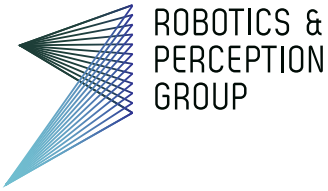
University of Zurich
Department of Informatics



Mathis Kappeler
13-765-482
Zürich Schweiz

Place Recognition and Loop Closure for 3D Mapping

Master Thesis

Robotics and Perception Group
University of Zurich

Supervision

Titus Cieslewski
Prof. Dr. Davide Scaramuzza

Date of Submission: 22th September 2016

Acknowledgement

First of all I would like to thank my supervisor Titus Cieslewski for his support and consulting during my thesis. Throughout this thesis he assisted me with game-changing advices, which made him to an important contributor to this work. Through his rigorous comments on my pull requests, I could improve my coding style and learn a lot about best practices within the given environment. Thanks to Professor Davide Scaramuzza, for letting me be part of the Robotics and Perception Group during both my master project and thesis, where I could utilize the knowledge and experience of experts within the field of research. Finally I want to thank my parents Elisabeth and Peter, for being such a huge support.

Contents

Abstract

Due to drift, visual odometry systems do not provide a globally consistent map [11]. SVO [10] by the RPG is a visual odometry system, specialized to use little computational resources and provides a locally consistent map. A so called SLAM (Simultaneous localization and mapping) system provides a globally consistent map, which is needed to perform more tasks that require metric precision at extended distances. The goal of this thesis is to lay the foundation needed to turn SVO into an online SLAM system while preserving the SVO advantages. We managed to turn SVO into an offline SLAM system, with online potential, by implementing a place recognition and loop closure producing bundle adjustment constraints. We used the bag of words method to perform the place recognition. Furthermore, we evaluated numerous parameters and methods for a future online implementation.

Nomenclature

Notation

A	Matrices are written in capital letters
A^T	The transpose of A
A_{ij}	Cell at i, j of A. Where i is the row index and j the column index
$\vec{v}$	Vectors

Acronyms and Abbreviations

CