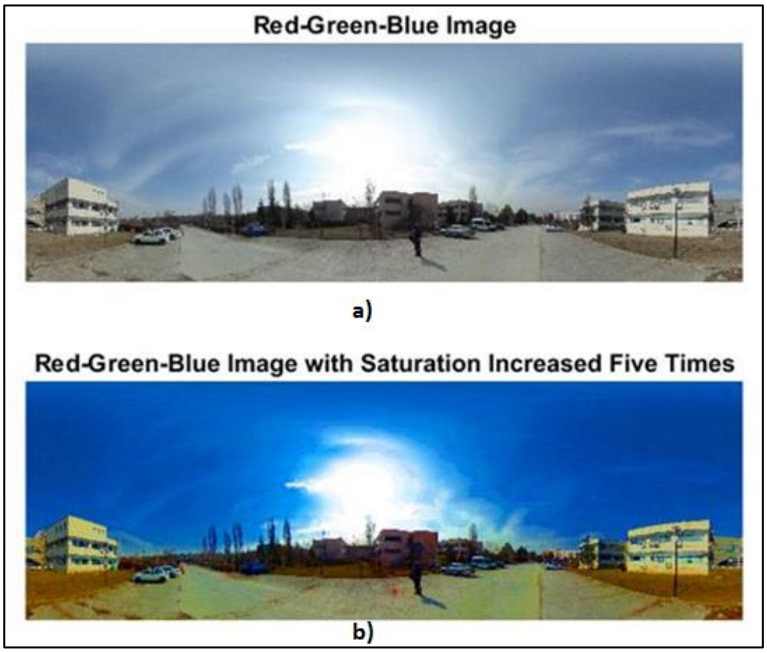
Skylines are extracted from images as 1-D signals. Width of these signals is equal to width of images which is 5376 pixels. In order to perform extraction, a number of transformations are applied to images. Pseudo code is given in Algorithm 1.

Algorithm 1: Skyline Extraction Process



```
function get_skyline_signal( image , cloud )
   hsv = RedGreenBlueFormatToHueSaturationValueFormat( image )
   hsv( : , : , 2 ) = hsv( : , : , 2 ) * 5  %Saturation multiplied by 5
   hsv( hsv > 1 ) = 1
   img = HueSaturationValueFormatToRedGreenBlueFormat( hsv )

   if cloud == true   %If user input is true for cloud prompt
      image = HistogramEquilization(image)

   imgBlue = image( : , : , 3 )  %Blue channel is extracted

   imgBW = ImageBinarization( imgBlue )

   StructuringElement1  = [0 1 0 ; 1 1 1 ; 0 1 0]
   StructuringElement2  = [0 0 0 ; 0 1 0 ; 1 1 1]
   imgBW = ErosionAndDilation( imgBW , StructuringElement1  , StructuringElement2  )

   for I = 1 : ImageWidth
      for j = 1 : ImageHeight
         if imgBW(i , j) == 0
            SkylineSignal(i)  = j

   return SkylineSignal
```

- Red-green-blue images are converted to hue-saturation-value images. Saturation values are multiplied by five. This process is helpful in distinguishing white buildings from sky and dark clouds from ground.
- Image is reconverted to red-green-blue format. The original and the latter are given in Figure 4.2.
- If image has dark clouds, then with the help of user input, histogram equalization is performed as shown in Figure 4.3. One important point is that histogram equalization must be performed when the image is not cropped.

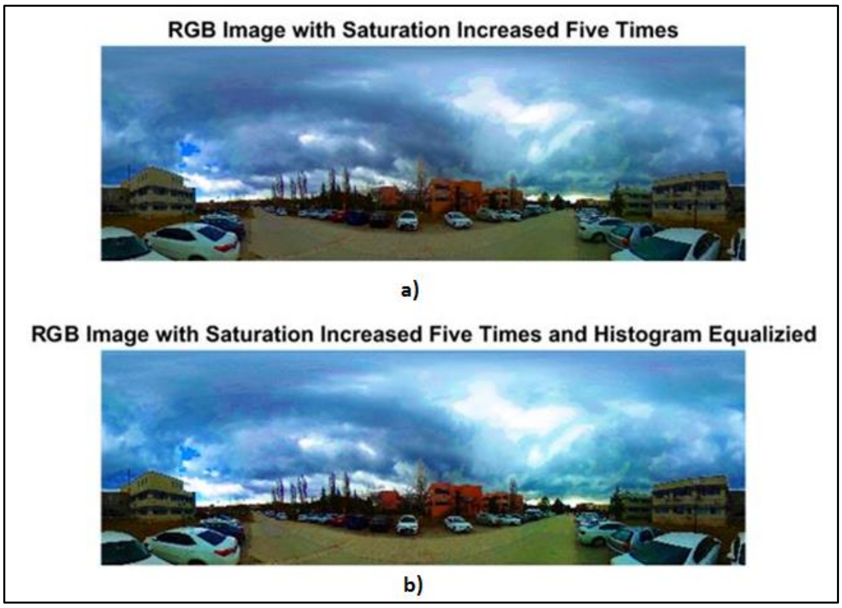


Red-Green-Blue Image

Red-Green-Blue Image with Saturation Increased Five Times

Figure 4.2. a) Red-Green-Blue Image

b) Red-Green-Blue Image with Saturation Increased Five Times



RGB Image with Saturation Increased Five Times

RGB Image with Saturation Increased Five Times and Histogram Equalizied

Figure 4.3. a) Red-Green-Blue Image of a Dark Cloudy Environment With Saturation Increased Five Times

b) Red-Green-Blue Image of a Dark Cloudy Environment With Saturation Increased Five Times and Histogram Equalized

- Grayscale image is obtained directly from the blue channel since blue is the most dominant color in sky. Binary image is formed using Otsu's thresholding method [25] as shown in Figure 4.4.

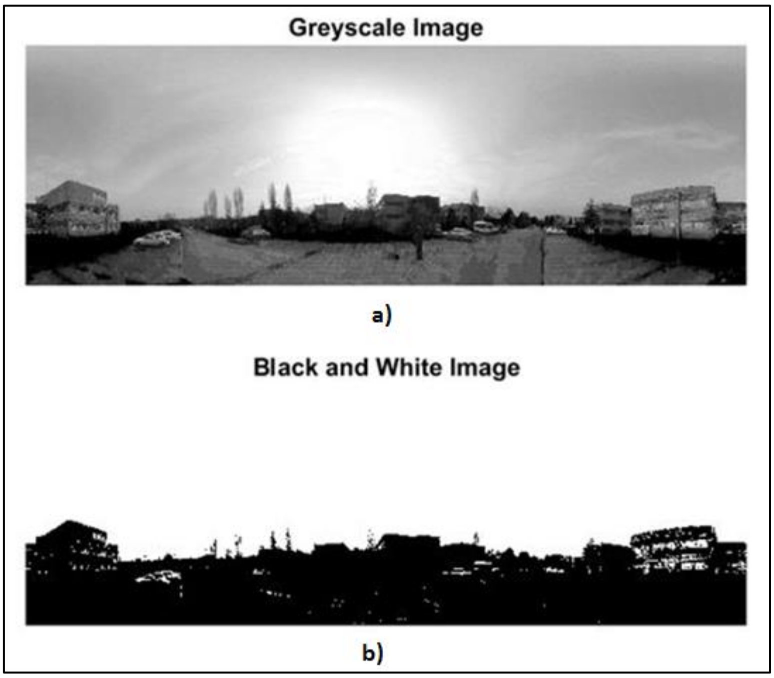


Figure 4.4. a) Greyscale Image b) Binary Image

- Erosion and dilation is applied to binary image in order to get rid of random disturbances which can distort the skyline.
- Binary image is scanned horizontally to search for the first ground pixel when scanned from top to bottom. Skyline signal is formed by using the location of those as given in Figure 4.5.

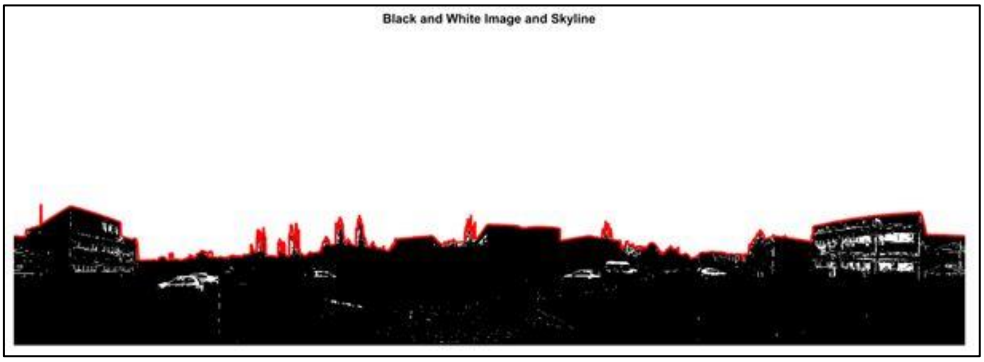
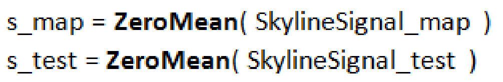


Figure 4.5. Skyline Signal Shown With Binary Image

- The final step is to make the mean of signal zero. Test and map skylines are obtained as shown in Algorithm 2. Signals are given in Figure 4.6.

Algorithm 2: Obtaining Map and Test Skyline Signals



```
s_map = ZeroMean( SkylineSignal_map )
s_test = ZeroMean( SkylineSignal_test )
```

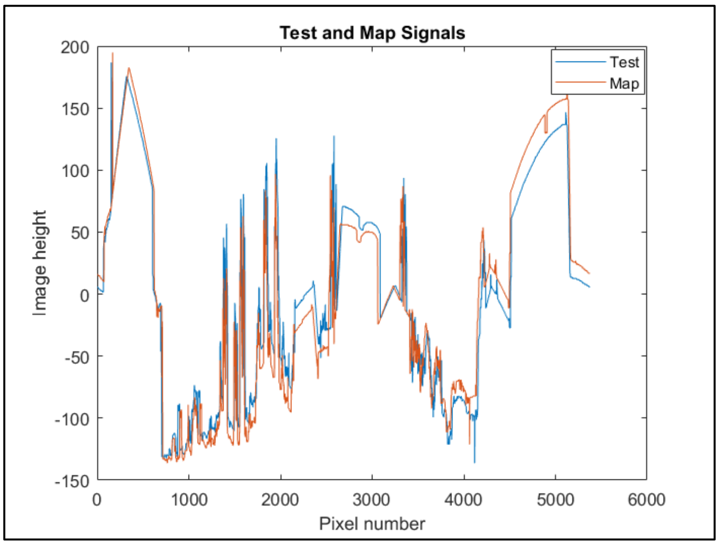
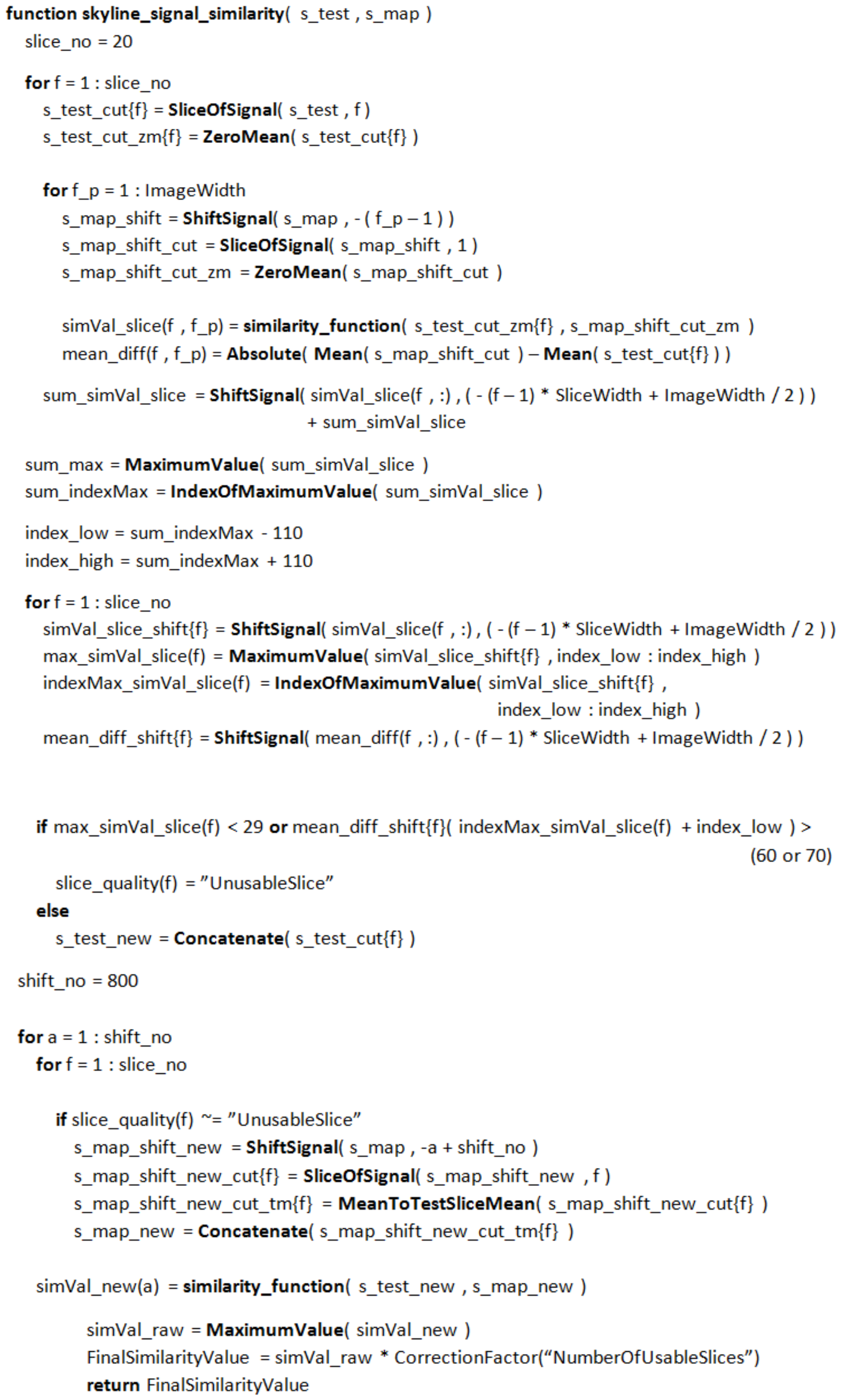


Figure 4.6. Test and Map Skyline Signals

4.2 Skyline Signal Similarity

As skylines from map and test photographs are extracted, the next step is to obtain similarity values between them in order to perform localization. For this purpose, a simple shift correction and similarity function as given in Algorithm 4 between two 1-D signals could be sufficient; however, this method would have given false results if the occlusion between signals is high due to inaccurate shift corrections. Therefore, as a new method, skyline signals are divided into pieces in order to search for occlusions and similarity values are found as the shift between them are corrected as explained in the subsection. Pseudo code is given in Algorithm 3.

Algorithm 3: Matching Algorithm for Skyline Signals



```
function skyline_signal_similarity( s_test , s_map )
  slice_no = 20

  for f = 1 : slice_no
    s_test_cut{f} = SliceOfSignal( s_test , f )
    s_test_cut_zm{f} = ZeroMean( s_test_cut{f} )

    for f_p = 1 : ImageWidth
      s_map_shift = ShiftSignal( s_map , - ( f_p – 1 ) )
      s_map_shift_cut = SliceOfSignal( s_map_shift , 1 )
      s_map_shift_cut_zm = ZeroMean( s_map_shift_cut )

      simVal_slice(f , f_p) = similarity_function( s_test_cut_zm{f} , s_map_shift_cut_zm )
      mean_diff(f , f_p) = Absolute( Mean( s_map_shift_cut ) – Mean( s_test_cut{f} ) )

    sum_simVal_slice = ShiftSignal( simVal_slice(f , :) , ( - (f – 1) * SliceWidth + ImageWidth / 2 ) )
                                              + sum_simVal_slice

  sum_max = MaximumValue( sum_simVal_slice )
  sum_indexMax = IndexOfMaximumValue( sum_simVal_slice )

  index_low = sum_indexMax - 110
  index_high = sum_indexMax + 110

  for f = 1 : slice_no
    simVal_slice_shift{f} = ShiftSignal( simVal_slice(f , :) , ( - (f – 1) * SliceWidth + ImageWidth / 2 ) )
    max_simVal_slice(f) = MaximumValue( simVal_slice_shift{f} , index_low : index_high )
    indexMax_simVal_slice(f) = IndexOfMaximumValue( simVal_slice_shift{f} ,
                                                       index_low : index_high )
    mean_diff_shift{f} = ShiftSignal( mean_diff(f , :) , ( - (f – 1) * SliceWidth + ImageWidth / 2 ) )


   if max_simVal_slice(f) < 29 or mean_diff_shift{f}( indexMax_simVal_slice(f) + index_low ) >
                                                                                   (60 or 70)
     slice_quality(f) = "UnusableSlice"
   else
     s_test_new = Concatenate( s_test_cut{f} )

 shift_no = 800

 for a = 1 : shift_no
   for f = 1 : slice_no

     if slice_quality(f) ~= "UnusableSlice"
       s_map_shift_new = ShiftSignal( s_map , -a + shift_no )
       s_map_shift_new_cut{f} = SliceOfSignal( s_map_shift_new , f )
       s_map_shift_new_cut_tm{f} = MeanToTestSliceMean( s_map_shift_new_cut{f} )
       s_map_new = Concatenate( s_map_shift_new_cut_tm{f} )

   simVal_new(a) = similarity_function( s_test_new , s_map_new )

        simVal_raw = MaximumValue( simVal_new )
        FinalSimilarityValue = simVal_raw * CorrectionFactor("NumberOfUsableSlices")
        return FinalSimilarityValue
```

Algorithm 4: Basic Similarity Function for Skyline Signals

```
function similarity_function( s1 , s2 )
    area_s1 = AbsoluteAreaUnderSignal( s1 )
    area_diff = AbsoluteAreaUnderSignal( s2 – s1 )
    simVal = MaximumValue( 0 , 100 – area_diff / area_s1 * 100 )
    return simVal
```

- Test signal is cut to 20 slices as shown in Figure 4.7.

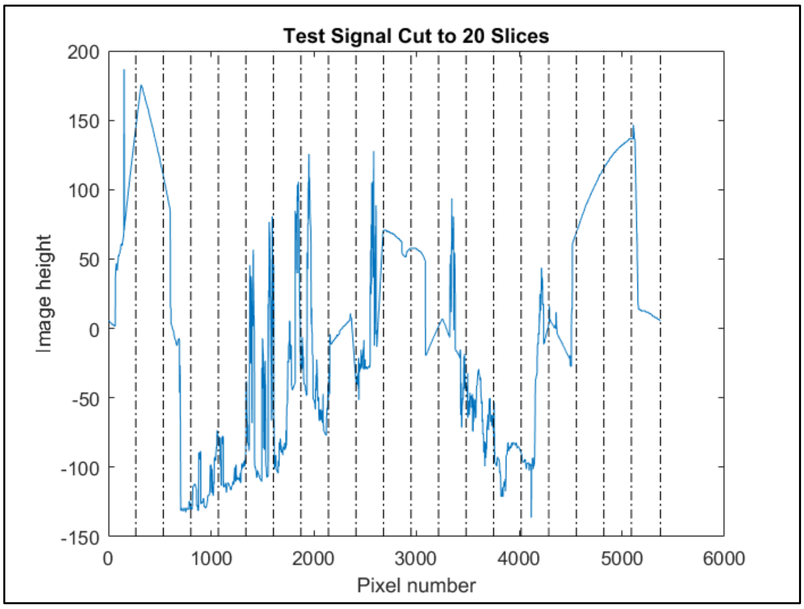


Figure 4.7. Test Signal Cut to Twenty Slices

- Map signal is shifted pixel by pixel along the width of image horizontally as shown in Figure 4.8. At each shift, the first slice of map signal is compared to the slice of test signal in zero mean versions as shown in Figure 4.9. Then, a similarity value is calculated using a similarity function as given in Algorithm 4. The difference between mean of a slice of map signal and mean of a slice of test signal is also saved.

- As a result of shifting and comparing signals, 20 similarity graphs are formed as shown in Figure 4.10. Each of them has a width equal to the width of image meaning that map signal is shifted through the whole image width. For highly similar two images, it is expected to have noticeable peaks in each graph. The location of these peaks should also be in alliance with each other. The distance between the location of maximum values and the starting point of slices as shown in Figure 4.7 should be close to each other. These starting points are 0, 268, 536, 804, 1072, 1340, 1608, 1876, 2144, 2412, 2680, 2948, 3216, 3484, 3752, 4020, 4288, 4556, 4824, and 5092. For the example given in Figure 4.10, shifts of the slices are -15, -19, -14, 2, -19, 2, 5, 7, 11, 22, 26, 27, 22, 15, 11, 14, 10, 11, 29, and -12. Although similarity values are also important, these values show that the two images are very much similar since the shift values are close to each other meaning that drifting of landmarks in skylines does not occur in a noticeable manner.

- After evaluating all test signal slices, a connection between similarity graphs should be established in order to estimate the original shift between test and map signals. If the map and test signals are both shifted with the same value, the location of maximum values at Figure 4.10 would also shift with that value. Using this concept, firstly each graph is shifted "f*slice_width" (f=0,1,..,19) to the left as shown Figure 4.11. It means that for every test signal slice comparison, both map and test signals are shifted so that always the first slice of test signal is being compared.
- Since the amount of shifting in Figure 4.11 might be confusing while finding the original shift value because of circularity of image, all graphs are now shifted to the middle "ImageWidth/2" as shown in Figure 4.12.

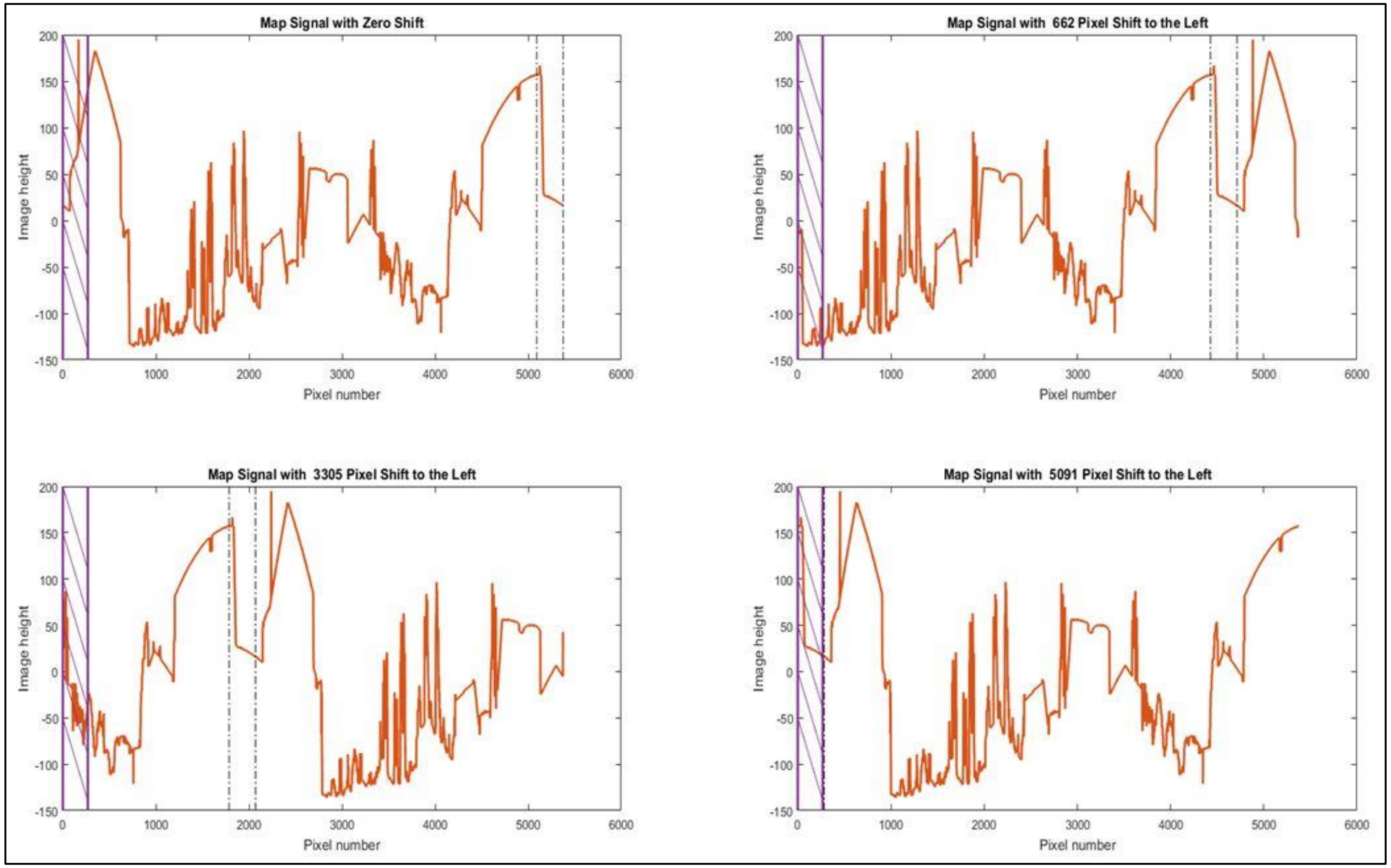


Figure 4.8. The most suitable slice to the last slice of test signal is shown with black lines. At each shift the first slice shown with purple lines is compared with test signal slice

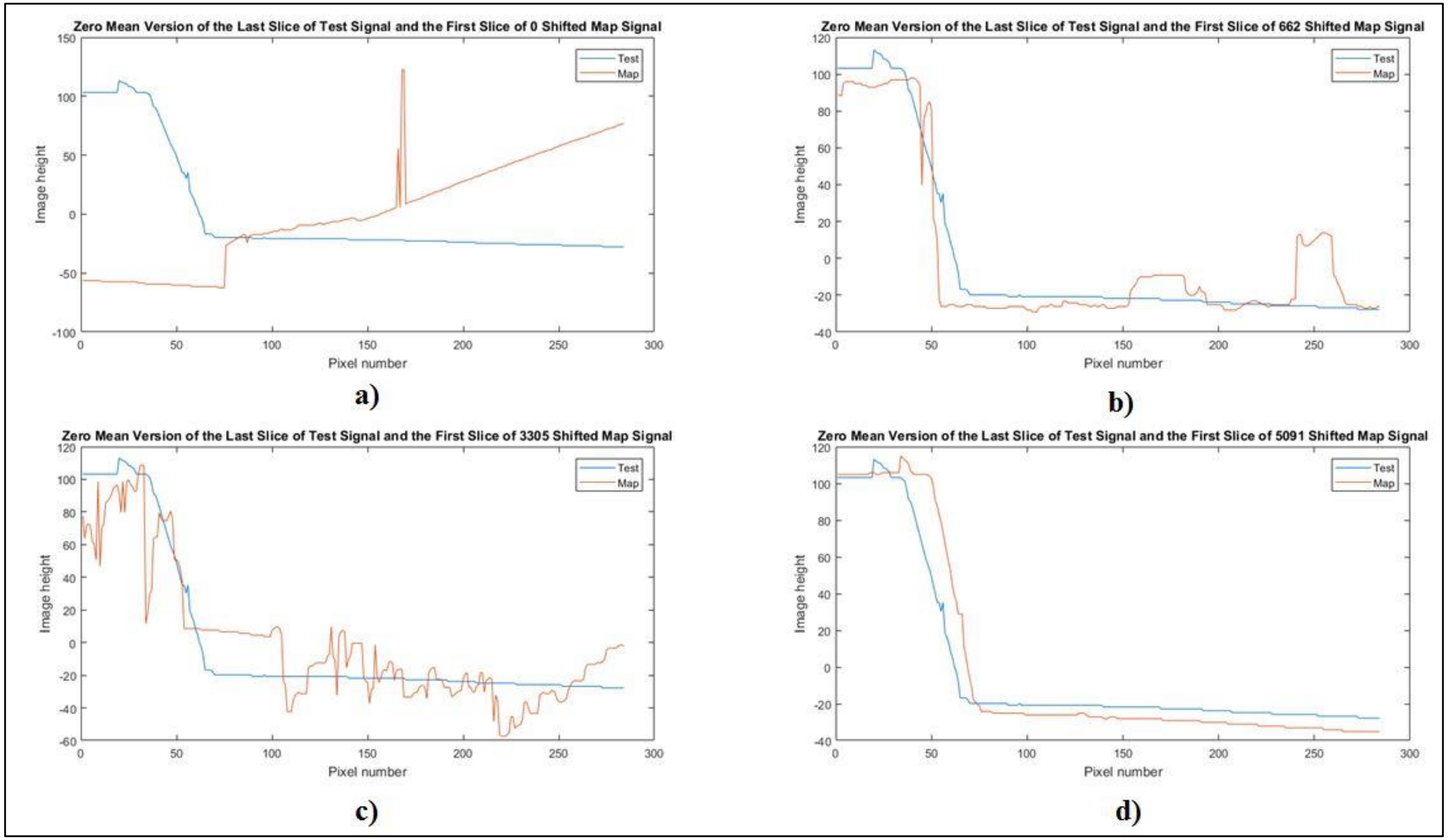


Figure 4.9. Comparison of the Last Slice of Test Signal with Slices from Map Signal Shifted Certain Amounts. The most suitable shift is given in (d). (b) and (c) also give relatively high similarity values since the features are alike

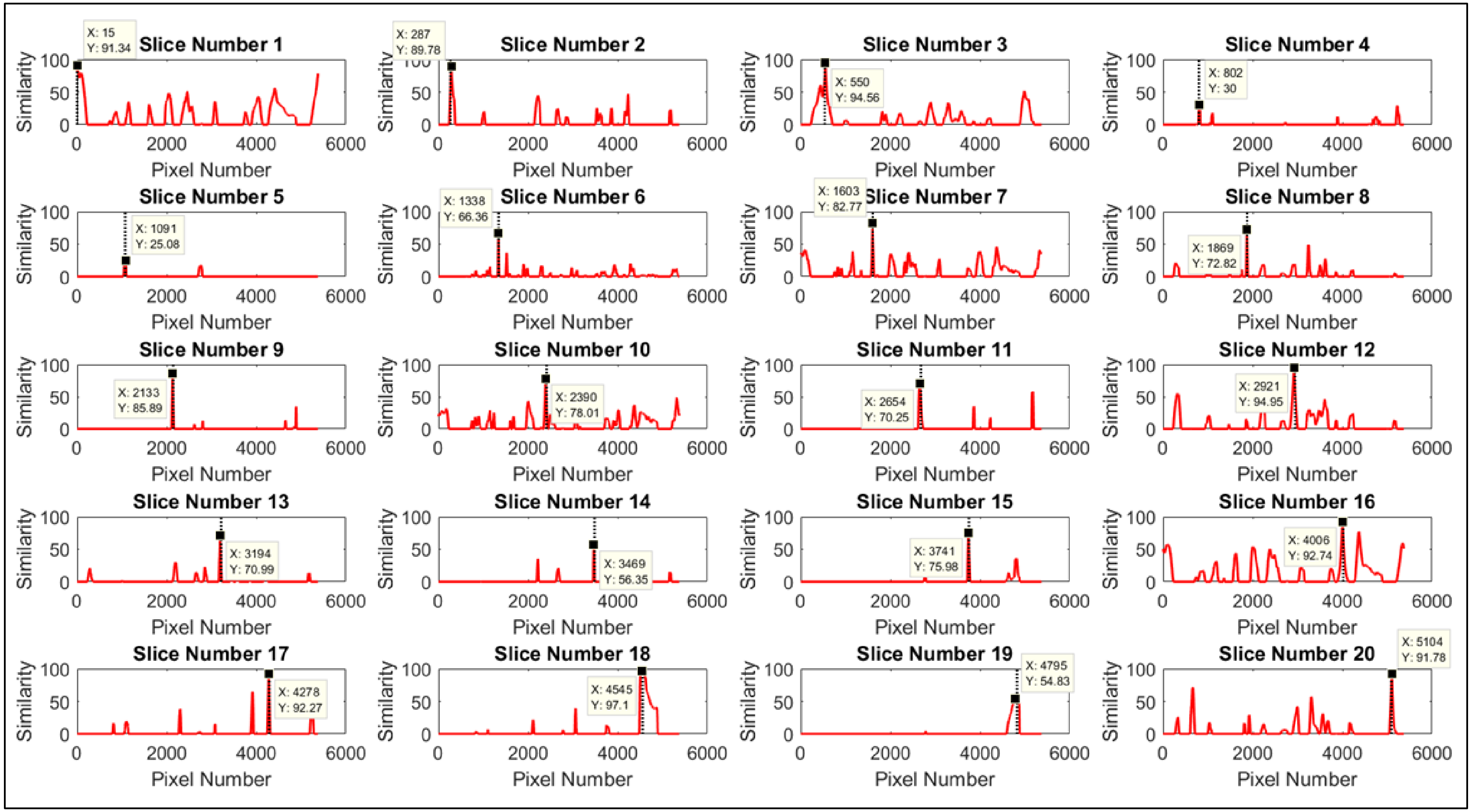


Figure 4.10. Similarity Graphs of Each 20 Test Signal Slice With the Map Signal. The peaks are very close to starting points of each slice as shown in Figure 4.7. The starting points are 0, 268, 536, 804, 1072, 1340, 1608, 1876, 2144, 2412, 2680, 2948, 3216, 3484, 3752, 4020, 4288, 4556, 4824, and 5092

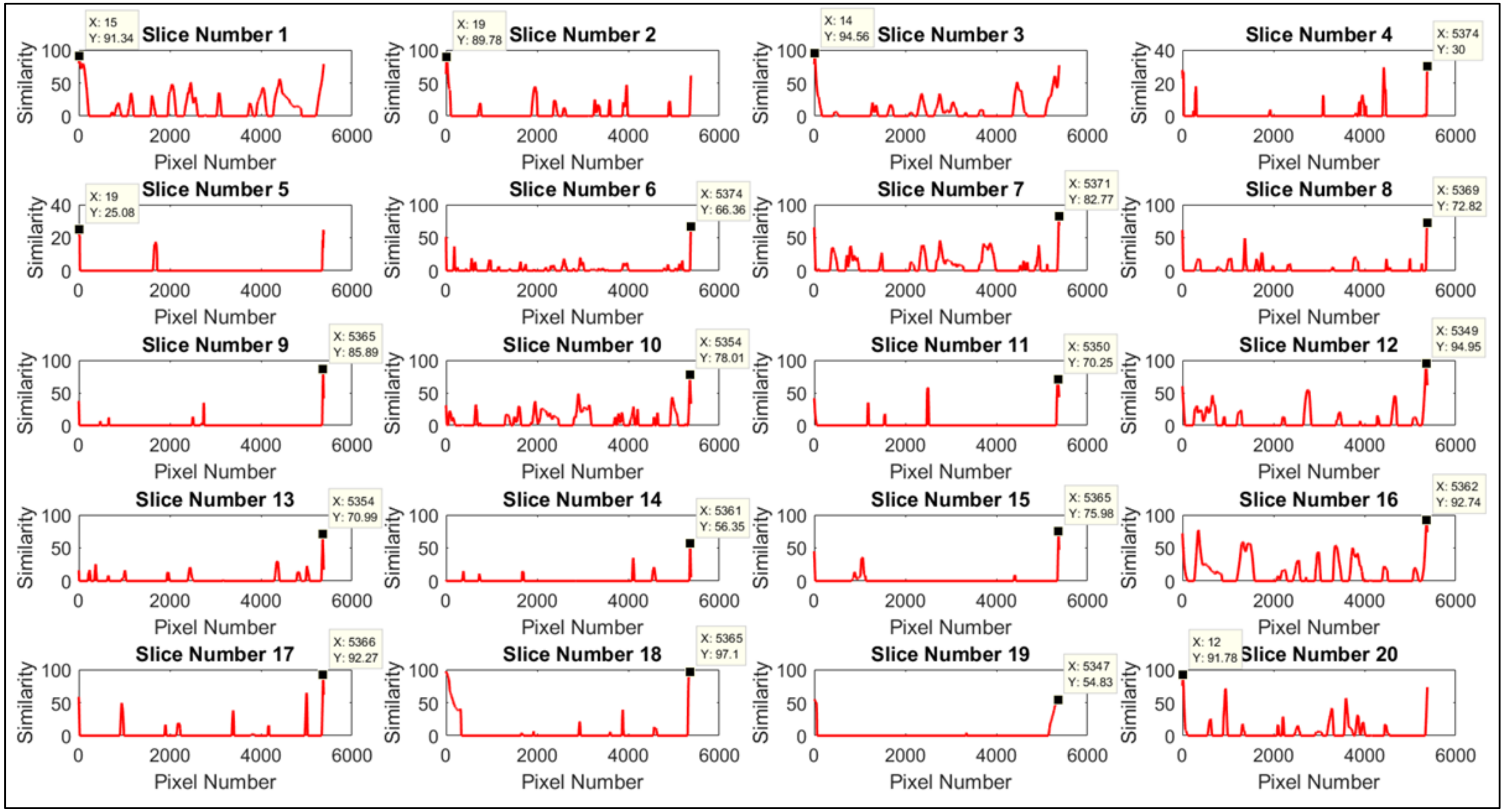


Figure 4.11. Each graph is shifted to the left by an amount of f*slice_width. This value equal to the starting points of each slices as explained in Figure 4.10.

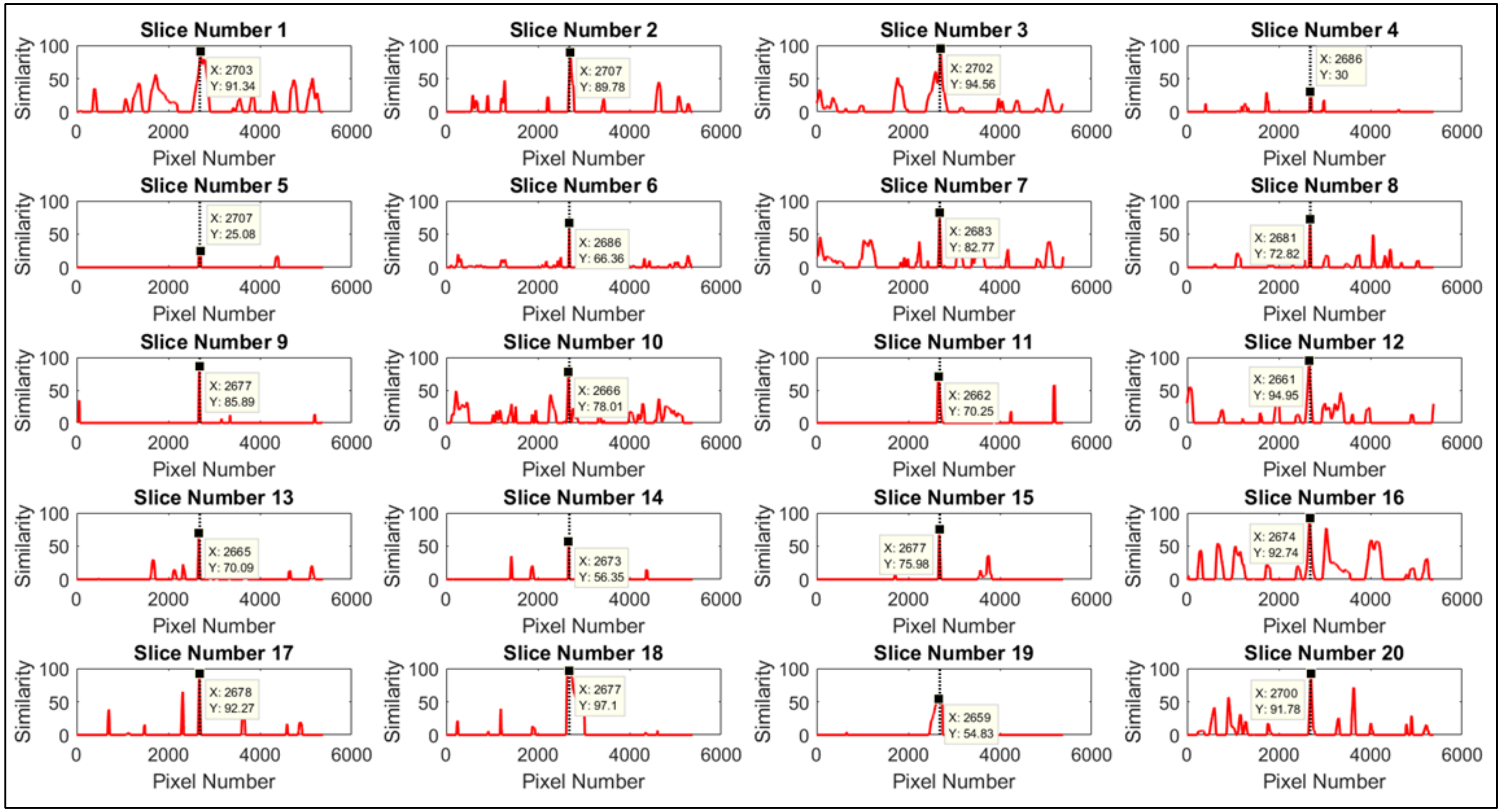


Figure 4.12. All similarity graphs are shifted by an amount of total ImageWidth/2 which is 2688 for this case

- After evaluating all test signal slices, a summation as shown in Figure 4.13 is obtained using the similarity graphs given in Figure 4.12. The location of maximum value of the summation graph gives a good estimate about the shift between test and map signals. For the example given in Figure 4.13, the shift between map and test signal is 10 pixels.

  Depending on images, there might be peaks close to the maximum value. Peaks which have amplitude of 85% of the maximum value are also examined while finding the final similarity value between the test and the map signal.
- The maximum value found in Figure 4.13 gives a nearly perfect estimate about similarity of two signals. It means that without executing the following steps and by using the beforehand calculations in order to contemplate the maximum value, localization may be succeeded as presented in EXPERIMENTS AND RESULTS with “Sum_max” method.
- Although the location of maximum value given in Figure 4.13 represents a good estimate for the value of shift between test and map signals, not every slice has to have a maximum value in the close vicinity of that value. The reason why is that some slices may have occlusions or may be similar to other slices due to repeating features in the skyline. Therefore, in Figure 4.12, maximum values and indices are searched within the limits of +-110 of the location of maximum value given in Figure 4.13. For this example, it is [2568, 2788].
- If the maximum similarity value is less than 29 out of 100, that slice is assumed to be unusable for similarity calculation purposes. It may be unusable due to occlusions or too many indentations that map and test signal could not match as for this case given in Figure 4.14.
- The other limiting factor is mean difference value between map and test signal slices. If it is more than 60 or 70 (The value depends on mean and standard deviation of mean difference of other slices with correspondent map signal slices.), that slice is also assumed to be unusable. The reason

why is that as map and test signal slices are compared in zero mean form as shown in Figure 4.9, it might be possible for two slices to give high similarity value even if they are far from each other when the whole image signals are considered.

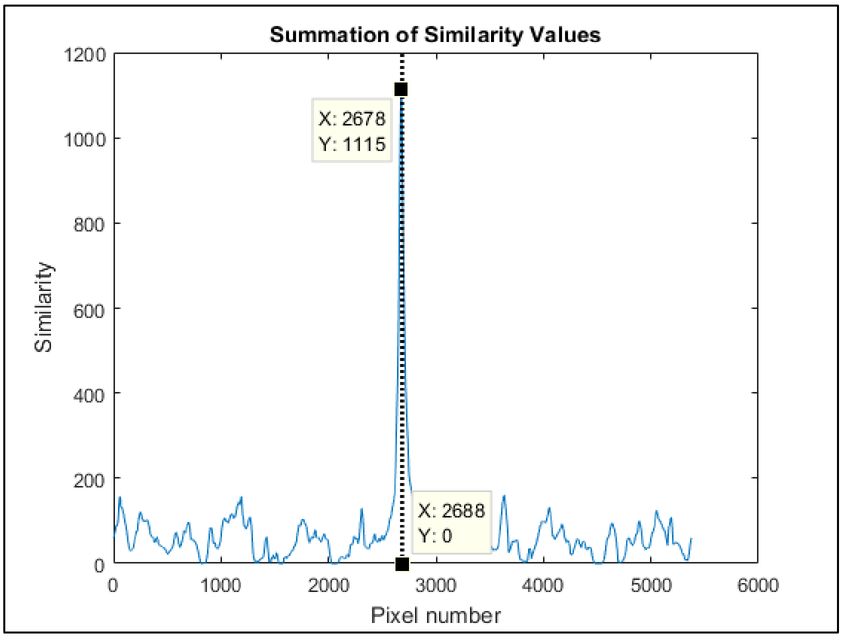


Figure 4.13. Summation of Twenty Similarity Graphs

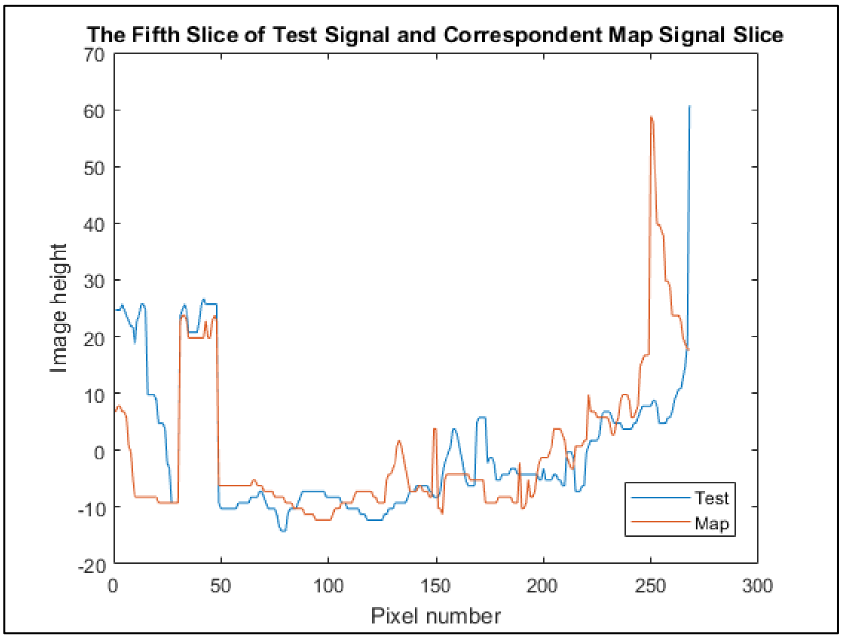


Figure 4.14. A Bad Match Between Test and Map Signal Slice

- A new test signal is formed by eliminating unusable slices as shown in Figure 4.15.

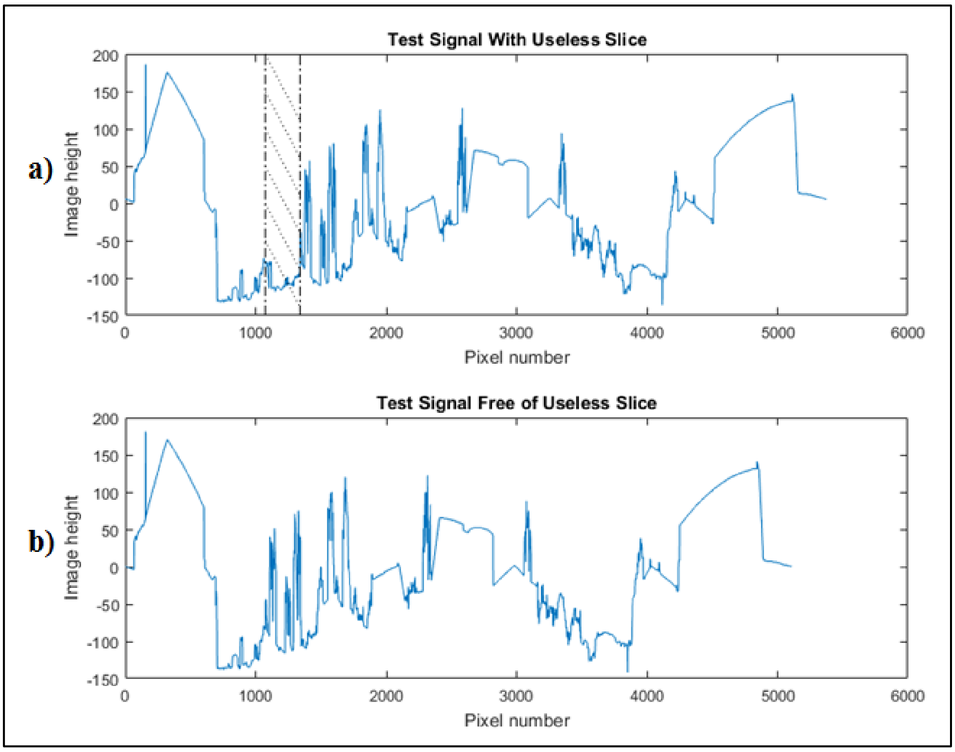


Figure 4.15. a) Test Signal with Soon-to-be Eliminated Unusable Slice
b) Test Signal without the Unusable Slice

- Although map signal is assumed to be perfect, it has to have the same width with test signal for comparison. Therefore, slices have to be eliminated from map signal until the widths are the same size. However, new map signal cannot be formed easily by eliminating slices corresponding to unusable test signal slices since there is a possible shift between test and map signals. Therefore, shift and compare method is used.

  Map signal is shifted along the width of image and the slices corresponding to the unusable slices in test signal are eliminated from map signal. While doing that, instead of just eliminating the slices, new signal is formed slice by slice in order to adjust mean of map signal slices to corresponding test signal slices as given in Figure 4.16. Mean adjustment is performed in order to get rid of S-like shape on horizon due to horizon correction issues of camera as shown in Figure 4.17.

Similarity values are obtained for each shifted and eliminated version of map signal. Finally, the one with the highest similarity is chosen.

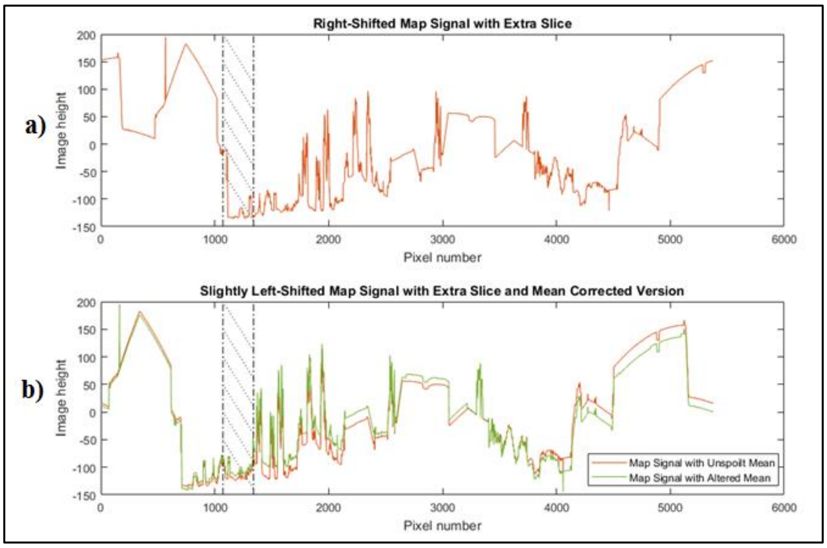


Figure 4.16. a) Map Signal Shifted Certain Amounts

b) The Most Suitable Shift Of Map Signal

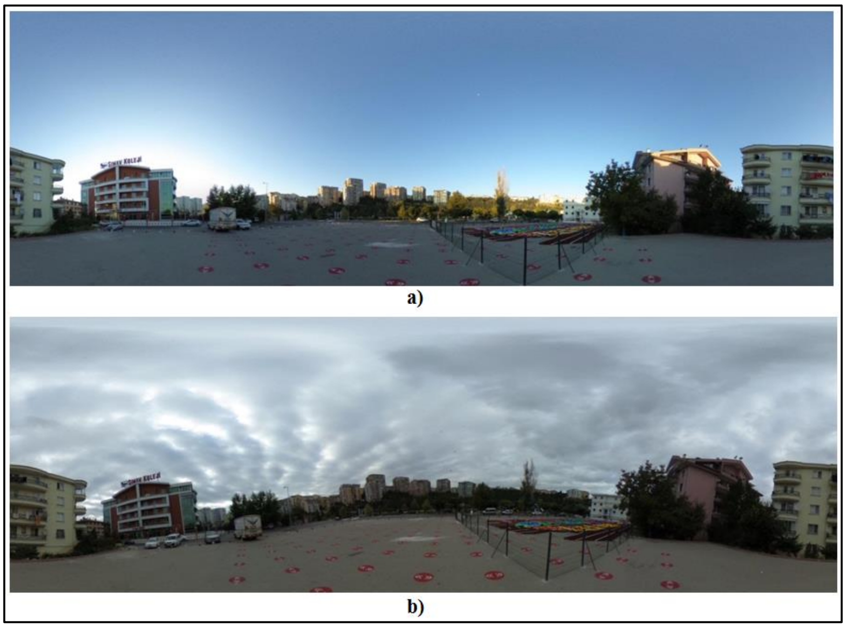


Figure 4.17. Photographs Taken at the Same Location

a) Successful Horizon Correction b) Unsuccessful Horizon Correction

- The highest similarity value is multiplied by a correction factor as given in Figure 4.18. It depends on the number of usable slices of a test signal. This is applied in order to avoid high similarity values with low usable slices.
- As a result, new test and map signals are formed as given in Figure 4.19.

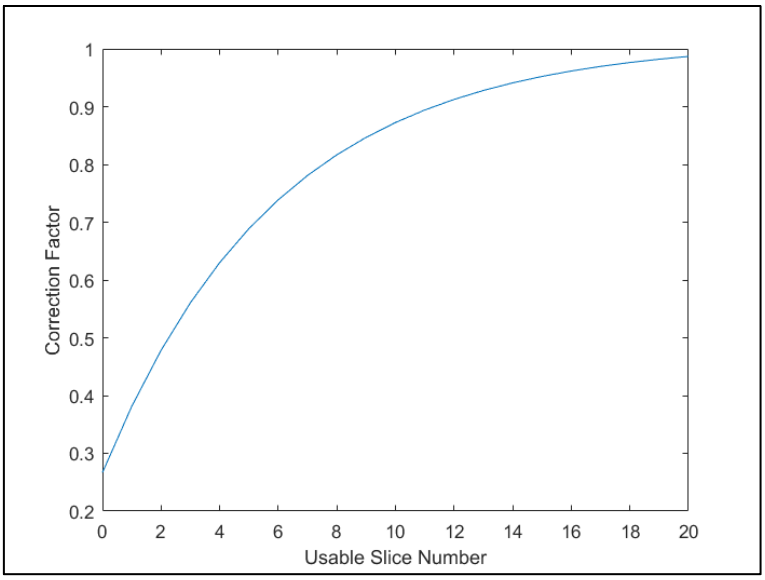


Figure 4.18. Usable Slice Number Versus Correction Factor

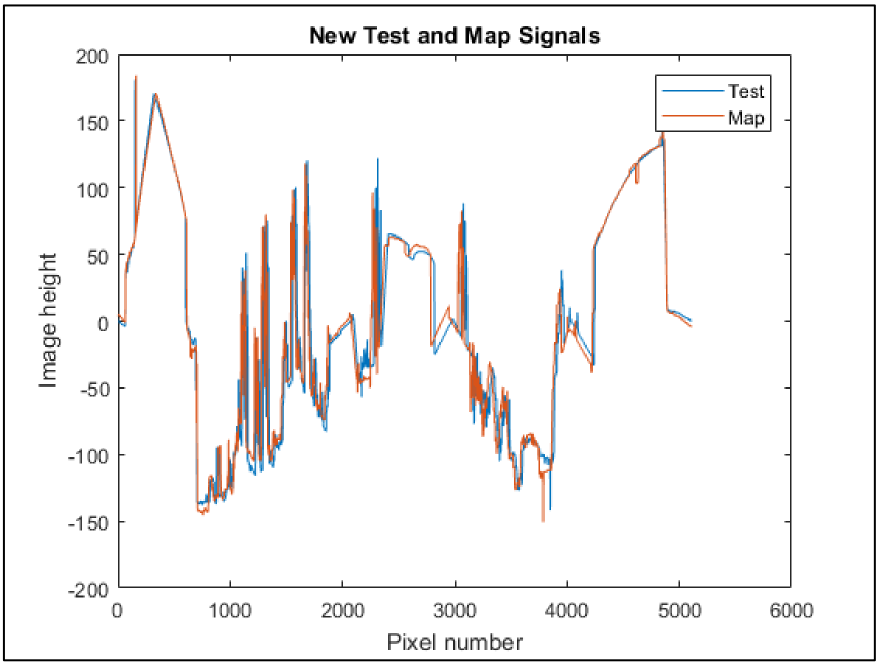


Figure 4.19. New Test and Map Signals to Obtain a Similarity Value

It is important to note that parameters (20 for slice numbers, 85% for peak search, +-110 for search limit, 29 for unusable slice elimination, 60 or 70 for mean difference limit) used for algorithm are found by common trial and error processes. After obtaining approximate values for parameters, fine work is performed by investigating skylines which give unexpected unsuccessful localizations. The unexpected situations are usually when test photograph is obtained under the same conditions with the map as will be explained later. Also, if one of the adjacent test signals results in an unsuccessful localization, then this situation is also examined.

Complexity of algorithm regarding map points and runtime can be obtained as linear. Algorithm 3 is executed for each map signal comparison with the test signal. That means as the number of map points (n) increase, runtime of algorithm would increase linearly with the number of map points (O(n)). However, it is important to note that since map area is defined as square region in this study, number of map points cannot increase one by one. It would increase by order of square with respect to number of points on one edge of square. For example, one bigger size of 10 meters x 10 meters area with 121 map points is 11 meter x 11 meters area with 144 map points. Besides localization complexity with respect to number of points, it should also be noted that number of slices in each image is an important parameter that affects localization performance, especially against occluded regions. Given that each slice from a test image is compared with all other slices in the maps, localization complexity is quadratic ($O(n^2)$) with respect to the number of slices being used.

CHAPTER 5

EXPERIMENTS AND RESULTS

An autonomous territorial robot is expected to become familiar with its environment by capturing photographs in an area, then by extracting skyline signals in order to use them as references for localization purposes. This process is called as mapping. As the map is established, robot is expected to match a test signal, which is defined as a skyline signal obtained at any point inside the map area, with the skyline signals of map and perform localization. In order to simulate this process, photographs for mapping and testing purposes are captured with Ricoh Theta S spherical camera manually by a human. Five different maps are formed for five different environments.

All the environments, “B”, “G”, “K “,”T” and “U”, have different features as shown in Figure 5.1. GPS coordinates are 39° 53’ 20’’ N 32° 46’ 55’’ E, 39° 53’ 22’’ N 32° 46’ 47’’ E, 39° 58’ 29’’ N 32° 52’ 29’’ E, 39° 53’ 49’’ N 32° 46’ 33’’ E and 39° 53’ 36’’ N 32° 46’ 35’’ E respectively. While building to tree ratio is close to one for “B” and “G”, this value increases for “K” and “T”. “U” has only trees. The distinctive factor for “B” and “G” is that “G” has the building as dominant feature. While in “K” buildings are far away from the camera, in “T” they are much closer. Also “T” has more trees than “K”. Except “K”, other environments are from METU campus. Google earth images of environments are given in Figure 5.2.

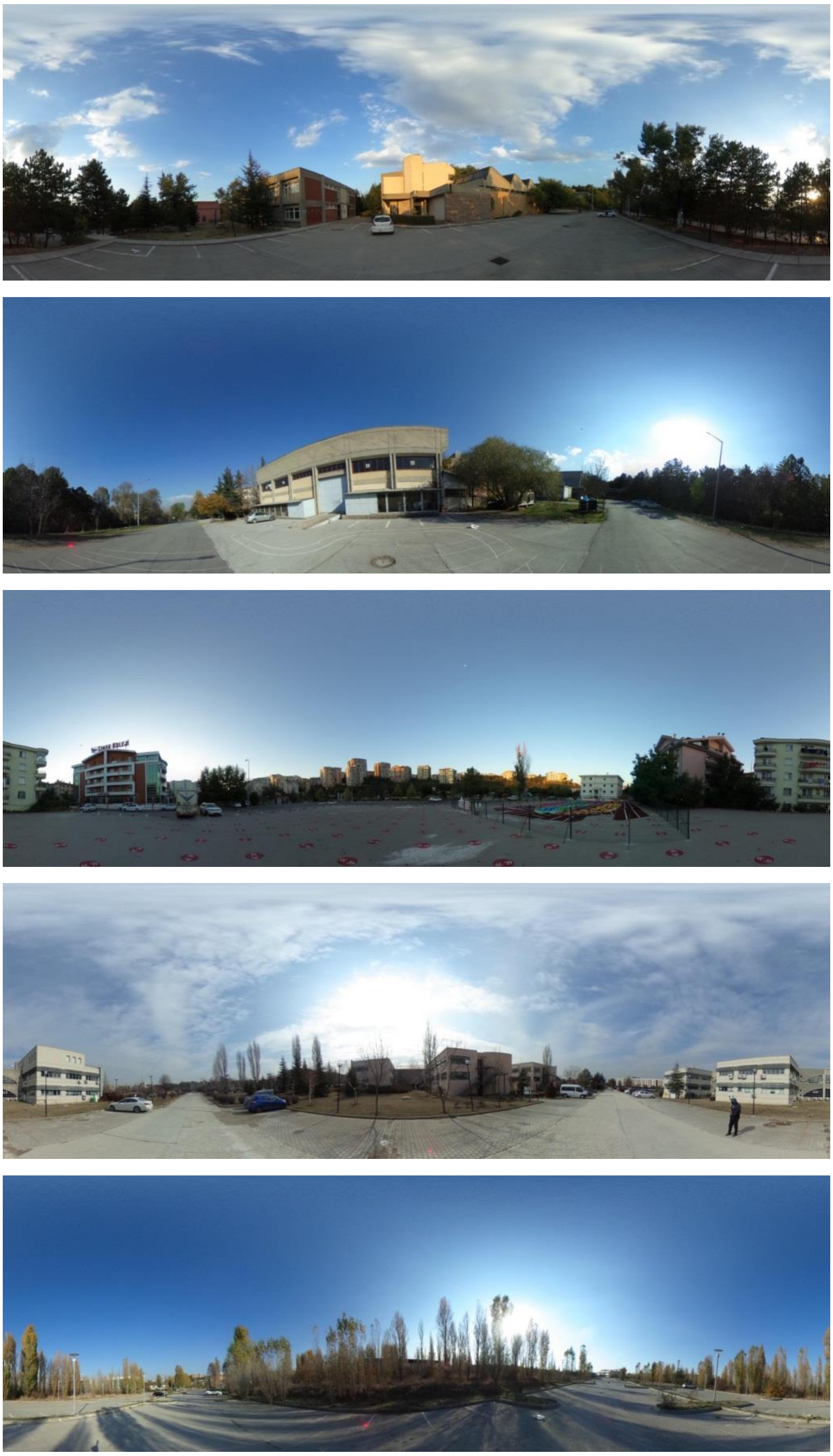


Figure 5.1. Experiment Environments of “B”, “G”, “K”, “T”, “U” respectively.

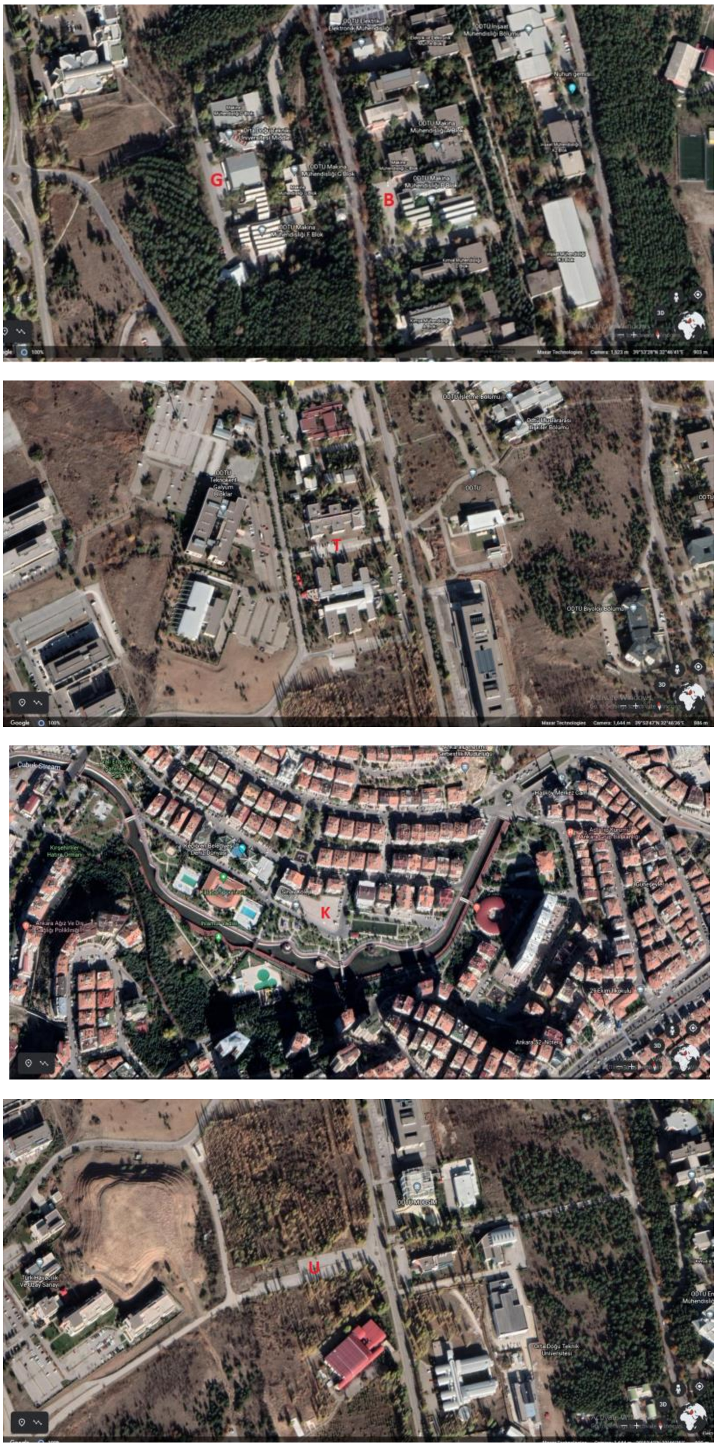
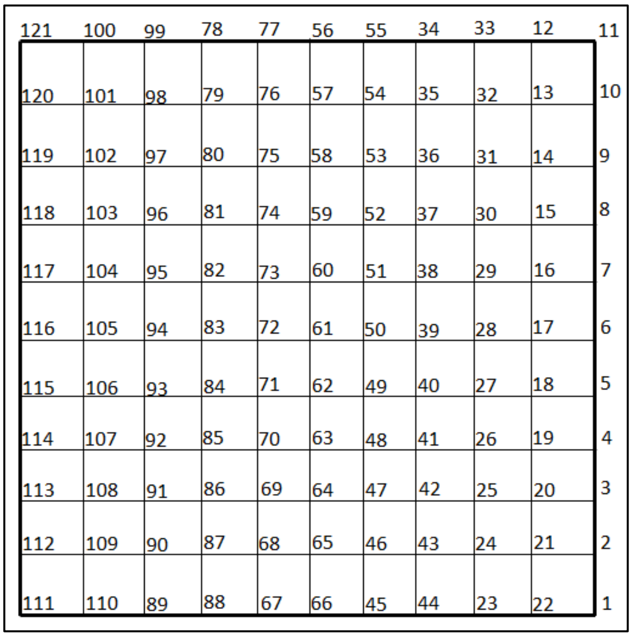


Figure 5.2. Google Earth Images of Environments “B”, “G”, “K”, “T”, “U” respectively

Total of 1018 photos are captured and used for the purpose of this thesis. While 605 of them are used for forming map, 413 are used for test purposes. In addition to natural test skyline signals, artificial occlusions to distort skyline are applied in order to observe at how much occlusion does the algorithm gives unreliable results for localization. With those, localizations of overall 1394 skyline signals are performed using the algorithm.

5.1 Map and Test Points

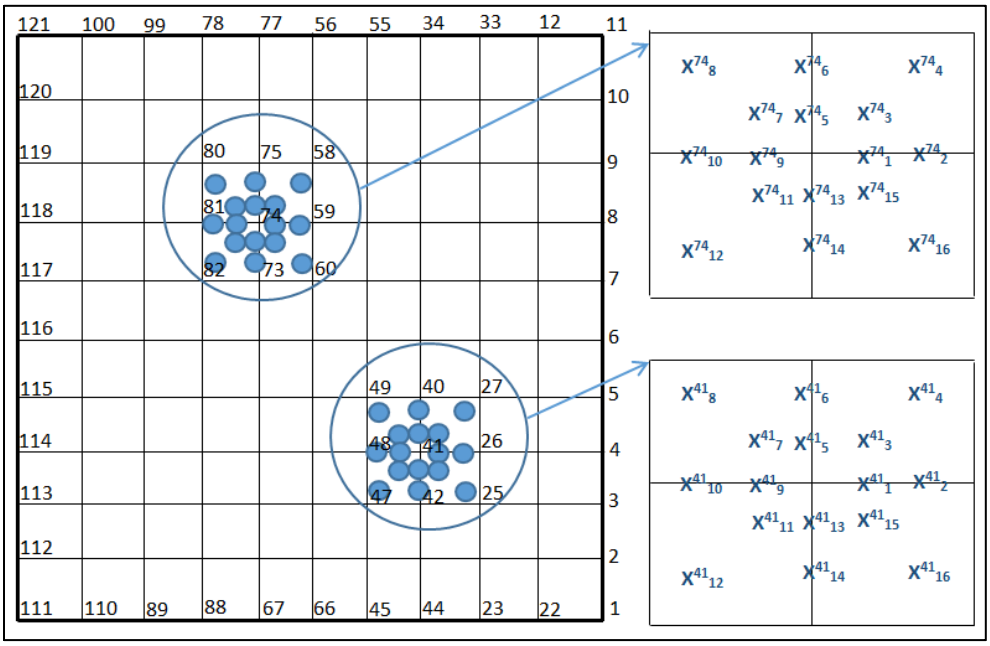
As explained in METHOD, maps are collections of points within a 10 meters x 10 meters region that are tessellated with 1 meter x 1 meter squares, where corners of the squares correspond to the reference points on the map as given in Figure 5.3. Since five different maps are formed for five different environments, points are labelled using environment and map location information as “X(Map,i)” where X=“B”, “G”, “K “,”T”, “U” and i=1,2,…,121.



121	100	99	78	77	56	55	34	33	12	11
120	101	98	79	76	57	54	35	32	13	10
119	102	97	80	75	58	53	36	31	14	9
118	103	96	81	74	59	52	37	30	15	8
117	104	95	82	73	60	51	38	29	16	7
116	105	94	83	72	61	50	39	28	17	6
115	106	93	84	71	62	49	40	27	18	5
114	107	92	85	70	63	48	41	26	19	4
113	108	91	86	69	64	47	42	25	20	3
112	109	90	87	68	65	46	43	24	21	2
111	110	89	88	67	66	45	44	23	22	1

Figure 5.3. Map Area and Points

Test sets are chosen with the purpose of covering different areas of map region. These sets are labeled as “41”, “74” as given in Figure 5.4, “Diag1”, “Diag2” as given in Figure 5.5 and “Horz”, “VertL”, “VertR” as given in Figure 5.6. Test points for “41” and “74” are labelled using environment, test set and test number information as “X(TestSet_1,j)” where X=“B”, “G”, “K “,”T”, “U”; TestSet_1 is either “41” or “74” and j=1,2,…,16. Test points for other test sets are also labelled using environment, test set and test number information as “X(TestSet_2,k)” where X=“B”, “G”, “K “,”T”, “U”; TestSet_2=”Diag1”, “Diag2”, “Horz”, “VertL”, “VertR” and k=1,2,…,10.



121
100
99
78
77
56
55
34
33
12
11
120
10
119
9
118
8
117
7
116
6
115
5
114
4
113
3
112
2
111
110
89
88
67
66
45
44
23
22
1
80
75
58
81
74
59
82
73
60
49
40
27
48
41
26
47
42
25
X^{74}_{8} X^{74}_{6} X^{74}_{4}
X^{74}_{7} X^{74}_{5} X^{74}_{3}
X^{74}_{10} X^{74}_{9} X^{74}_{1} X^{74}_{2}
X^{74}_{11} X^{74}_{13} X^{74}_{15}
X^{74}_{12} X^{74}_{14} X^{74}_{16}
X^{41}_{8} X^{41}_{6} X^{41}_{4}
X^{41}_{7} X^{41}_{5} X^{41}_{3}
X^{41}_{10} X^{41}_{9} X^{41}_{1} X^{41}_{2}
X^{41}_{11} X^{41}_{13} X^{41}_{15}
X^{41}_{12} X^{41}_{14} X^{41}_{16}

Figure 5.4. Points of Test Areas “41” and “74”

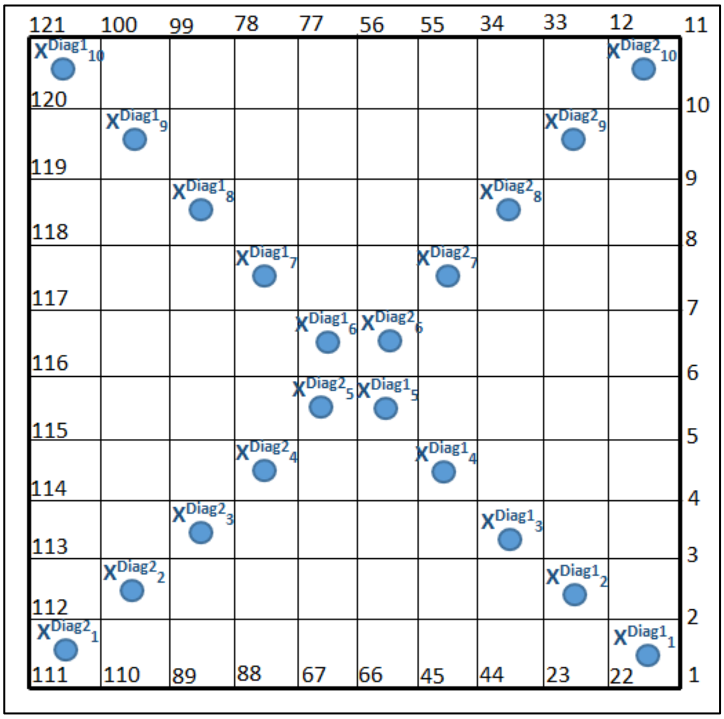


Figure 5.5. Points of Test Areas “Diag1” and “Diag2”

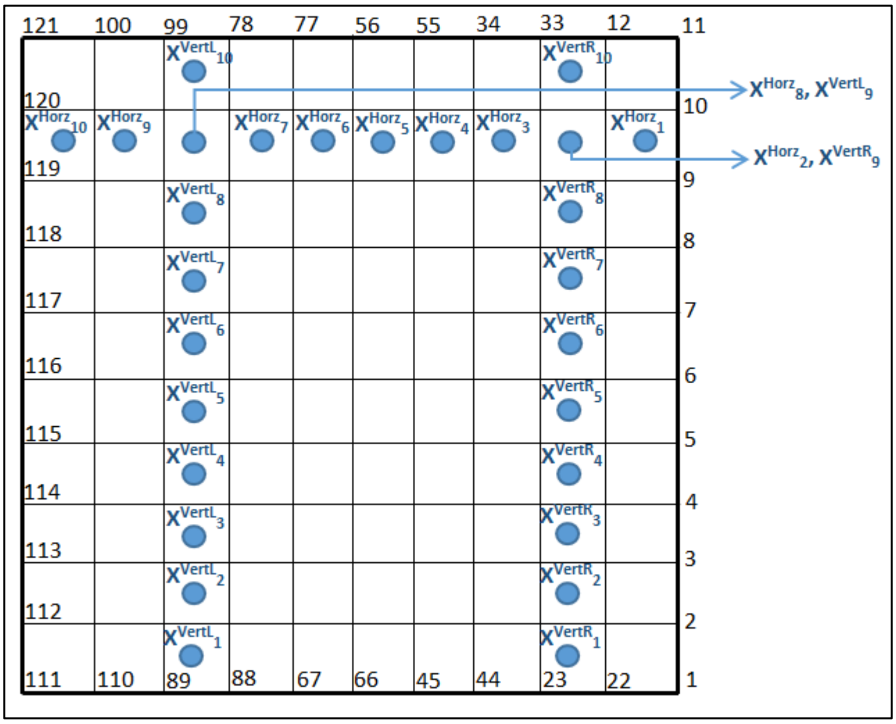


Figure 5.6. Points of Test Areas “Horz”, “VertL” and “VertR”

5.2 Experiment Conditions

Map photographs are captured in five different environments as explained before and maps are formed for those environments only once whereas test photographs are captured several times in these environments on different seasons and under different weather and lighting conditions. These conditions are added to the end of map and test labelings. Following sub-sections investigate comparisons between skyline signals that are obtained at the same point but different times. While the reference condition is the one when map is formed, other conditions differ from location to location. It is expected to have a number of unusable slices in skyline signals during the process of comparison since changing conditions are expected to cause disturbances in skylines although they are obtained at the same point.

5.2.1 Test Location "B"

Map condition for location "B" is sunset and trees with leaves. Two test sets, "41" and "74", are also obtained at the same condition. Other conditions are noon and trees without leaves, afternoon and trees without leaves, dark clouds and trees without leaves. Test sets "Diag1" and "Diag2" are obtained at conditions trees without leaves and noon, afternoon and dark clouds. All the conditions are given in Table 5.1.

Table 5.1 Different Test Conditions for Location"B"

B	Noon	Afternoon	Sunset	Dark Clouds
Trees Without Leaves	Diag1, Diag2	Diag1, Diag2 *		Diag1, Diag2 **
Trees With Leaves			Map, 41, 74	

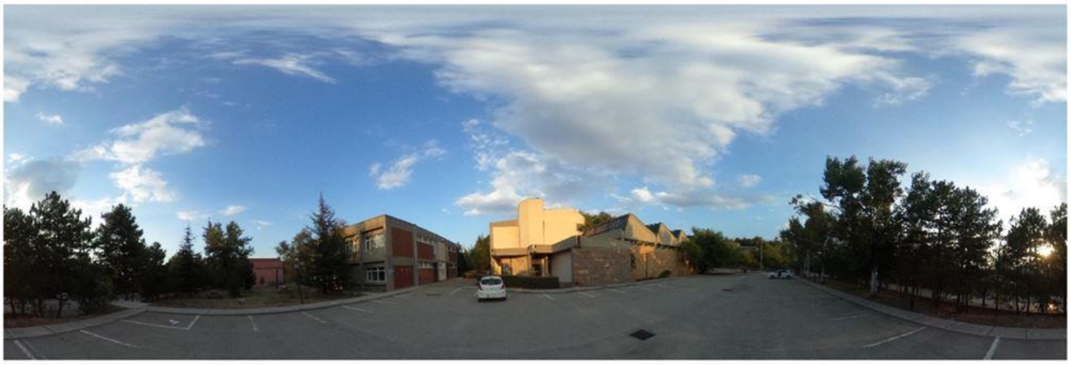
*Due to construction that started after mapping, not all test points were available during experiments.
** Due to construction and a car, not all test points are used.

For the comparison of skylines, while “B(Map,50,Leaves-Sunset)” is used as reference signal since “B(Diag1,5,Leaves-Sunset)” does not exist. “B(Diag1,5,NoLeaves-Noon)”, “B(Diag1,5,NoLeaves-Afternoon)”, “B(Diag1,5,NoLeaves-DarkClouds)” are used for different conditions. Photographs are shown in Figure 5.7. Skyline signals are given in Figure 5.8.

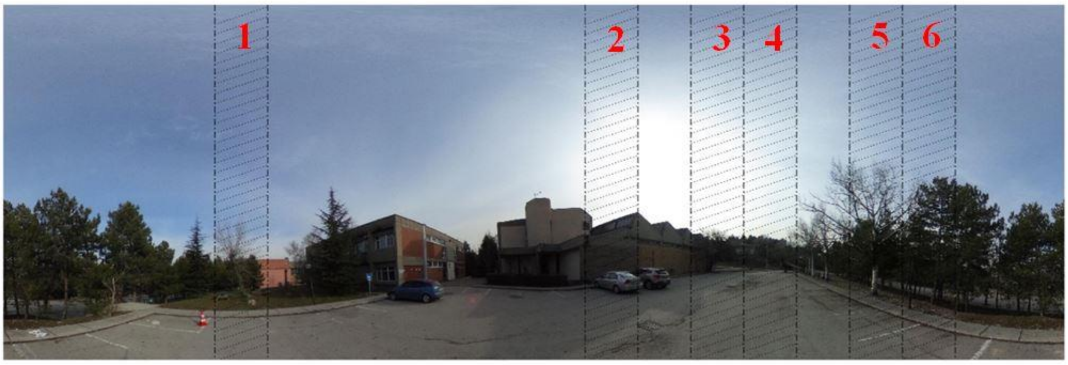
Figure 5.7.b and corresponding skyline signal Figure 5.8.a have six unusable slices. Unusable slice numbers 1, 2, 5 and 6 are eliminated due to seasonal changes. While trees have leaves on map photograph as shown in Figure 5.7.a, trees on test photograph have no leaves. The sun position is responsible for unusable slice numbers 3 and 4. It causes skylines to be different in test and map photographs and these differences are not tolerated by the algorithm.

There are a lot of unusable slices in Figure 5.7.c and corresponding skyline signal Figure 5.8.b. Unusable slice numbers 2, 3 and 7 are eliminated due to seasonal changes. Number 4 is eliminated due to a combination of reasons. In test signal, due to construction, the color of building has changed to white and as the sun hits against the white building, it distorts the skyline. Number 5 and 6 are also eliminated due to white building and the sun position combination. Small differences in skyline are not tolerated by the algorithm in unusable slice numbers 8 and 9 similar to the case in Figure 5.7.b for unusable slice numbers 3 and 4. The sun dramatically damages the skyline in test signal, causing unusable slice number 10 and 11 to be eliminated. For unusable slice number 13, the sun is also responsible for distortion of skyline in map signal. Finally, unusable slice number 1 and 12 are eliminated due to high mean differences. This is due to S-like warping that occurs on horizon as explained in METHOD. Horizon correction is not fully achieved. Photographs in Figure 5.7.c and Figure 5.7.d are captured exactly at the same point and the skyline comparison is given in Figure 5.9. As can be seen from the figure, there are also mean differences in slices corresponding to unusable slice numbers 1 and 12 in Figure 5.7.c. This combining with the fact that Figure 5.7.a and Figure 5.7.c are not captured exactly at the same point and the skyline is expected to change; unusable slice numbers 1 and 12 are eliminated.

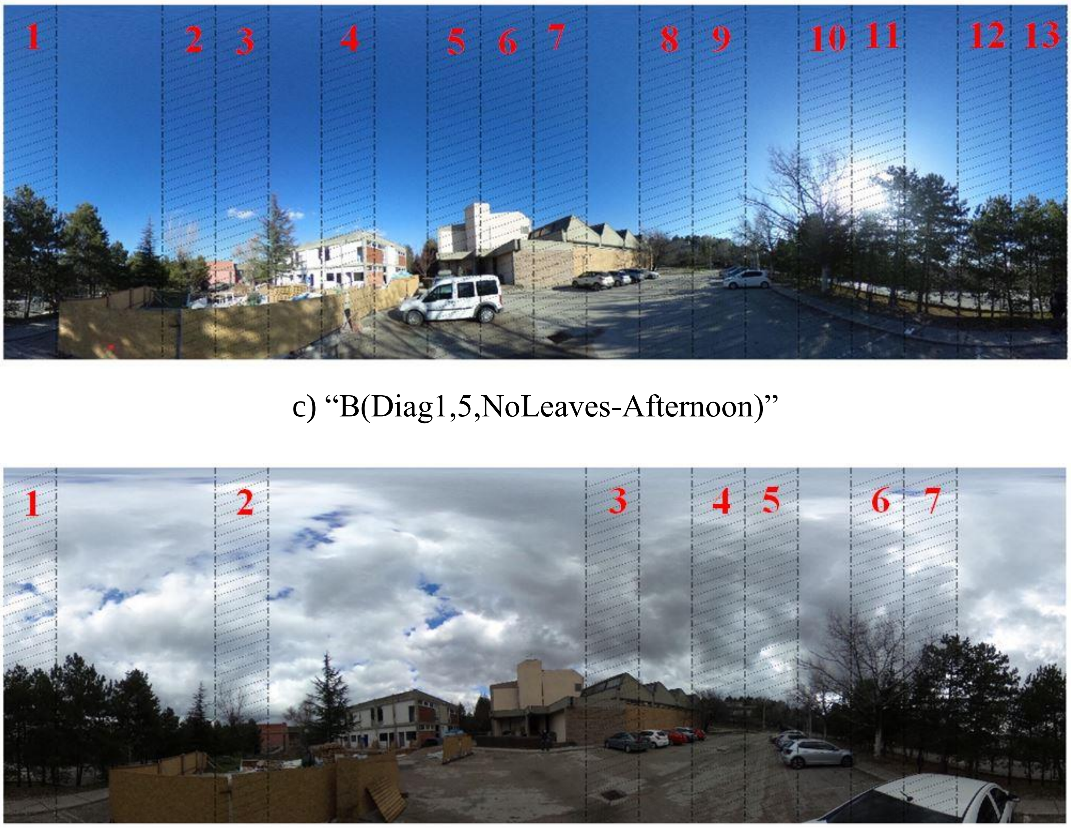
Figure 5.7.d and corresponding skyline signal Figure 5.8.c have seven unusable slices. The first one is due to small changes in tree skyline due to weather conditions and the fact that the photographs are not captured exactly at the same point. The reasons for the others are the same as in Figure 5.7.b and Figure 5.8.a.



a) “B(Map,50,Leaves-Sunset)”



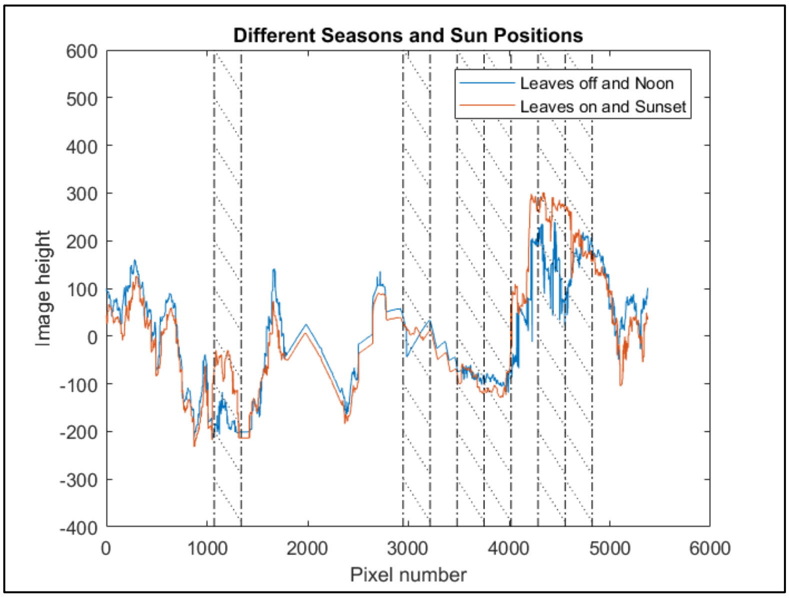
b) “B(Diag1,5,NoLeaves-Noon)”



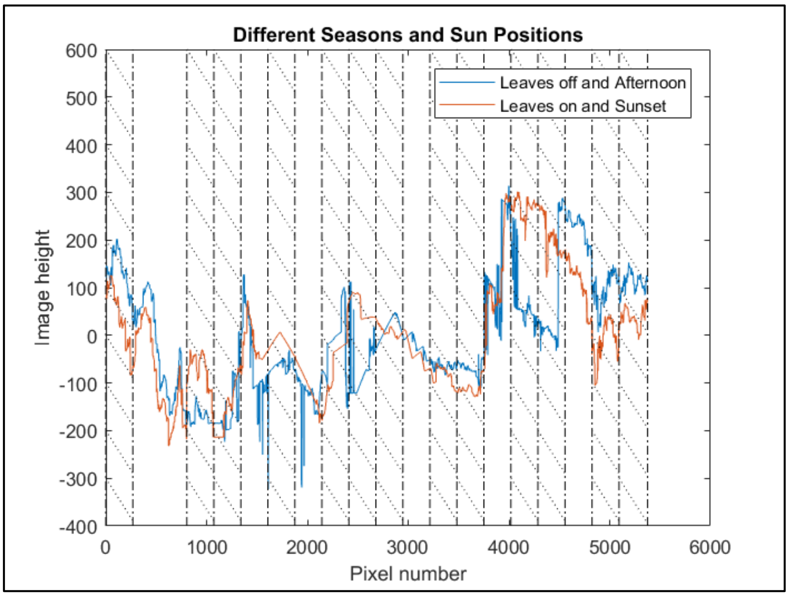
c) “B(Diag1,5,NoLeaves-Afternoon)”

d) “B(Diag1,5,NoLeaves-DarkClouds)”

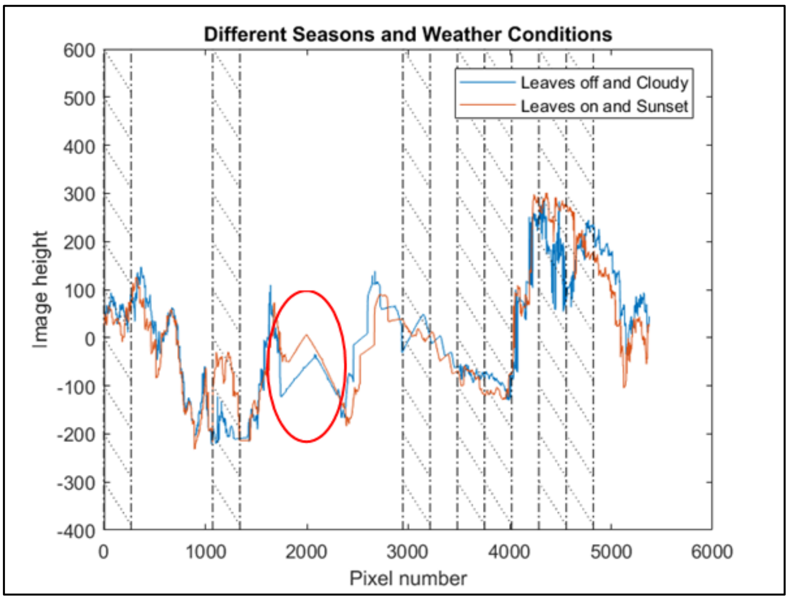
Figure 5.7. Photographs Captured at Different Conditions for Test Location “B”



a) “B(Map,50,Leaves-Sunset)” and “B(Diag1,5,NoLeaves-Noon)”



b) “B(Map,50,Leaves-Sunset)” and “B(Diag1,5,NoLeaves-Afternoon)”



c) “B(Map,50,Leaves-Sunset)” and “B(Diag1,5,NoLeaves-DarkClouds)”

Figure 5.8. Skyline Signals of Photographs Captured at Different Conditions for Test Location “B”

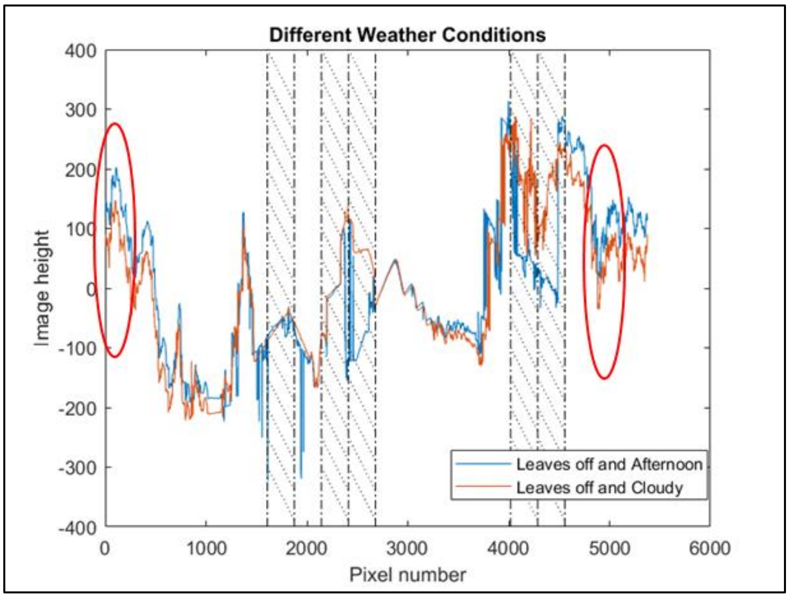


Figure 5.9. Skyline Signals of “B(Diag1,5,NoLeaves-Afternoon)” and “B(Diag1,5,NoLeaves-DarkClouds)”

5.2.2 Test Location “G”

Map condition for location “G” is afternoon and trees with leaves. Four test sets, “Diag1”, “Diag2”, “Horz”, “VertL”, are also obtained at the same condition. Test sets “Diag1” and “Diag2” are obtained at conditions noon and trees without leaves, sunset and trees without leaves, dark clouds and trees without leaves. All the conditions are given in Table 5.2.

Table 5.2 Different Test Conditions for Location “G”

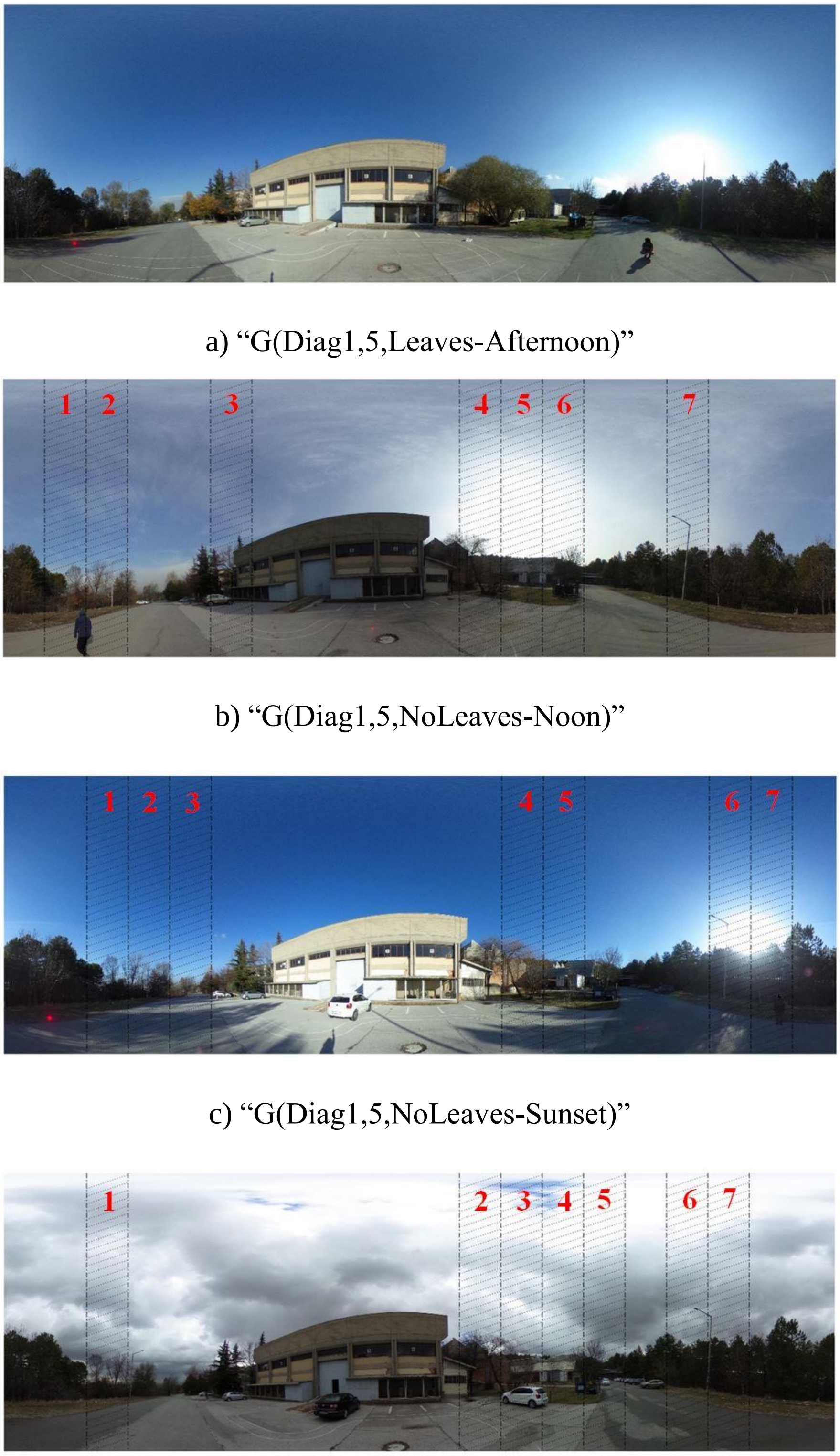
G	Noon	Afternoon	Sunset	Dark Clouds
Trees Without Leaves	Diag1, Diag2		Diag1, Diag2	Diag1, Diag2
Trees With Leaves		Map, Diag1, Diag2, Horz, VertL		

For the comparison of skylines, while “G(Diag1,5,Leaves-Afternoon)” is used as reference signal, “G(Diag1,5,NoLeaves-Noon)”, “G(Diag1,5,NoLeaves-Sunset)”, “G(Diag1,5,NoLeaves-DarkClouds)” are used for different conditions. Photo^{g}raphs are shown in Figure 5.10. Skyline signals are given in Figure 5.11.

For Figure 5.10.^{b} and corresponding skyline signal Figure 5.11.a; unusable slice numbers 1, 2, 4, 5 are eliminated due to seasonal changes. While “G(Diag1,5,Leaves-Afternoon)” has tree leaves, “G(Diag1,5,NoLeaves-Noon)” does not have them. For unusable slice number 3, skyline signal for a tree with no leaves in “G(Diag1,5,Leaves-Afternoon)” is unable to form while “G(Diag1,5,NoLeaves-Noon)” has skyline for that. For unusable slice number 6, skylines do not match due to too many indentations caused by a tree with no leaves. Due to sun position in “G(Diag1,5,NoLeaves-Noon)”, the skyline is much more vivid as can be seen in Figure 5.11.a for that slice. For unusable slice number 7, skyline for pole is unable to form in “G(Diag1,5,Leaves-Afternoon)” due to sun position.

For Figure 5.10.c and corresponding skyline signal Figure 5.11.b; unusable slice numbers 1, 2, 4, 5 are eliminated due to seasonal changes. For unusable slice number 3, the skylines do not match due to too many indentations caused by a tree with no leaves. For unusable slice numbers 6 and 7, sunset in “G(Diag1,5,NoLeaves-Sunset)” has caused a cavity in skyline as can be seen in Figure 5.11.b.

For Figure 5.10.d and corresponding skyline signal Figure 5.11.c; unusable slice numbers 1, 2, 3 are eliminated due to seasonal changes. The skyline of a tree without leaves has more distinct features in a dark cloudy environment; therefore, weather conditions causes unusable slice numbers 4 and 5 to be eliminated. For unusable slice number 6 and 7, skyline for pole is unable to form in “G(Diag1,5,Leaves-Afternoon)” due to sun position while it is present in a dark cloudy weather.



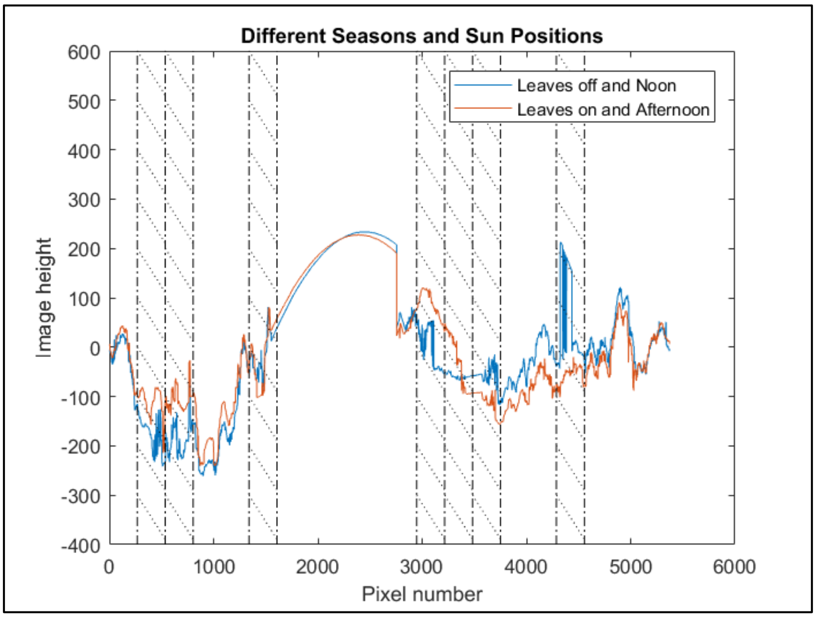
a) “G(Diag1,5,Leaves-Afternoon)”

b) “G(Diag1,5,NoLeaves-Noon)”

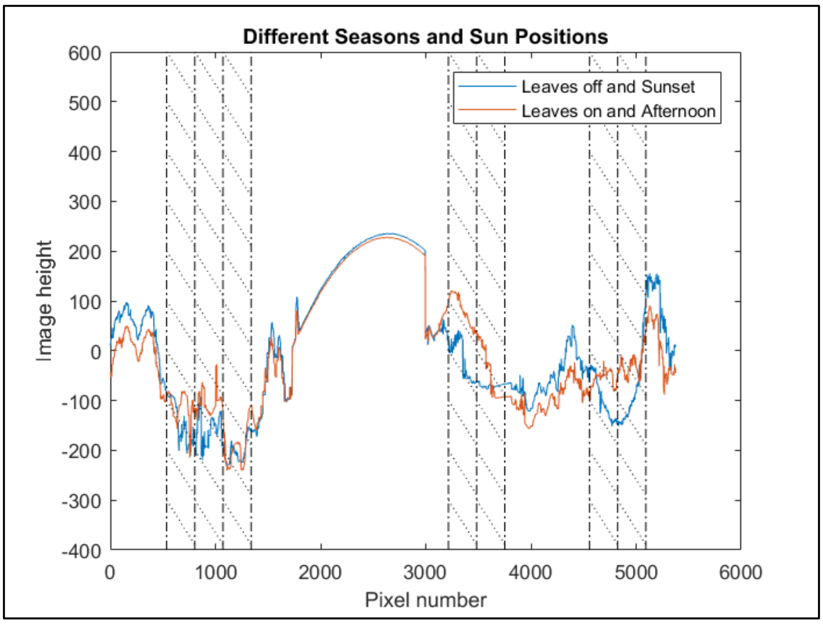
c) “G(Diag1,5,NoLeaves-Sunset)”

d) “G(Diag1,5,NoLeaves-DarkClouds)”

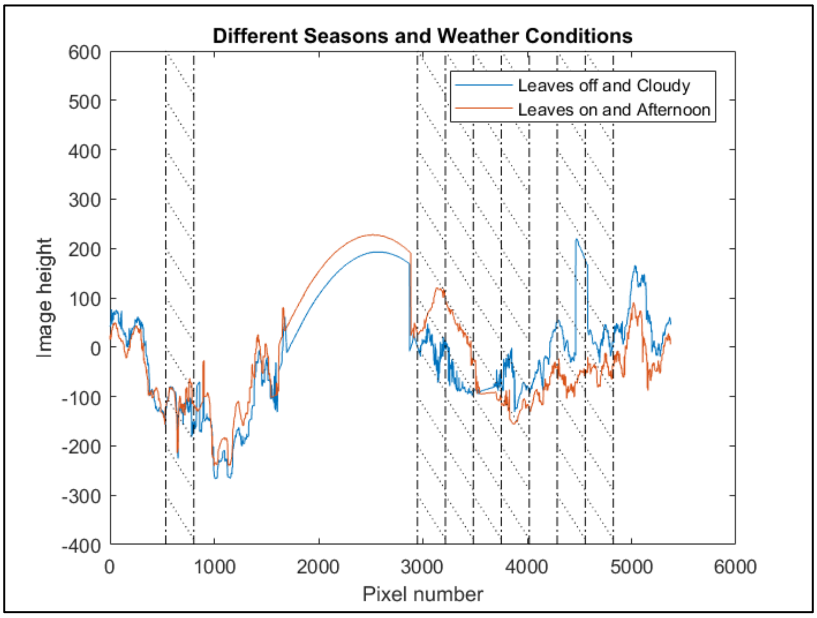
Figure 5.10. Photographs Captured At Different Conditions for Test Location “G”



a) "G(Diag1,5,Leaves-Afternoon)" and "G(Diag1,5,NoLeaves-Noon)"



b) "G(Diag1,5,Leaves-Afternoon)" and "G(Diag1,5,NoLeaves-Sunset)"



c) “G(Diag1,5,Leaves-Afternoon)” and “G(Diag1,5,NoLeaves-DarkClouds)”

Figure 5.11. Skyline Signals of Photographs Captured at Different Conditions for Test Location “G”

5.2.3 Test Location “K”

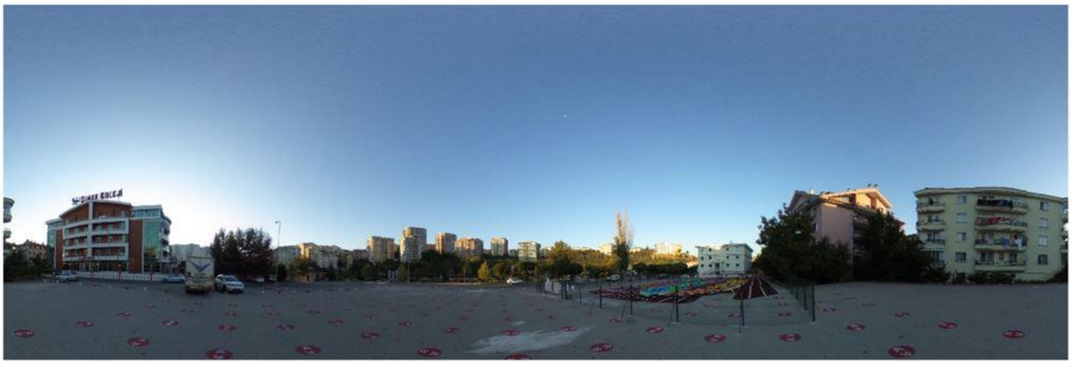
Map condition for location “K” is sunrise and trees with leaves. The other condition is cloudy and trees with leaves. Four test sets, “Diag1”, “Diag2”, “Horz”, “VertR”, are obtained at these two conditions. All the conditions are given in Table 5.3.

Table 5.3 Different Test Conditions for Location “K”

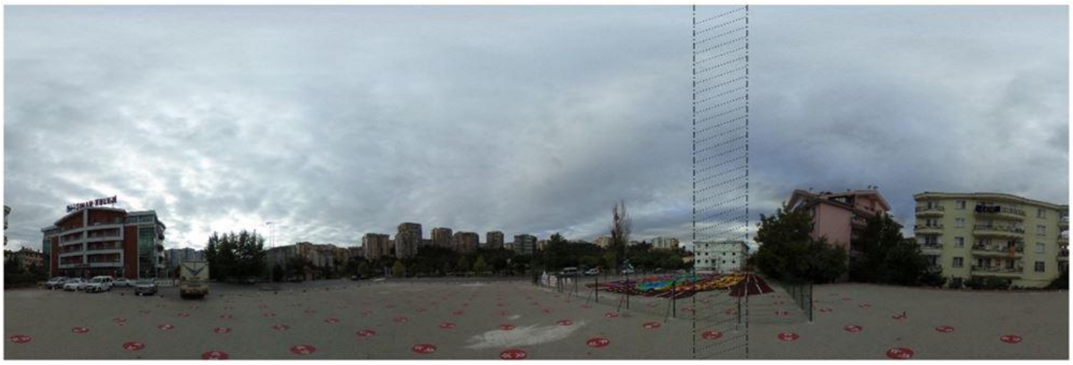
K	Sunrise	Cloudy
Trees With Leaves	Map, Diag1, Diag2, Horz, VertR	Diag1, Diag2, Horz, VertR

For the comparison of skylines, while “K(Diag1,5,Leaves-Sunrise)” is used as reference signal, “K(Diag1,5,Leaves-Cloudy)” is used for the other condition.

Photographs are shown in Figure Figure 5.12. Skyline signals are given in Figure 5.13.



a) “K(Diag1,5,Leaves-Sunrise)”



b) “K(Diag1,5,Leaves-Cloudy)”

Figure 5.12. Photographs Captured At Different Conditions for Test Location “K”

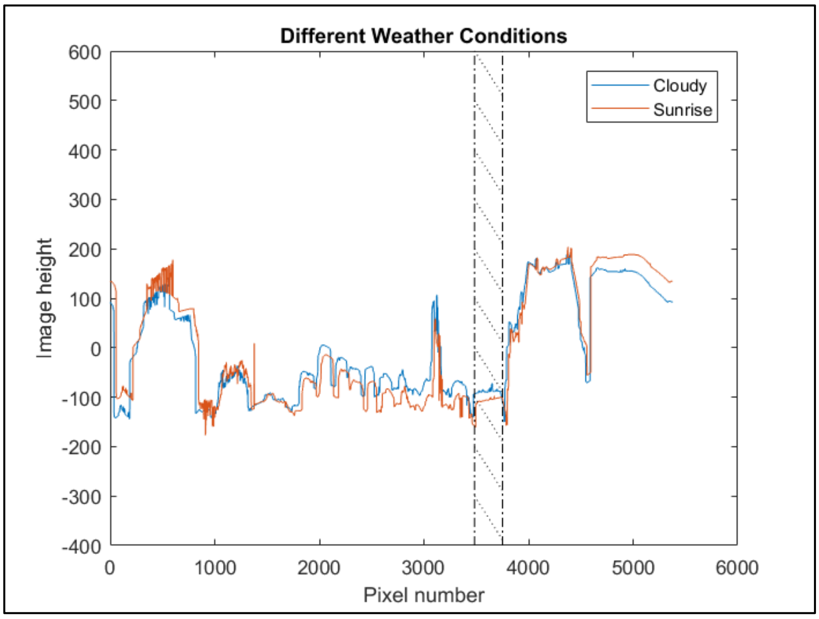


Figure 5.13. Skyline Signal of Photographs Captured at Different Conditions for Test Location “K”, “K(Diag1,5,Leaves-Sunrise)” and “K(Diag1,5,Leaves-Cloudy)”

The only unusable slice is the fourteenth slice of “K(Diag1,5,Leaves-Cloudy)”. It is shown in Figure 5.14. According to similarity formula as given in Algorithm 4 in METHOD, when “area_s1” is small, even tiny differences between test and map signal may not be tolerated since ratio of “area_diff/area_s1” is used for correlation. An example of tolerating case can be the nineteenth slice of “K(Diag1,5,Leaves-Cloudy)” as can be seen in Figure 5.15. Although “area_s1” value, which is 600, is smaller than the one for case given in Figure 5.14 which is 895, since “area_diff”, which is 297, is also smaller than the other one which is 713; for the case of nineteenth slice, the differences between “K(Diag1,5,Leaves-Cloudy)” and “K(Diag1,5,Leaves-Sunrise)” are tolerated.

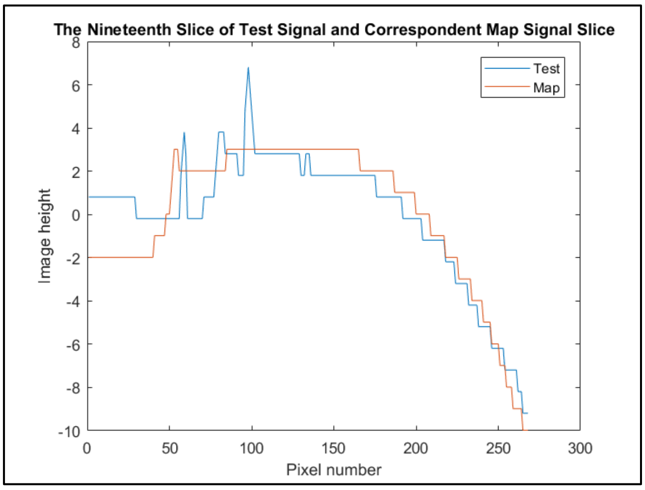


Figure 5.14. The Only Unusable Slice of “K(Diag1,5,Leaves-Cloudy)”

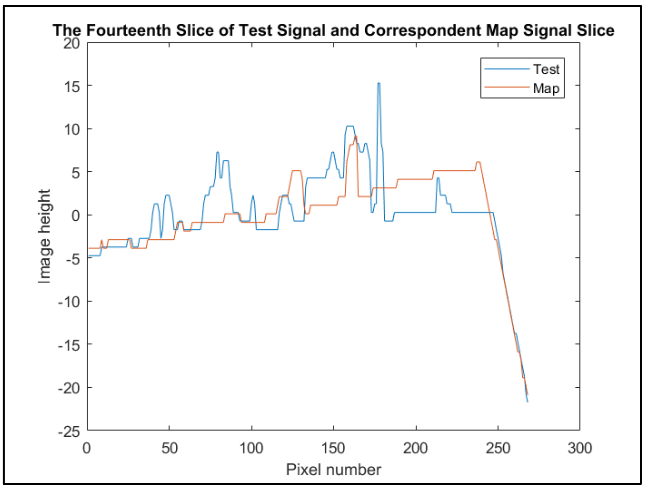


Figure 5.15. The Nineteenth Slice of “K(Diag1,5,Leaves-Cloudy)”

Although the intolerant case can be considered as a downside of algorithm, it is not expected to affect localization in an adverse way since that part of test signal is also expected to be unusable when compared with other map signals. This is shown in Figure 5.16. There are four close map points (“K(Map,38,Leaves-Sunrise)”, “K(Map,50,Leaves-Sunrise)”, “K(Map,71,Leaves-Sunrise)”, “K(Map,73,Leaves-Sunrise)”) to the test point “K(Diag1,5,Leaves-Cloudy)”. “K(Map,50,Leaves-Sunrise)” is the correct localization with the highest similarity value which is 87.9.

The others are 81.0, 80.1 and 75.3, respectively. In all four cases, the problematic test signal slice, which is the fourteenth, is eliminated.

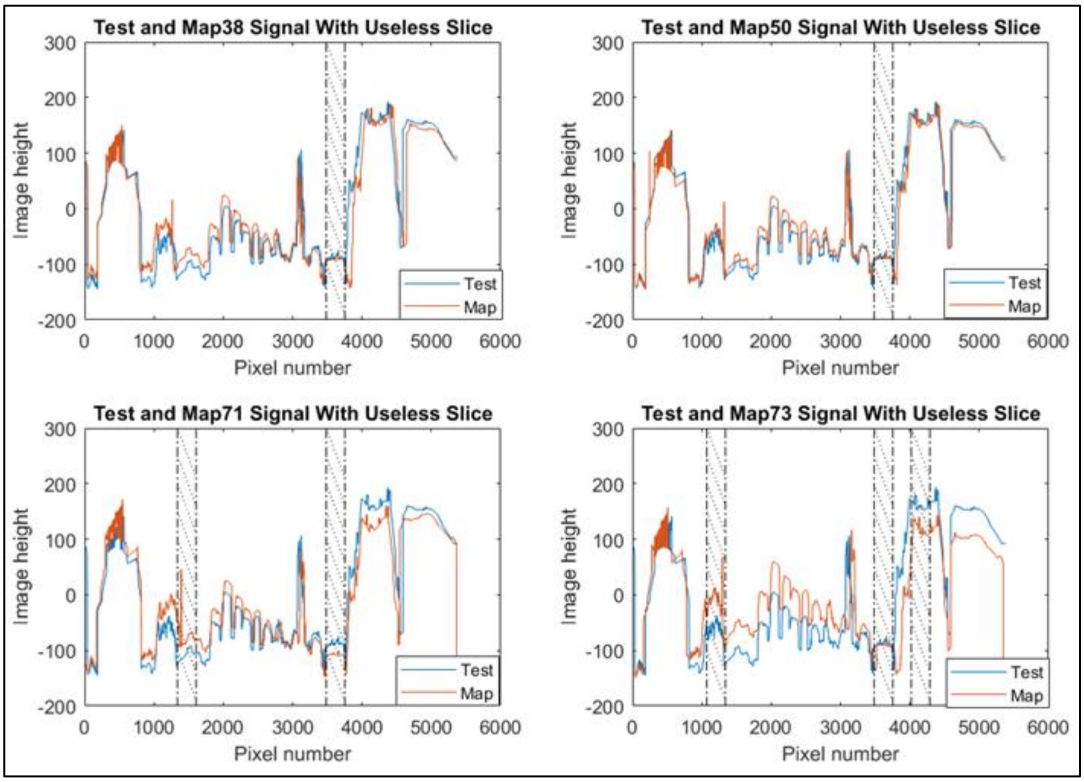


Figure 5.16. Skylines of "K(Diag1,5,Leaves-Cloudy)" and Some Map Points

5.2.4 Test Location "T"

Map condition for location "T" is noon and trees without leaves. Other two conditions are sunset and trees without leaves and dark clouds and trees without leaves. Two test sets, "Diag1" and "Diag2" are obtained at all three conditions. All the conditions are given in Table 5.4.

Table 5.4 Different Test Conditions for Location "T"

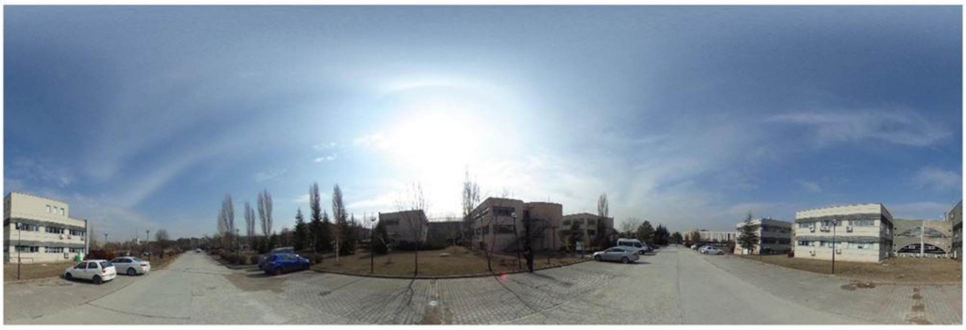
T	Noon	Afternoon	Sunset	Dark Clouds
Trees Without Leaves	Map, Diag1, Diag2		Diag1, Diag2*	Diag1, Diag2

* The last 3 photos are absent in Diag2.

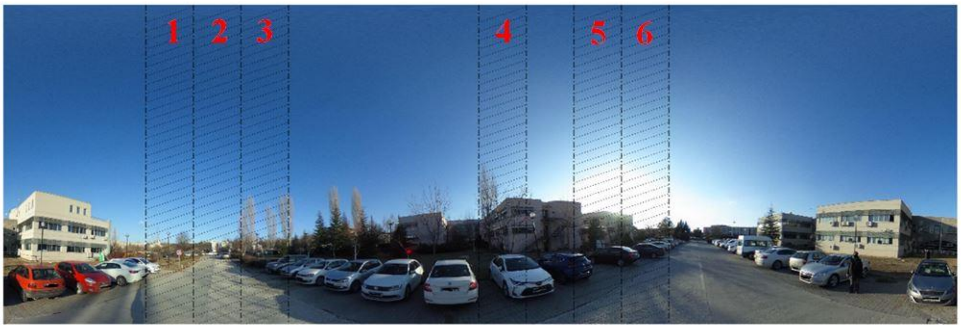
For the comparison of skylines, while “T(Diag1,5,NoLeaves-Noon)” is used as reference signal, “T(Diag1,5,NoLeaves-Sunset)” and “T(Diag1,5,NoLeaves-DarkClouds)” are used for different conditions. Photographs are shown in Figure 5.17. Skyline signals are given in Figure 5.18.

Figure 5.17.b and corresponding skyline signal Figure 5.18a have six unusable slices. For unusable slice numbers 1 and 3, when the sun hits against trees without leaves, skyline is distorted in “T(Diag1,5,NoLeaves-Sunset)”. The sun is also responsible for unusable slice number 2 causing the skyline of white building to be problematic in “T(Diag1,5,NoLeaves-Sunset)”. For unusable slice numbers 4, 5 and 6, the skyline is distorted due to sun positions.

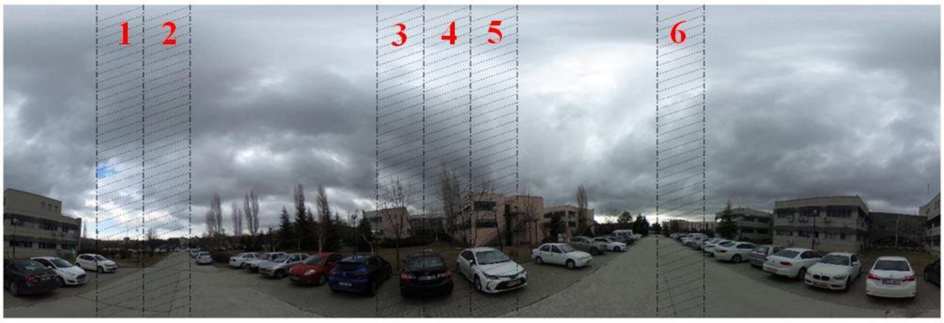
There are six unusable slices in Figure 5.17.c and corresponding skyline signal Figure 5.18.b. Unusable slices 1 and 2 are eliminated due to skyline distortion caused by small piece of clear weather and trees without leaves in “T(Diag1,5,NoLeaves-DarkClouds)”. Dark clouds cause a huge distortion in skyline for unusable slice numbers 3, 4 and 5 in “T(Diag1,5,NoLeaves-DarkClouds)”. Finally, white building in “T(Diag1,5,NoLeaves-Noon)” and dark clouds in “T(Diag1,5,NoLeaves-DarkClouds)” cause skylines to not match with each other leading to unusable slice number 6 to be eliminated.



a) “T(Diag1,5,NoLeaves-Noon)”

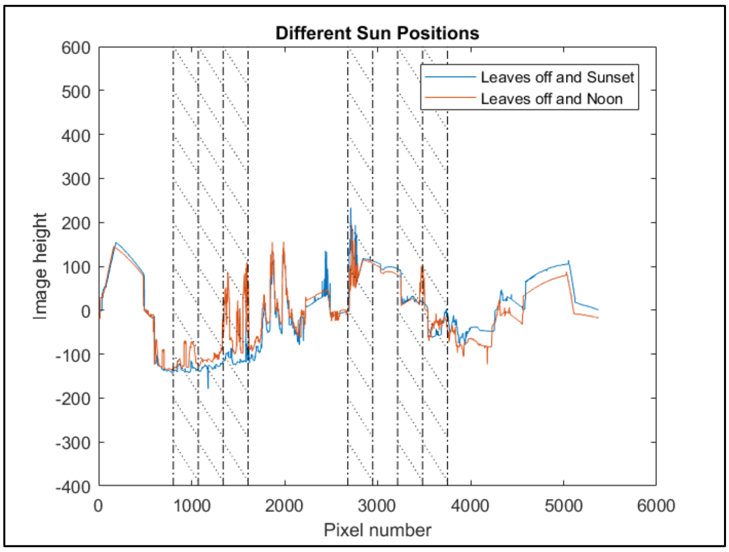


b) “T(Diag1,5,NoLeaves-Sunset)”

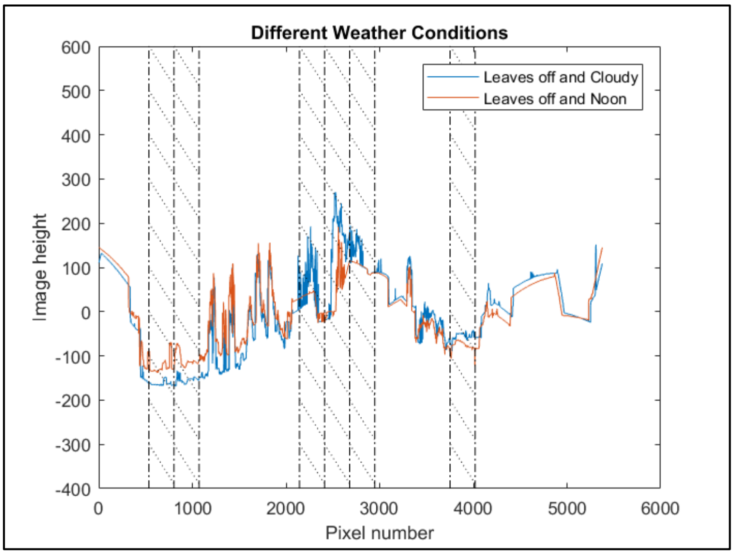


c) “T(Diag1,5,NoLeaves-DarkClouds)”

Figure 5.17. Photographs Captured At Different Conditions for Test Location “T”



a) “T(Diag1,5,NoLeaves-Noon)” and “T(Diag1,5,NoLeaves-Sunset)”



b) “T(Diag1,5,NoLeaves-Noon)” and “T(Diag1,5,NoLeaves-DarkClouds)”

Figure 5.18. Skyline Signals of Photographs Captured At Different Conditions for Test Location “T”

5.2.5 Test Location “U”

Map condition for location “U” is afternoon and trees with leaves. Four test sets, “Diag1”, “Diag2”, “Horz”, “VertL”, are also obtained at the same condition. Test sets “Diag1” and “Diag2” are obtained at conditions noon and trees without leaves, sunset and trees without leaves, dark clouds and trees without leaves. All the conditions are given in Table 5.5.

Table 5.5 Different Test Conditions for Location “U”

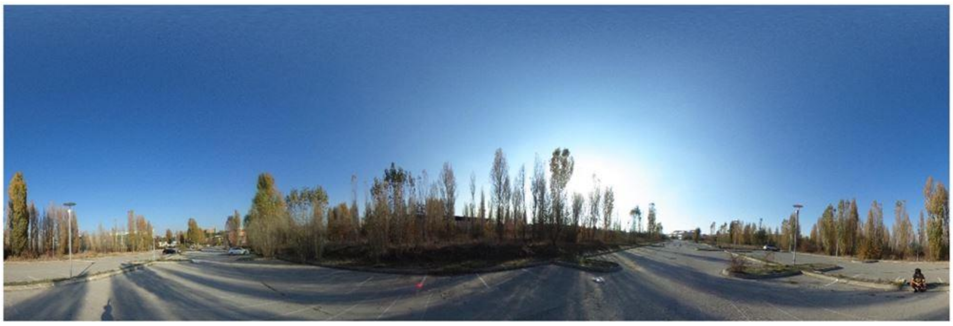
U	Noon	Afternoon	Sunset	Dark Clouds
Trees Without Leaves	Diag1, Diag2		Diag1, Diag2*	Diag1, Diag2
Trees With Leaves		Map, Diag1, Diag2, Horz, VertL		

* The last 3 photos are absent in Diag2.

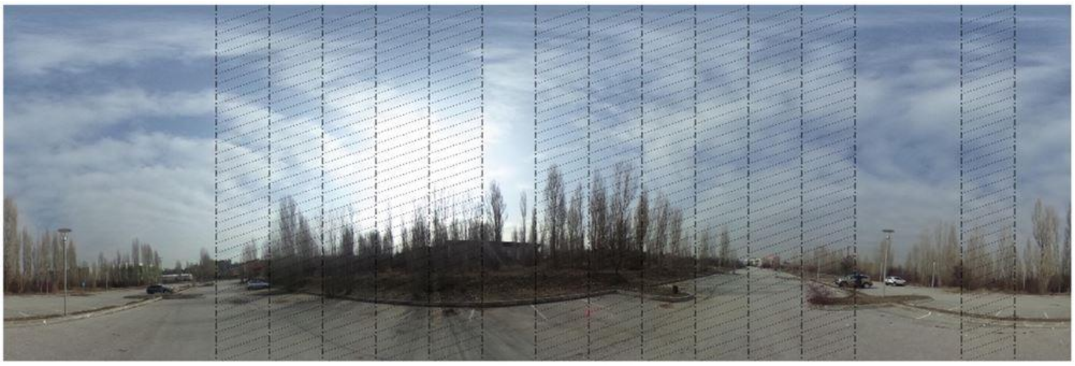
For the comparison of skylines, while “U(Diag1,5,Leaves-Afternoon)” is used as reference signal, “U(Diag1,5,NoLeaves-Noon)”, “U(Diag1,5,NoLeaves-Sunset)”, “U(Diag1,5,NoLeaves-DarkClouds)” are used for different conditions. Photographs are shown in Figure 5.19. Skyline signals are given in Figure 5.21. Also “U(Diag2,5,Leaves-Afternoon)” is used as another reference signal for comparison of “U(Diag2,5,NoLeaves-DarkClouds)”. Photographs and skyline signals of “U(Diag2,5,Leaves-Afternoon)” and “U(Diag2,5,NoLeaves-DarkClouds)” are given in Figure 5.20 and Figure 5.22.

Usable slice numbers are less than other environments with the same conditions as explained above in previous sections. The reason is the absence of buildings with distinct and unchanging features. Therefore, seasonal changes at trees cause drastic changes in skylines. Seasonal changes and the direct sun position are the common cause of unusable slices in Figure 5.19.b, Figure 5.19.c, Figure 5.19.d and Figure 5.20.b and corresponding skyline signals in Figure 5.21.a, Figure 5.21.b, Figure 5.21.c and Figure 5.22.

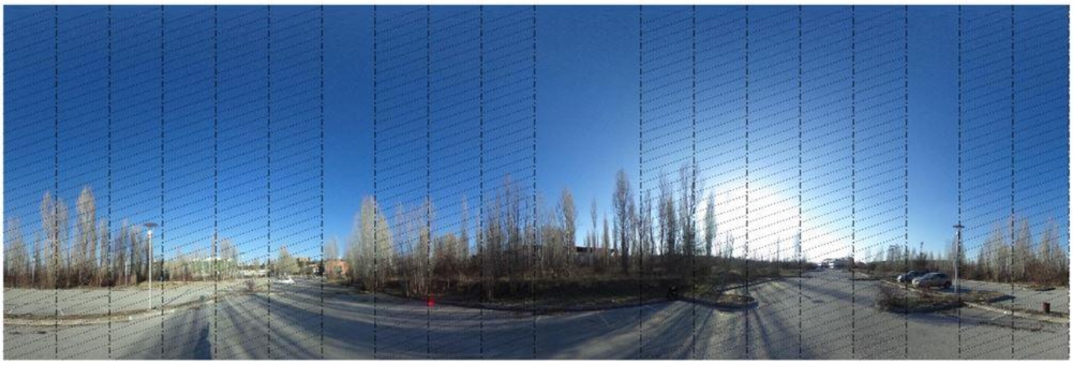
However, there are also other reasons. For Figure 5.19.c, for the first 4-5 slices, the sun position indirectly changes skyline when it hits against the trees. For Figure 5.19.d, despite of dark clouds, a little bit of sun, being at the same position as in Figure 5.19.a, rescues the skyline for the first slices. Absence of the sun in Figure 5.20.b changes the skyline for the first slices in spite of having the same conditions as in Figure 5.19.d. Therefore, the first few slices are eliminated just like in Figure 5.19.c.



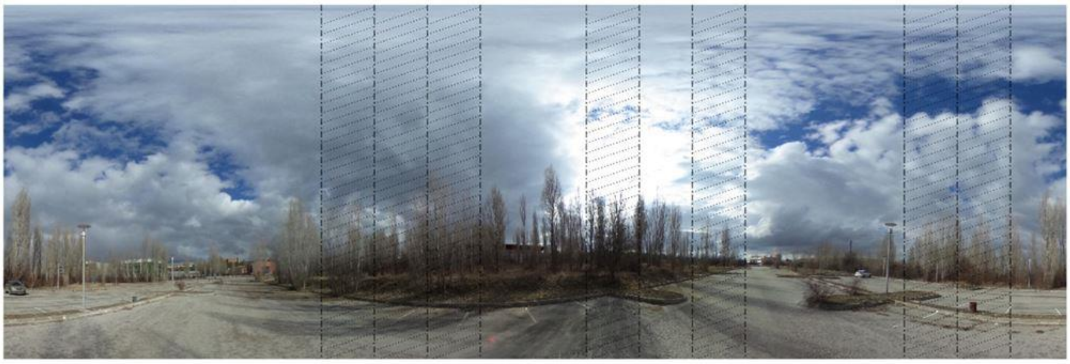
a) “U(Diag1,5,Leaves-Afternoon)”



b) “U(Diag1,5,NoLeaves-Noon)”

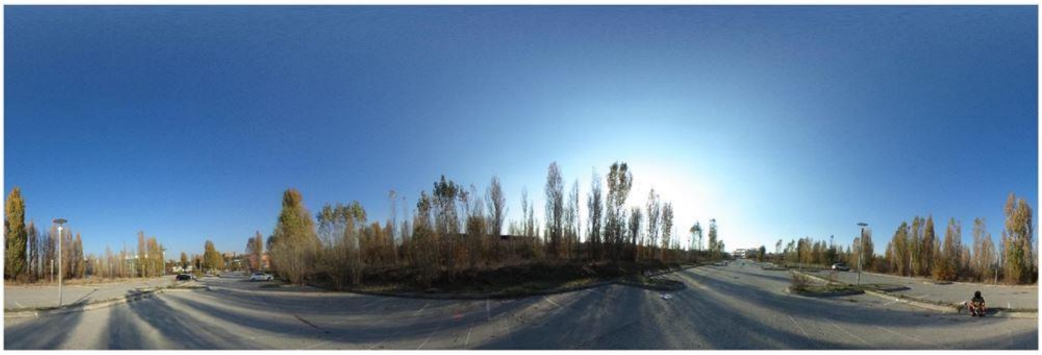


c) “U(Diag1,5,NoLeaves-Sunset)”

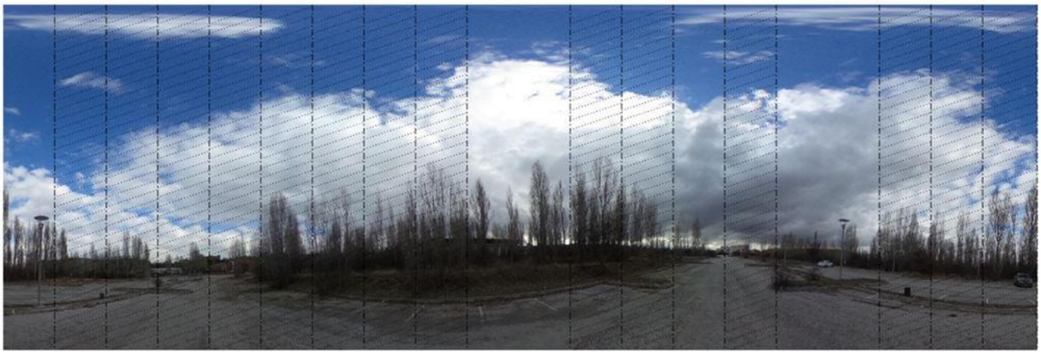


d) “U(Diag1,5,NoLeaves-DarkClouds)”

Figure 5.19. Photographs Captured At Different Conditions for Test Location “U”

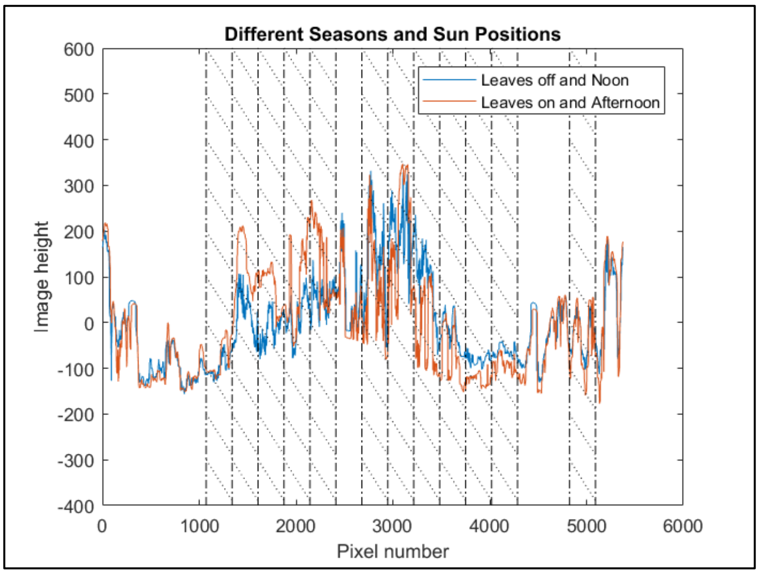


a) “U(Diag2,5,Leaves-Afternoon)”

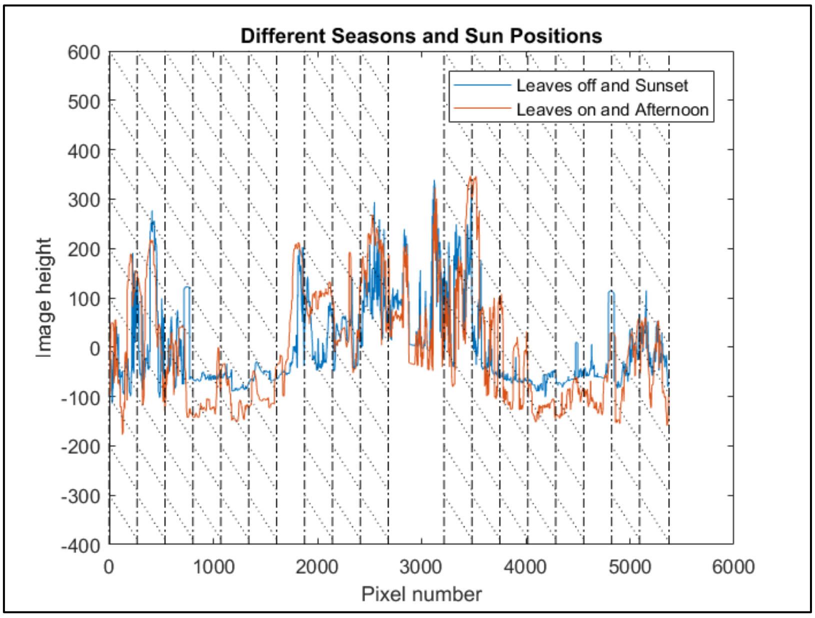


b) “U(Diag2,5,NoLeaves-DarkClouds)”

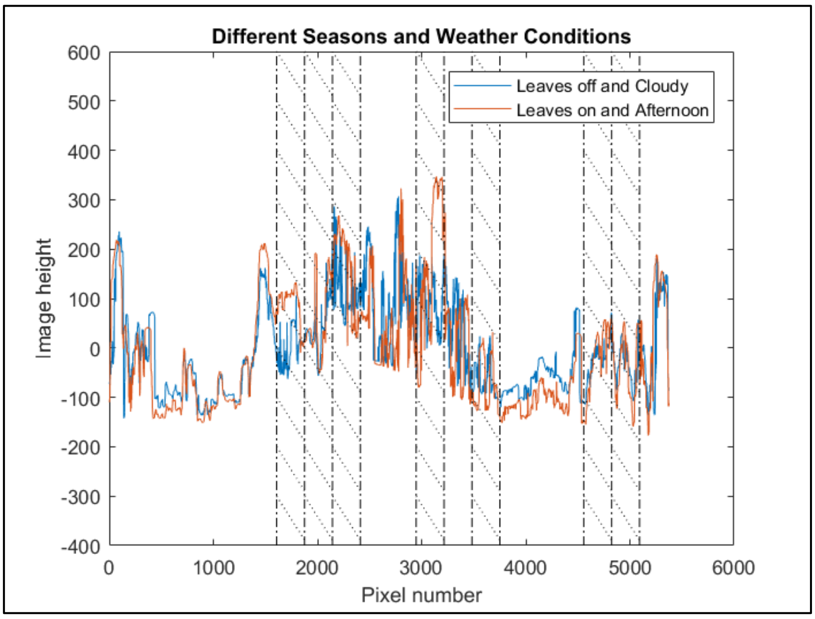
Figure 5.20. Photographs Captured At Different Conditions for Test Location “U”



a) “U(Diag1,5,Leaves-Afternoon)” and “U(Diag1,5,NoLeaves-Noon)”



b) “U(Diag1,5,Leaves-Afternoon)” and “U(Diag1,5,NoLeaves-Sunset)”



c) “U(Diag1,5,Leaves-Afternoon)” and “U(Diag1,5,NoLeaves-DarkClouds)”

Figure 5.21. Skyline Signals of Photographs Captured At Different Conditions for Test Location “U”

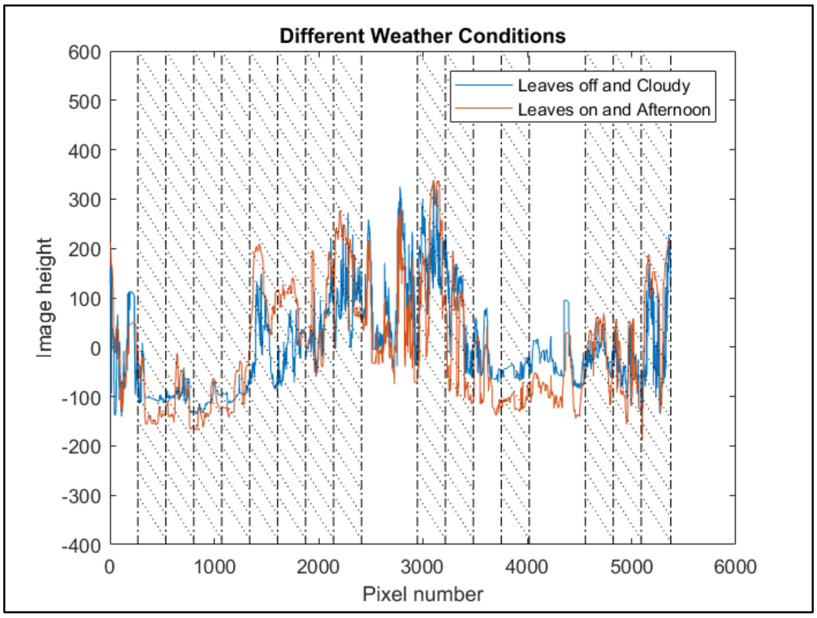


Figure 5.22. Skyline Signals of Photographs Captured At Different Conditions for Test Locations “U”, “U(Diag2,5,Leaves-Afternoon)” and “U(Diag2,5,NoLeaves-DarkClouds)”

5.3 Artificial Occlusions

Linear, sinusoidal and triangular occlusions with variable proximity to skylines are added to signals with increasing intensity in order to observe when the localization information obtained using the algorithm fails to give reliable results. These occlusion shapes are chosen to be random and challenging for algorithm in terms of its ability for distinguishing unusable parts.

While “G”, “K” and “T” have occlusion cases to cover most of their skylines, “B” and “U” have only sample cases. The three environments were considered as sufficient in order to observe failure point of the algorithm. As can be seen from the graphs, the algorithm was successful to eliminate occlusions from skylines. The results for localization will be examined in next section.

5.3.1 Test Location “B”

Occlusions are added to “B(Diag1,5,NoLeaves-Noon)”. The occlusion and excluded version of skylines are given in Figure 5.23.

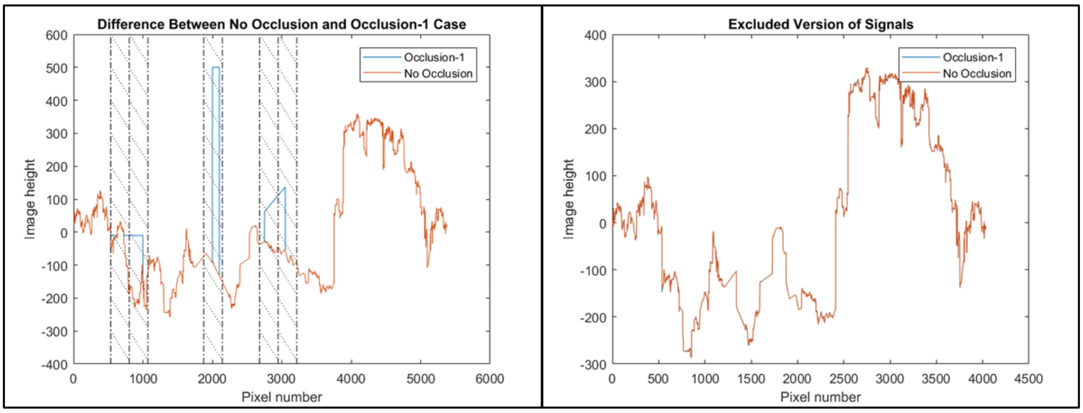


Figure 5.23. Occlusion and Excluded Version of Skyline Signal “B(Diag1,5,NoLeaves-Noon)”

5.3.2 Test Location “G”

Occlusions are added to “G(Diag1,5,Leaves-Afternoon)”. The amount of occlusions and as a result of that, number of usable slices is given in Table 5.6. The occlusion and excluded version of skylines are given in Figure 5.24 and Figure 5.25. In occlusion-2 case as can be seen from Figure 5.25, although the last slice is occluded, it was included to calculation. The reason is that it was similar to the non-occluded version of signal slice within the limits of allowable shift as explained in METHOD.

Table 5.6 The Amount of Occlusion and Number of Usable Slices for “G”

	The Amount of Occlusion	Number of Usable Slices
Occlusion 1	20.4%	15/20
Occlusion 2	64.5%	5/20

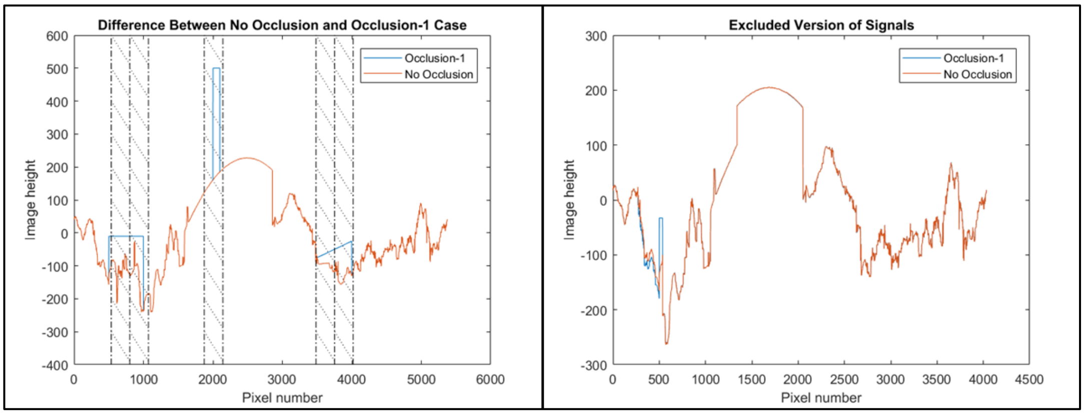


Figure 5.24. Occlusion-1 and Excluded Version of Skyline Signal "G(Diag1,5,Leaves-Afternoon)"

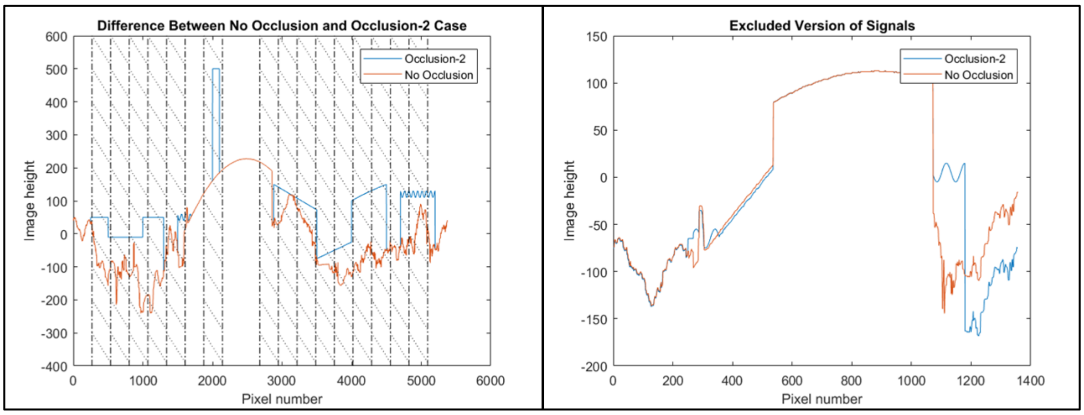


Figure 5.25. Occlusion-2 and Excluded Version of Skyline Signal "G(Diag1,5,Leaves-Afternoon)"

5.3.3 Test Location "K"

Occlusions are added to "K(Diag1,5,Leaves-Sunrise)". The amount of occlusions and as a result of that, number of usable slices is given in Table 5.7. The occlusion and excluded version of skylines are given in Figure 5.26, Figure 5.27, Figure 5.28 and Figure 5.29. In Occlusion-1 and 2 cases as can be seen from Figure 5.26 and Figure 5.27, although one slice is occluded, it was included to calculation as shown in figures. The reason is that it was similar to the non-occluded version of signal slice within the limits of allowable shift. A similar situation also occurs in Occlusion-2 and 3 cases for other two slices shown in Figure 5.27 and Figure 5.28. For those cases, the shift is also very small and the structure of occlusion is very similar to the skyline just below it. Also, the occlusion is not far from the skyline; therefore, mean difference is within allowable limits. Although this may be considered as a downside of algorithm, it is not expected to affect localization in an adverse way since that part of occluded signal is also expected to appear usable when compared with other map signals. For Occlusion-4 case a slice becomes usable even though it is unusable for Occlusion-2 and 3 cases as shown in Figure 5.29. The reason is the summation of similarity values as explained in METHOD. The highest similarity value between occlusion and non-occlusion case should be obtained when the shift is equal to zero meaning that the maximum value of summation of similarity values should be 2688 as shown in Figure 5.30 for Occlusion-3 case. However, since the amount of occlusions is high, indices of three maximum values of summation of similarity values are 3025, 1446 and 2235 as shown in Figure 5.31; and the highest similarity is obtained when the shift is 337 pixels (3025-2688). The value of 2689 is not calculated since it is smaller than 0.85*151.9=129.1. In conclusion, the slice shown in Figure 5.29 becomes similar to the non-occlusion slice as it is shifted 337 pixels.

Table 5.7 The Amount of Occlusion and Number of Usable Slices for “K”

	The Amount of Occlusion	Number of Usable Slices
Occlusion 1	4.6%	19/20
Occlusion 2	32.5%	15/20
Occlusion 3	67.4%	8/20
Occlusion 4	83.2%	2/20

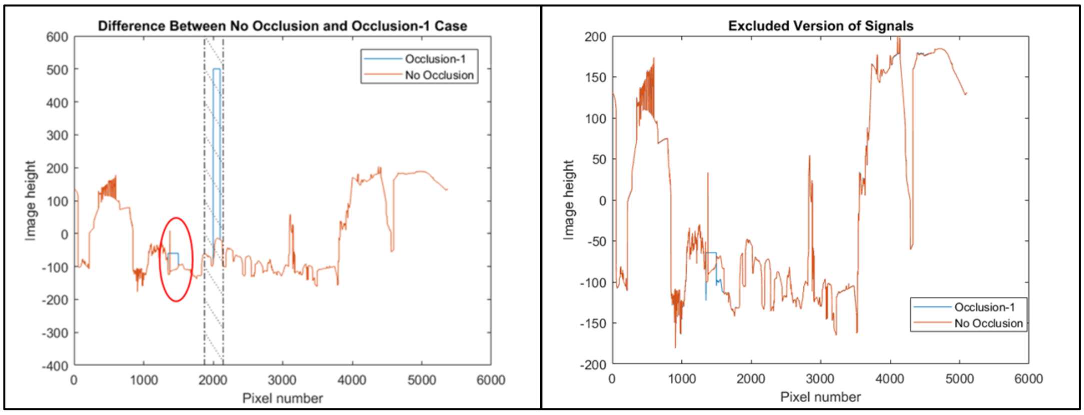


Figure 5.26. Occlusion-1 and Excluded Version of Skyline Signal
“K(Diag1,5,Leaves-Sunrise)”

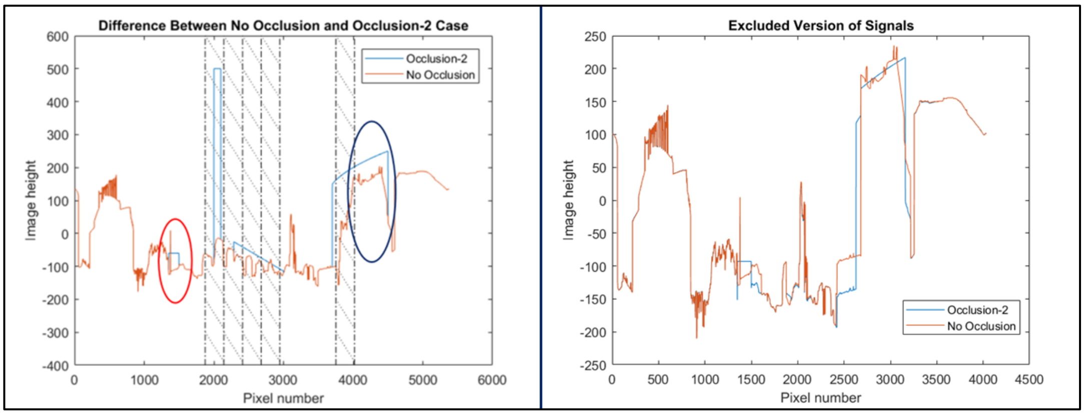


Figure 5.27. Occlusion-2 and Excluded Version of Skyline Signal
“K(Diag1,5,Leaves-Sunrise)”

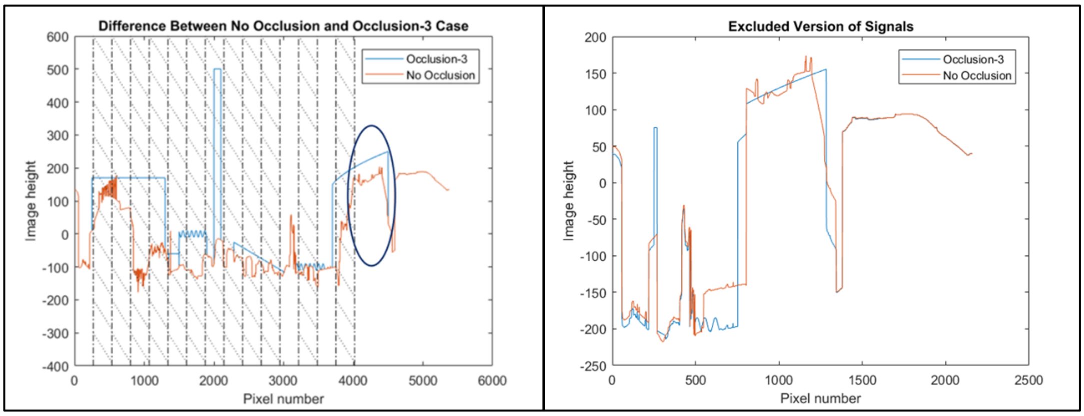


Figure 5.28. Occlusion-3 and Excluded Version of Skyline Signal
“K(Diag1,5,Leaves-Sunrise)”

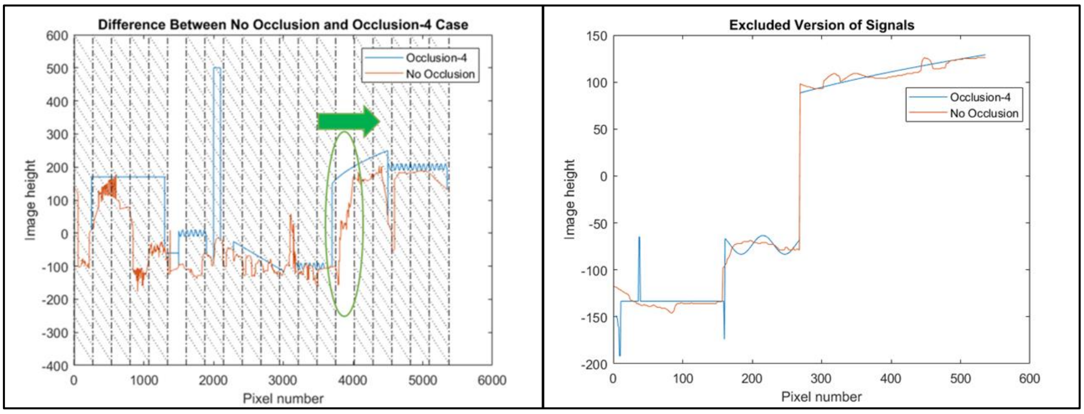


Figure 5.29. Occlusion-4 and Excluded Version of Skyline Signal "K(Diag1,5,Leaves-Sunrise)"

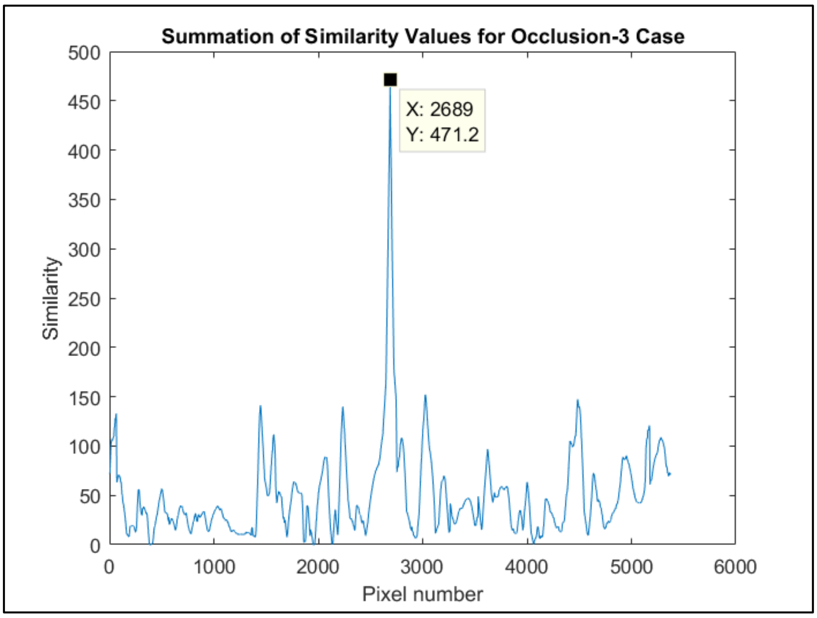


Figure 5.30. Summation of Similarity Values for "K(Diag1,5,Leaves-Sunrise)"

Occlusion-3 Case

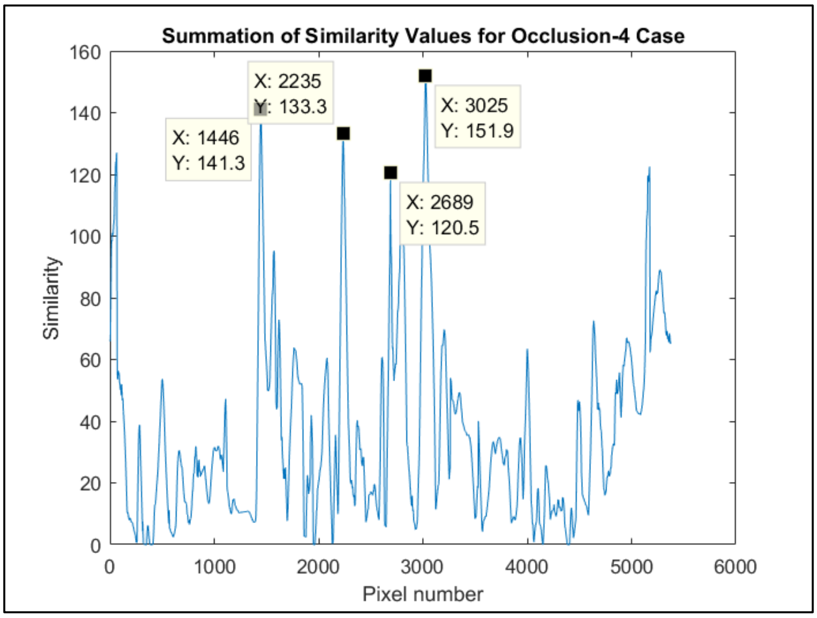


Figure 5.31. Summation of Similarity Values for “K(Diag1,5,Leaves-Sunrise)” Occlusion-4 Case

5.3.4 Test Location “T”

Occlusions are added to “T(Diag1,5,NoLeaves-Noon)”. The amount of occlusions and as a result of that, number of usable slices is given in Table 5.8. The occlusion and excluded version of skylines are given in Figure 5.32, Figure 5.33, Figure 5.34, Figure 5.35 and Figure 5.36.

Table 5.8 The Amount of Occlusion and Number of Usable Slices for “T”

	The Amount of Occlusion	Number of Usable Slices
Occlusion 1	14.1%	15/20
Occlusion 2	21.6%	13/20
Occlusion 3	35.3%	9/20
Occlusion 4	55.0%	8/20
Occlusion 5	74.8%	3/20

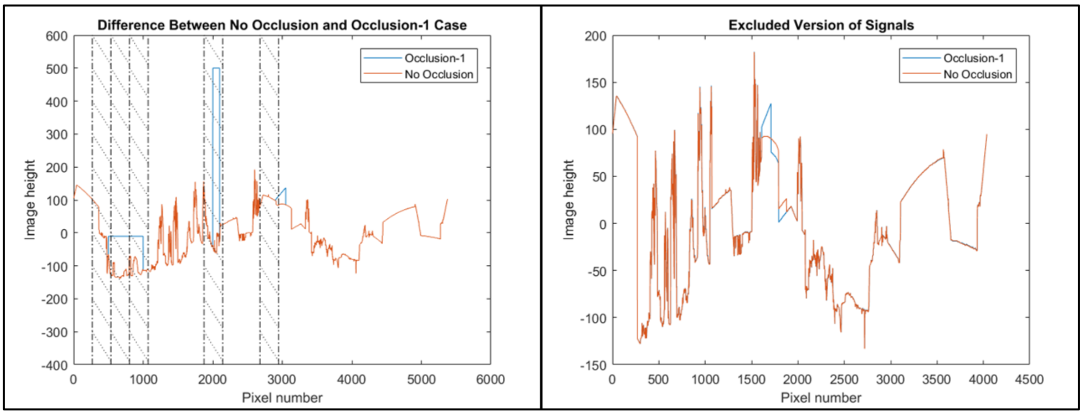


Figure 5.32. Occlusion-1 and Excluded Version of Skyline Signal
“T(Diag1,5,NoLeaves-Noon)”

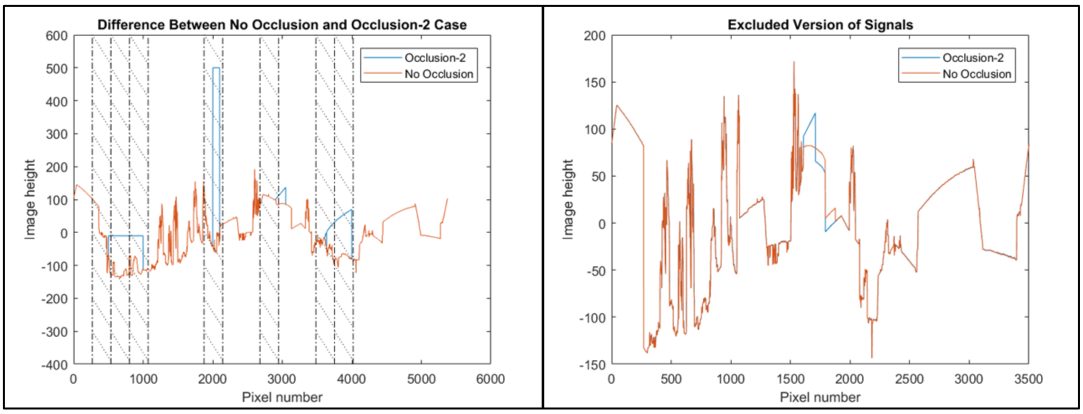


Figure 5.33. Occlusion-2 and Excluded Version of Skyline Signal

“T(Diag1,5,NoLeaves-Noon)”

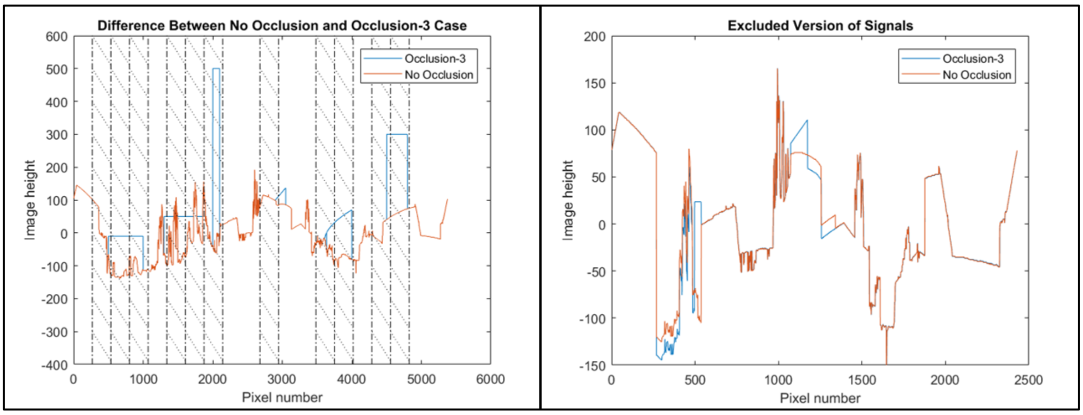


Figure 5.34. Occlusion-3 and Excluded Version of Skyline Signal
“T(Diag1,5,NoLeaves-Noon)”

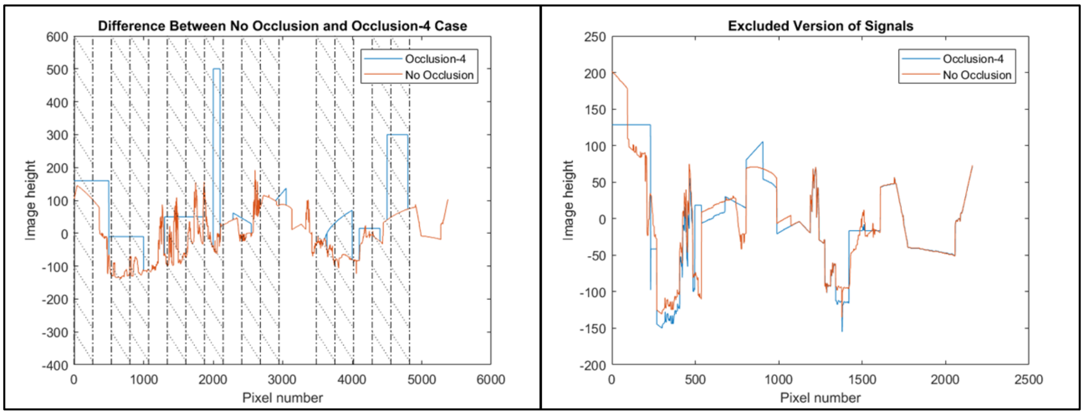


Figure 5.35. Occlusion-4 and Excluded Version of Skyline Signal “T(Diag1,5,NoLeaves-Noon)”

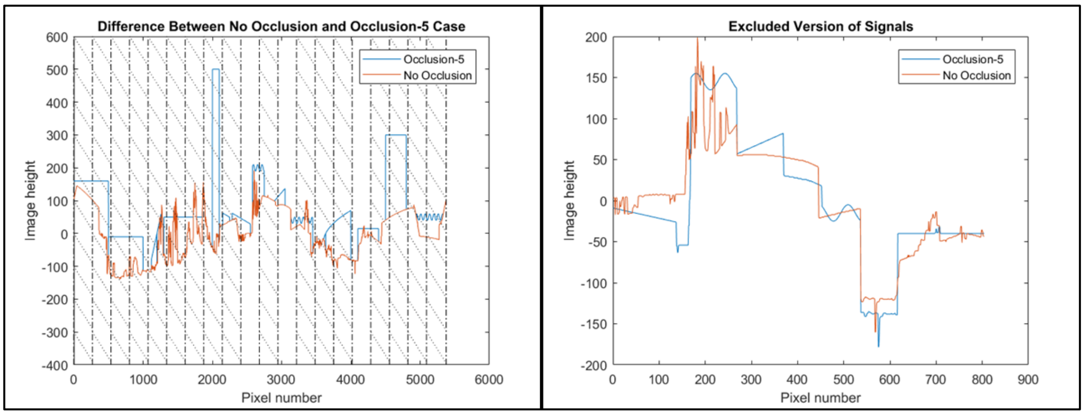


Figure 5.36. Occlusion-5 and Excluded Version of Skyline Signal “T(Diag1,5,NoLeaves-Noon)”

5.3.5 Test Location “U”

Occlusions are added to “U(Diag1,5,Leaves-Afternoon)”. The occlusion and excluded version of skylines are given in Figure 5.37.

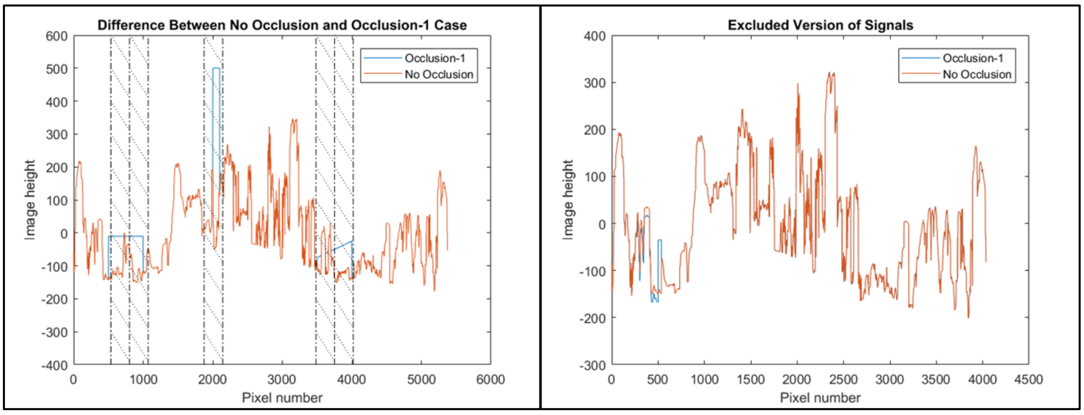


Figure 5.37. Occlusion-1 and Excluded Version of Skyline Signal "U(Diag1,5,Leaves-Afternoon)"

5.4 Results

As stated earlier, localizations of 1394 test skyline signals are performed in five different maps. While 413 of them are natural skyline signals meaning that they are directly obtained from photographs captured at different locations and times using spherical camera, 981 skyline signals are formed with artificial occlusions added to natural skyline signals. As explained in METHOD, as the skyline signals are obtained, in order to execute localization, firstly similarity values of a test signal compared with the map signals are found. Using these values, localization is executed. Apart from this method, an alternative method was also explained. Maximum value of summation of similarity values of slices ("Sum_max") as shown in Figure 4.13 can also be used for localization. For both methods, localization is achieved using the map point with the highest similarity value as the center of 2 meters x 2 meters localization square as explained in Figure 4.1.If the 2 meters x 2 meters square does not include the test point, then this localization is labelled as unsuccessful. Numbers of successful localizations only for captured test photographs –occluded ones are not included– in different environments and conditions with two different methods are given in Table 5.9. Also, for similarity comparison method, averages of usable slice numbers and for “Sum_max” comparison method, averages of “Sum_max” values are given.

Table 5.9 Numbers of Successful Localizations for Captured Test Photographs

Cases	Numbers of Successful Localizations (Similarity Comparison)	Average of Usable Slice Numbers (Similarity Comparison)	Numbers of Successful Localizations (“Sum_max” comparison)	Average of “Sum_max” (“Sum_max” comparison)
B(41,Leaves-Sunset)	16/16	19.9/20	16/16	1142.7
B(74,Leaves-Sunset)	16/16	20/20	16/16	1288.1
B(Diag1,NoLeaves-Noon)	9/10	13.3/20	10/10	719.8
B(Diag2,NoLeaves-Noon)	10/10	13.7/20	10/10	757.2
B(Diag1,NoLeaves-Afternoon)	3/8[1]	7.5/20	2/8	388.3
B(Diag2,NoLeaves-Afternoon)	4/7[1]	8/20	4/7	469.4
B(Diag1,NoLeaves-DarkClouds)	0/5[2]	11.2/20	1/5	582.0
B(Diag2,NoLeaves-DarkClouds)	5/7[2]	12/20	4/7	580.0
G(Diag1,Leaves-Afternoon)	10/10	19.8/20	10/10	1229.6
G(Diag2,Leaves-Afternoon)	10/10	19.5/20	10/10	1211.5

Table 5.9 cont'd

G(Horz,Leaves-Afternoon)	10/10	20/20	10/10	1282.1
G(VertL,Leaves-Afternoon)	10/10	19.8/20	10/10	1250.7
G(Diag1,NoLeaves-Noon)	10/10	14.1/20	10/10	848.8
G(Diag2,NoLeaves-Noon)	10/10	12.7/20	10/10	818.8
G(Diag1,NoLeaves-Sunset)	10/10	13.6/20	10/10	864.2
G(Diag2,NoLeaves-Sunset)	10/10	13.3/20	10/10	794.6
G(Diag1,NoLeaves-DarkClouds)	10/10	14.5/20	10/10	869.0
G(Diag2,NoLeaves-DarkClouds)	10/10	13.3/20	9/10	821.2
K(Diag1,Leaves-Sunrise)	10/10	19.5/20	10/10	1302.3
K(Diag2,Leaves-Sunrise)	10/10	19.7/20	10/10	1320.7
K(Horz,Leaves-Sunrise)	10/10	19.6/20	10/10	1326.1
K(VertR,Leaves-Sunrise)	10/10	19.4/20	9/10	1266.2
K(Diag1,Leaves-Cloudy)	10/10	19.8/20	10/10	1254.1
K(Diag2,Leaves-Cloudy)	10/10	19.6/20	10/10	1289.9

Table 5.9 cont'd

K(Horz,Leaves-Cloudy)	10/10	19.7/20	9/10	1335.1
K(VertR,Leaves-Cloudy)	6/10[3]	18.5/20	8/10	1213.3
T(Diag1,NoLeaves-Noon)	10/10	19.4/20	10/10	1129.2
T(Diag2,NoLeaves-Noon)	10/10	18.9/20	10/10	1178.1
T(Diag1,NoLeaves-Sunset)	9/10	13.6/20	10/10	830.6
T(Diag2,NoLeaves-Sunset)	7/7	14.6/20	7/7	865.6
T(Diag1,NoLeaves-DarkClouds)	10/10	15/20	10/10	915.1
T(Diag2,NoLeaves-DarkClouds)	9/10	12.5/20	10/10	728.4
U(Diag1,Leaves-Afternoon)	10/10	19.3/20	10/10	1004.6
U(Diag2,Leaves-Afternoon)	10/10	18.7/20	10/10	1030.3
U(Horz,Leaves-Afternoon)	10/10	17.9/20	10/10	1001.4
U(VertL,Leaves-Afternoon)	10/10	18.8/20	10/10	985.0
U(Diag1,NoLeaves-Noon)	8/10	7.9/20	10/10	370.1
U(Diag2,NoLeaves-Noon)	9/10	7.5/20	9/10	369.0

Table 5.9 cont'd

U(Diag1,NoLeaves-Sunset)	6/10	4.3/20	7/10	204.1
U(Diag2,NoLeaves-Sunset)	6/7	4/20	7/7	218.6
U(Diag1,NoLeaves-DarkClouds)	8/10	8.4/20	9/10	394.0
U(Diag2,NoLeaves-DarkClouds)	6/10	4.4/20	10/10	198.9

The downside of "Sum_max" method is the absence of a maximum value to compare "Sum_max" values unlike comparison of usable slice numbers to maximum usable slice number of 20 for similarity comparison method. For "Sum_max" method, experimentally a maximum value can be obtained using the test data. Even though "Sum_max" method seems to be disadvantageous, it actually gives good localizations and "Sum_max" values decreases as the usable slice number decreases as shown in Figure 5.38.

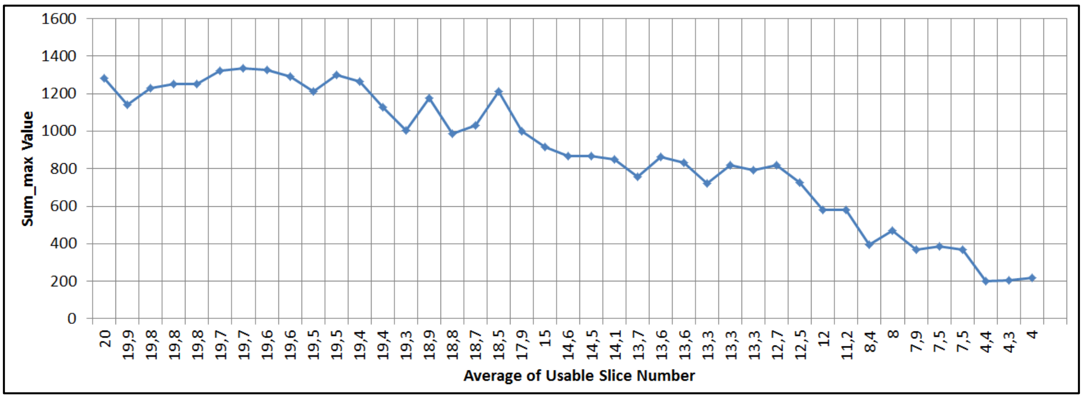


Figure 5.38. "Sum_max" Values Compared With Average of Usable Slice Numbers

Average of usable slice number is between 20 and 17.9 for test photographs captured at the same time with the map and for the environment "K" with conditions trees with leaves and cloudy. The fact that “K(Leaves-Cloudy)” gives a high average of usable slice numbers show that change in weather conditions on its own does not dramatically distort skyline as explained in Experiment Conditions - Test Location “K”.

Average of usable slice number is between 15 and 11.2 for environments “B(NoLeaves-Noon)”, “B(NoLeaves-DarkClouds)”, “G(NoLeaves-Noon)”, “G(NoLeaves-Sunset)”, “G(NoLeaves-DarkClouds)”, “T(NoLeaves-Sunset)” and “T(NoLeaves-DarkClouds)”. As explained in Experiment Conditions - Test Location “B”, Test Location “G”, and Test Location “T”, all 7 cases have 13-14 usable slice numbers for test point “5”. The results match with those values.

Average of usable slice number is between 8.4 and 7.5 for environments “B(NoLeaves-Afternoon)”, “U(NoLeaves-Noon)” and “U(Diag1,NoLeaves-DarkClouds)”. As shown in Experiment Conditions - Test Location “B” and Test Location “U”, the three cases have 7, 8, and 13 usable slice numbers respectively for test point “5”. Except for “U(Diag1,NoLeaves-DarkClouds)”, the numbers match with results. The reason why “U(Diag1,NoLeaves-DarkClouds)” does not match is that for environment “U”, the mean of skylines shift a lot due to seasonal changes in trees and similarity values of slices are lower than other environments. Therefore, small differences in mean or similarity values cause slices to be unusable. As a result, changes in usable slice numbers for “U(Diag1,NoLeaves-DarkClouds)” occur. Also, the last two test points of “U(Diag1,NoLeaves-DarkClouds)” have usable slice numbers of 3 and 4 due to the absence of the sun as explained in Experiment Conditions - Test Location “U”.

Average of usable slice number is between 4.4 and 4 for environments “U(NoLeaves-Sunset)” and “U(Diag2,NoLeaves-DarkClouds)”. As shown in Experiment Conditions - Test Location “U”, the two cases have 4 and 6 usable slice numbers respectively for test point “5”. These values also match with each other.

As can be seen in Table 5.9, as the usable slice number gets smaller, number of successful localizations also gets smaller. This can be seen more clearly as test data with occlusions are investigated. However, there are some cases that are not in correlation with this fact.

1) “B(Diag1,NoLeaves-Afternoon)” and “B(Diag2,NoLeaves-Afternoon)”

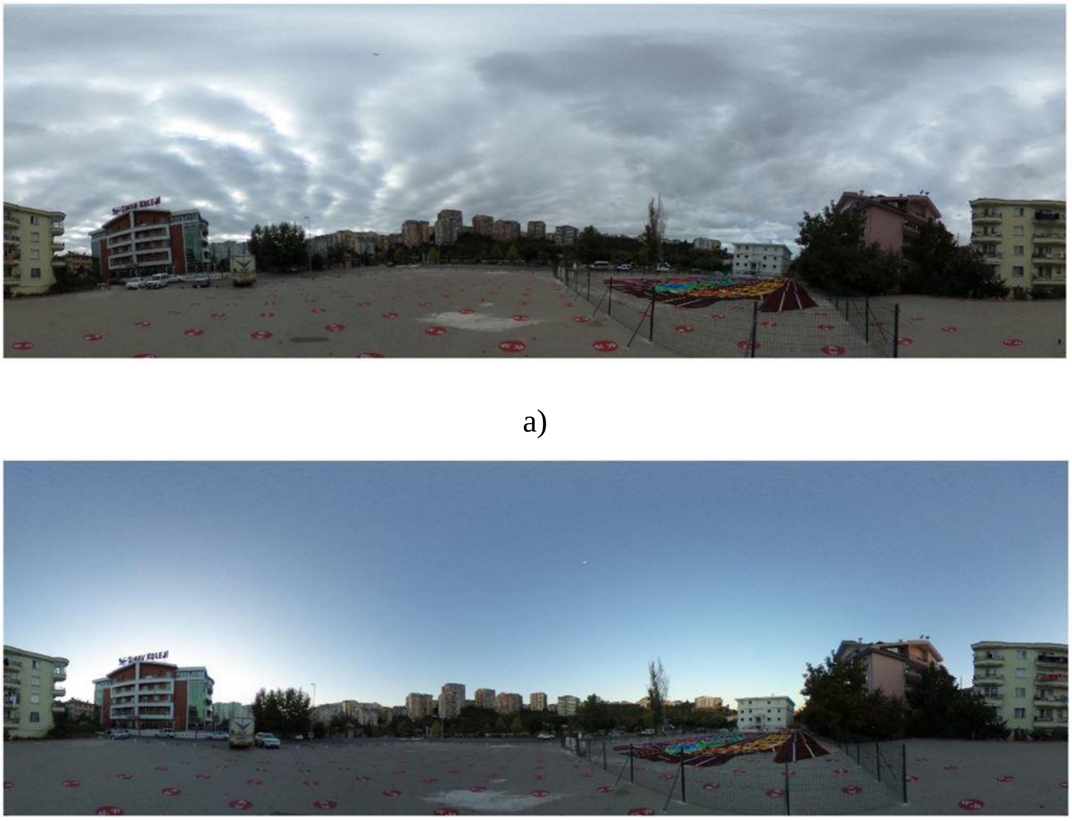
   Numbers of successful localizations are not similar to the ones with similar average of usable slice numbers, specifically “U(NoLeaves-Noon)” and “U(Diag1,NoLeaves-DarkClouds)”. As explained in Experiment Conditions - Test Location “B”, S-like shape confuses the algorithm as shown in Figure 5.9.

2) “B(Diag1,NoLeaves-DarkClouds)” and “B(Diag2,NoLeaves-DarkClouds)”

   Numbers of successful localizations are not similar to the ones with similar average of usable slice numbers. In spite of having similar usable slice numbers with case “B(NoLeaves-Noon)”, “B(NoLeaves-DarkClouds)” has low numbers of successful localizations. The biggest reason is downsizing of the building by conserving its outer features as indicated in Figure 5.8.c. Also S-like shape in horizon has effects, too.

3) ”K(VertR,Leaves-Cloudy)”

   Numbers of successful localizations are not similar to the ones with similar average of usable slice numbers. While obtaining good numbers of successful localizations for other test points, the reason of having four unsuccessful localizations for ”K(VertR,Leaves-Cloudy)” is S-like shape on horizon in test photographs. ”K(VertR,5,Leaves-Cloudy)” and its one of correct localization points “K(Map,28,Leaves-Sunrise)” are given in Figure 5.39. As given in Figure 5.40, although “K(Map,28,Leaves-Sunrise)” is the correct point, when compared with the test photograph, the last five unusable slices are eliminated due to high mean differences. Not one slice of incorrect localization point “K(Map,39,Leaves-Sunrise)” is eliminated as given in Figure 5.41 and since “K(Map,28,Leaves-Sunrise)” and “K(Map,39,Leaves-Sunrise)” are very close to each other, algorithm gives incorrectly “K(Map,39,Leaves-Sunrise)” as localization point. Even though not all correct localization points are incorrectly reduced in size, these S-like shapes on horizon in photographs still confuse the algorithm causing incorrect localization points.



a)

b)

Figure 5.39. Photographs of a) ”K(VertR,5,Leaves-Cloudy)” and b) “K(Map,28,Leaves-Sunrise)”

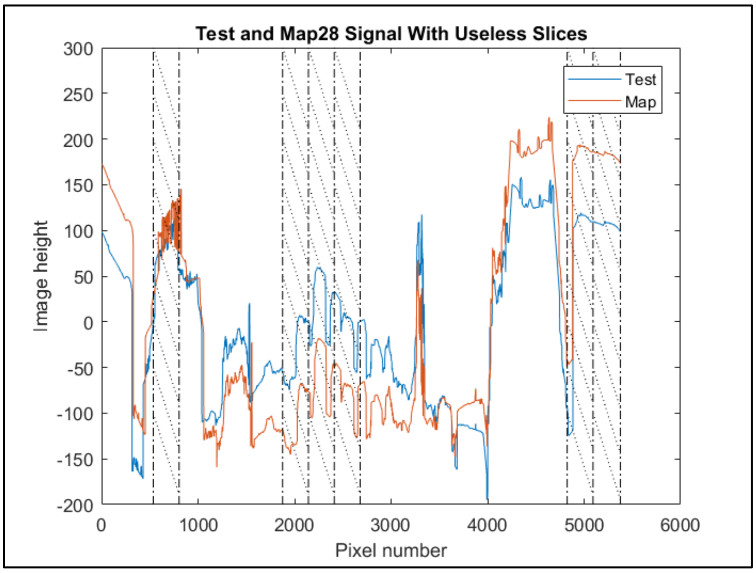


Figure 5.40. Skyline Signals of ”K(VertR,5,Leaves-Cloudy)” and “K(Map,28,Leaves-Sunrise)”

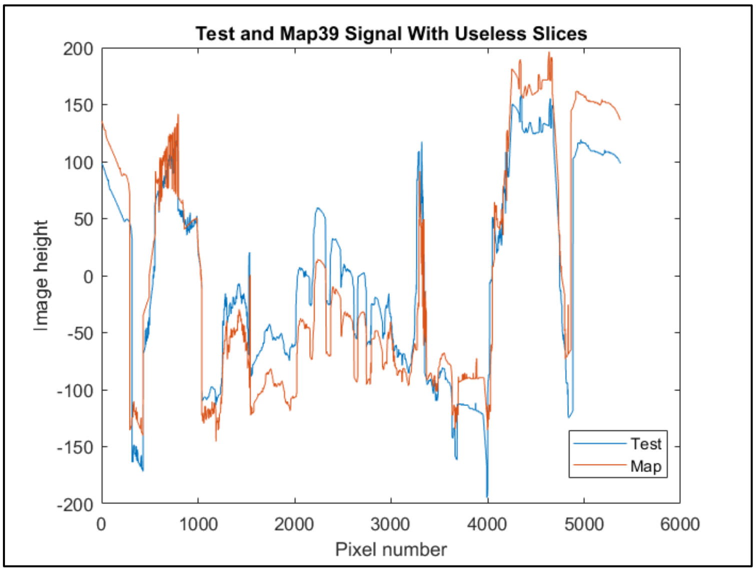


Figure 5.41. Skyline Signals of ”K(VertR,5,Leaves-Cloudy)” and “K(Map,39,Leaves-Sunrise)”

Examining the test photographs and environments, one can conclude that change in weather conditions on its own have the least effect on numbers of successful localizations. While the effect of change in sun position and seasons is not to be underestimated, the greatest factor is the change in sun position with a white building and downsizing of a building while preserving its outer features. Also seasonal changes affect the localization the most when skyline is formed only with trees.

Considering all 1394 test data, including occluded versions of signals, a relation between number of unsuccessful localizations and usable slice number can be obtained as given in Table 5.10. Relation between number of unsuccessful localizations and “Sum_max” values is also given in Table 5.11 along with the average of distances of false localization points to the closest correct localization points. For example, for ”X(Diag1,1)” test point, the algorithm gives ”X(Map,24)”. The closest correct localization to ”X(Map,24)” would be ”X(Map,21)”. Therefore, the value is 1 unit. According to results given in Table 5.10 and Table 5.11, there is a sharp increase in number of unsuccessful localizations below usable slice number 8 or “Sum_max” value of 400. In addition to that, average of distances of false localization points to the closest correct localization points increases as usable slice numbers and “Sum_max” values decrease meaning that localization information obtained from algorithm is being more and more unreliable.

Table 5.10 Number of Unsuccessful Localizations with Usable Slice Numbers

Usable Slice Numbers	Number Of Unsuccessful Localizations	Number Of Unsuccessful Localizations (%)	Average of Distances of False Localization Points to the Closest Correct Localization Points (units)
15 to 20	6/471	1.27%	1
13 to 14	6/182	3.30%	1
11 to 12	14/125	11.2%	1.07
10	16/75	21.33%	1.34
9	8/82	9.76%	1.53
8	27/84	32.1%	2.23
5 to 7	158/272	58.1%	3.01
3 to 4	62/90	68.9%	5.05
1 to 2	13/13	100%	6.29

Table 5.11 Number of Unsuccessful Localizations with “Sum_max” Value

“Sum_max” value	Number Of Unsuccessful Localizations	Number Of Unsuccessful Localizations (%)	Average of Distance of False Localization Points to the Closest Correct Localization Point (units)
600 to 1543.2 (max. value)	12/786	1.53%	1
500 to 600	20/129	15.50%	1.47
400 to 500	60/161	37.27%	1.75
100 to 400	173/318	54.40%	3.31
0 to 100	None	None	None

Numbers of successful localizations are summed in Table 5.12 for two different methods. Higher success rates are obtained for localizations of natural skylines which are obtained from photographs captured at different environments and times. In addition, localizations of skylines extracted from images which are captured at the same conditions with the map are nearly perfect. Number of successful localizations decrease as occluded versions are included as anticipated since these occlusions are added to skylines for the purpose of examining reliability of localization results.

Table 5.12 Numbers of Successful Localizations

	Numbers of Successful Localizations (Similarity Comparison)	Numbers of Successful Localizations ("Sum_max" comparison)
All test data	1082/1394 (77.6%)	1129/1394 (81.0%)
Only captured test photographs	377/413 (91.3%)	387/413 (93.7%)
Test photographs captured at the same time with the map	172/172 (100%)	171/172 (99.4%)

So far, while performing localizations, photographs which are captured at the same environment are used for both test and map. An example of a localization of a test photograph which is obtained at the same environment with map is given in Figure 5.42 for "K(Diag1,1,Leaves-Sunrise)" and "K(Map,Leaves-Sunrise)" whereas localization of "G(Diag1,1,Leaves-Afternoon)" with "K(Map,Leaves-Sunrise)" is given in Figure 5.43. Corners of squares indicate map points as given in Figure 5.3. Location of "K(Diag1,1,Leaves-Sunrise)" and "G(Diag1,1,Leaves-Afternoon)" can be observed in Figure 5.5. Figure 5.42 represents a nearly perfect localization with similarity values decrease as the distance between map points and the test point increases which is illustrated with color white being gradually turning to black. Also, the highest similarity value 93.87 is quite close to 100 which shows that a good matching between test and the map point is achieved. In the light of these information, Figure 5.43 represents the opposition of a good localization as expected since “G(Diag1,1,Leaves-Afternoon)” has no similarity to any points of “K(Map,Leaves-Sunrise)”.

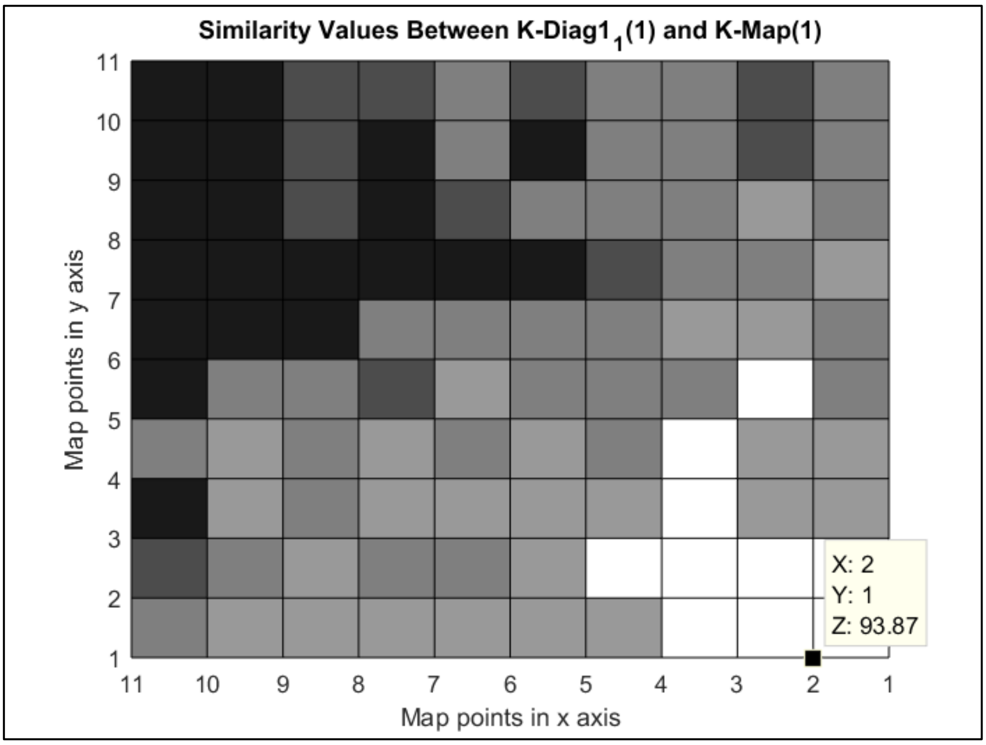


Figure 5.42. Localization of “K(Diag1,1,Leaves-Sunrise)” with “K(Map,Leaves-Sunrise)”

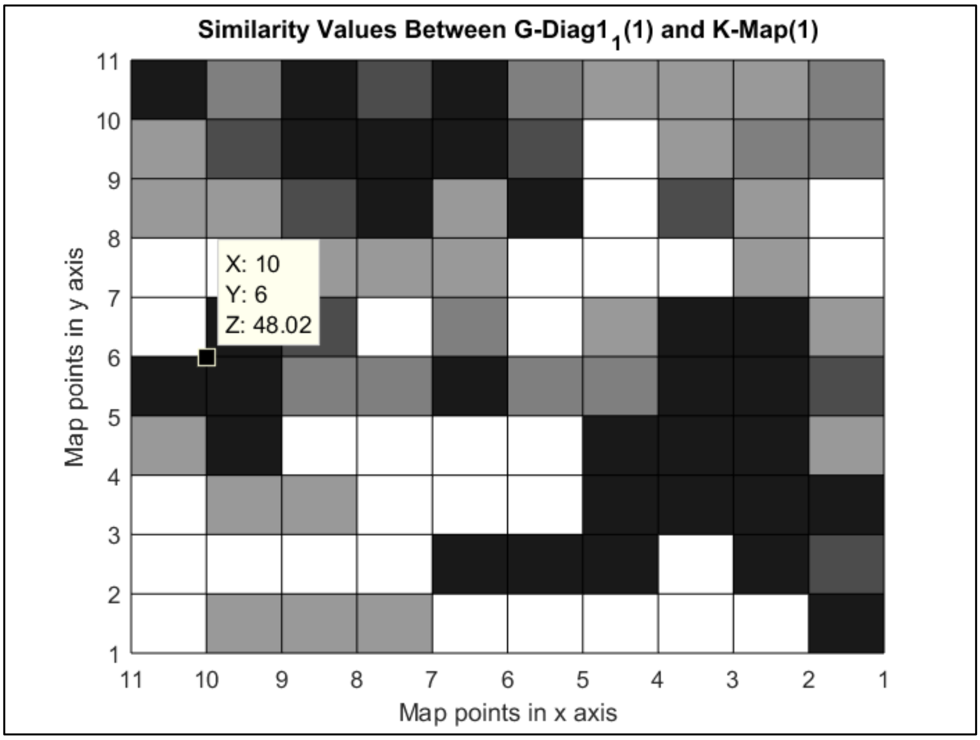


Figure 5.43. Localization of “G(Diag1,1,Leaves-Afternoon)”with “K(Map,Leaves-Sunrise)”

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

There are numerous solutions to localization problem of autonomous territorial robots in the literature. Researches have focused on vision based methods more recently due to affordability of cameras. In this study, a spherical camera is used for localization purposes. Instead of using the panoramic image as a whole, one dimensional skyline signal is extracted with the purpose of obtaining a special feature of a location.

Robot is planned to be dropped off to an environment which it has no prior knowledge. Therefore, it is expected to form a map of its own by taking photographs, extracting skylines and matching those with the position of captured image. As the map is created, robot will become responsible of that area by comparing skylines of an unknown location with already acquired map skyline signals. This process is defined as localization. Two methods are discussed for localization purpose even though one is more explicit and gives more information about the performed localization.

In order to observe performance of localization algorithm, tests are conducted in five different environments at different times of a day and different seasons. As a result, for 2 meters x 2 meters localization in a 11 meters x 11 meters area, 91.3% and 93.7% success rates are obtained with two methods. The rates are 100% and 99.4% respectively for two methods when the test photographs are captured at the same time with the map.

In addition to naturally obtained skylines, algorithm is tested on skylines with artificially occlusions with the purpose of investigation of failure point of matching algorithm. As expected, false localizations are obtained as the percentage of occlusions are increased on a skyline.

6.2 Future Work

In the light of localization results, the method for localization of a robot using skylines explained in this thesis gives promising results although there are some improvements that can be made.

First of all, optimization of five parameters used for similarity finding algorithm can be performed in a more comprehensive way by defining all the variables for parameters. However, if five parameters were to have only three variables, then that means 3^5=243 experiments would have to be conducted with 1394 test signal which makes 338,742 localizations. With current speed of algorithm, which is approximately 3 minutes, that would take 16,937 hours. Therefore, either speed of algorithm must be increased or experiment numbers must be decreased systematically. Aside from this, new similarity measures can be studied which do not have many parameters similar to “Sum_max” method.

Skyline extraction process might be improved to make skylines be unaffected by dark clouds or sun position as discussed in previous sections. With current state of algorithm, a user input is expected in order to feed the algorithm with information of whether the clouds are really dark or not. First of all, darkness of clouds is a bit subjective definition. Secondly, as the algorithm is implemented on a mobile platform, the platform should be able to localize without expecting any user input. Therefore, this part has to be improved. Sun position can directly or indirectly affect skylines. If it is behind trees or any other features, it may cause distortions on skylines due to flare. Another case occurs when the sun hits against some features, especially white buildings. Although image transformation from red-green-blue to hue-saturation-value and saturation multiplication is performed, skylines may still be distorted. Aside from those, snowy cases are not studied during this thesis. If the roofs of buildings and trees are fully covered with snow, then this case would be similar to white building cases if the sun is also present in photographs. If the sun is absent due to weather conditions or early hours, then fully white skyline is likely to be distinguished from sky. Also, foggy environments may be studied as skylines are most likely to be captured piece by piece. Although cloudy environments are studied, rainy environments were not investigated. However, as the algorithm is expected to be implemented on a mobile platform, it is most likely to have rainy weather conditions, too. For such cases, a casing should be designed for spherical camera. Rain droplets on casings should not affect skyline and this must be studied thoroughly. Considering those, as new datasets are obtained at different conditions such as snowy, foggy, rainy, cloudy, sunny, these photographs can be studied with deep learning. Then, without having to convert images back and forward and searching the whole image for the first ground pixel, algorithm would directly give the skyline of a photograph in a more humane way.

S-like shapes due to camera was another problem that is encountered during this study. Although Ricoh Theta S camera has its own built-in gyro sensor, it is observed that due to some unknown issues, sometimes horizon correction is not fully achieved. In order to overcome this situation, an external gyro sensor can be used. Ricoh Theta S camera saves angle information of camera as metadata and by switching that information with the external one could be a successful way to get rid of S-like shapes. Another solution to S-like shapes can be using a stabilization platform.

We considered skylines as 1-D signals and made comparisons by using pixels. However, frequency domain analysis can be studied. Occlusions or distortions due to changing conditions can be observed more clearly on frequency domain.

Although localization results are only given for 2 m x 2 m square areas, 1 m x 1 m or larger area localizations can be investigated. During early stages of algorithm, 1 m x 1 m localization results were not satisfying; however, with the current state of algorithm, successful localizations might be obtained for 1 m x 1 m square area especially for test photographs obtained under the same conditions with the map. Larger area localizations might be more useful as the map area increases.

The algorithm explained in this thesis is designed for outdoor purposes. However, an interpretation for indoor purposes can be established. Since the use of skyline is based on the principle that a certain location has a specific skyline and features on skylines drift apart as the distance between reference points increase, this idea can be implemented to indoor purposes. As the line between ceiling and walls can be used, also an imaginary skyline can be formed with features reflected on walls such as television, tables, armchair etc.

There are also some improvements to make in order to use the algorithm on a mobile platform. Initial studies have shown that a simple raspberry pi is not fast enough for outdoor localization purposes. As the use of faster processor is inevitable, certain improvements can be made with algorithm to make it run faster.

- Skyline extraction process might be improved as discussed previously. Let alone deep learning studies, if issues concerning white buildings and dark clouds are eliminated with a simpler study, then image conversion from red-green-blue to hue-saturation-value to again red-green-blue would be eliminated which would result in faster skyline extraction process.
- Resolution of images can be decreased in order to obtain low image widths which would make similarity finding algorithm to give results faster since shifting of a map signal is made twice when finding similarity value of a test signal with a map signal. However, it is important to note that very first studies with decreasing resolution did not give satisfying results.
- A similar way for speeding up the algorithm can be making the shift of map signals not one by one but larger numbers of pixels.
- Experiments were performed on 10 m x 10 m areas for five different environments. If the mobile platform were to work on larger areas, then it is in our best interest to make localizations on a smaller area of the map. Once the first localization is made with the whole map points, then as the mobile

platform moves around, it would have an idea about its current location by using the beforehand localization. Therefore, when making another localization it would be enough to compare the test signal with a less number of map signals.

- Finally, as explained before, test signal is cut to 20 slices and comparison of each slice is made with map signals one by one. This process is suitable for parallelization which would make the algorithm run faster.

REFERENCES

[1] Chiodini, S., Pertile, M., Debei, S., Bramante, L., Ferrentino, E., Villa, A. G. Barrera, M. (2017). Mars rovers localization by matching local horizon to surface digital elevation models. 2017 IEEE International Workshop on Metrology for AeroSpace

[2] Buluswar, S. D., & Draper, B. A. (1998). Color machine vision for autonomous vehicles. Engineering Applications of Artificial Intelligence, 11(2), 245–256.

[3] Forlizzi, J., & DiSalvo, C. (2006). Service robots in the domestic environment. Proceeding of the 1st ACM SIGCHI/SIGART Conference on Human-Robot Interaction - HRI ’06.

[4] Kolski, S., Ferguson, D., Bellino, M., & Siegwart, R. (n.d.). Autonomous Driving in Structured and Unstructured Environments. 2006 IEEE Intelligent Vehicles Symposium.

[5] Aguiar, A. S., dos Santos, F. N., Cunha, J. B., Sobreira, H., & Sousa, A. J. (2020). Localization and Mapping for Robots in Agriculture and Forestry: A Survey. Robotics, 9(4), 97.

[6] Drumheller, M. (1987). Mobile Robot Localization Using Sonar. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-9(2), 325–332.

[7] Garcia-Fidalgo, E., & Ortiz, A. (2015). Vision-based topological mapping and localization methods: A survey. Robotics and Autonomous Systems, 64, 1–20.

[8] Getting, I. (1993). Perspective/navigation-The Global Positioning System. IEEE Spectrum, 30(12), 36–38.

[9] Gothard, B.M., Etersky, R.D., and Ewing, R.E., 1993, “Lessons Learned on a Low-Cost Global Navigation System for the Surrogate Semi-Autonomous Vehicle.” SPIE Proceedings, Vol. 2058, Mobile Robots VIII, pp. 258-269.

[10] Nghia Ho, & Jarvis, R. (2008). Vision based global localisation using a 3D environmental model created by a laser range scanner. 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems.

[11] Zhang, A. M., & Kleeman, L. (2009). Robust Appearance Based Visual Route Following for Navigation in Large-scale Outdoor Environments. The International Journal of Robotics Research, 28(3), 331–356.

[12] Badino, H., Huber, D., & Kanade, T. (2011). Visual topometric localization. 2011 IEEE Intelligent Vehicles Symposium (IV).

[13] Botterill, T., Mills, S., & Green, R. (2010). Bag-of-words-driven, single-camera simultaneous localization and mapping. Journal of Field Robotics, 28(2), 204–226.

[14] Valgren, C., & Lilienthal, A. J. (2007). Sift, surf and seasons: Long-term outdoor localization using local features. European Conference on Mobile Robots (EMCR).

[15] Stein, F., & Medioni, G. (n.d.). Map-based localization using the panoramic horizon. Proceedings 1992 IEEE International Conference on Robotics and Automation.

[16] Saurer, O., Baatz, G., Köser, K., Ladický, L., & Pollefeys, M. (2015). Image Based Geo-localization in the Alps. International Journal of Computer Vision, 116(3), 213–225.

[17] Johns, D., & Dudek, G. (n.d.). Urban Position Estimation from One Dimensional Visual Cues. The 3rd Canadian Conference on Computer and Robot Vision (CRV’06).

[18] Ramalingam, S., Bouaziz, S., Sturm, P., & Brand, M. (2009). Geolocalization using skylines from omni-images. 2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops.

[19] Nüchter, A., Gutev, S., Borrmann, D., & Elseberg, J. (2011). Skyline-based registration of 3D laser scans. Geo-Spatial Information Science, 14(2), 85–90.

[20] Stone, T., Mangan, M., Ardin, P., & Webb, B. (2014). Sky segmentation with ultraviolet images can be used for navigation. In Proceedings Robotics: Science and Systems

[21] Stone, T., Differt, D., Milford, M., & Webb, B. (2016). Skyline-based localisation for aggressively manoeuvring robots using UV sensors and spherical harmonics. 2016 IEEE International Conference on Robotics and Automation (ICRA).

[22] Ho N., Chakravarty P., "Localization on freeways using the horizon line signature", Workshop on visual place recognition in changing environments, in conjunction with ICRA 2014, 8 pp., June 1, 2014, Hong Kong, China

[23] Koku, A . (2019). Panoramik görüntülerde gökyüzü sınır çizgisini kullanarak hareketli robotlar için yer kestirimi . Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi , 34 (3) , 1629-1644

[24] Bonin-Font, F., Ortiz, A., & Oliver, G. (2008). Visual Navigation for Mobile Robots: A Survey. Journal of Intelligent and Robotic Systems, 53(3), 263–296.

[25] Otsu, N. (1979). A Threshold Selection Method from Gray-Level Histograms.

IEEE Transactions on Systems, Man, and Cybernetics, 9(1), 62-66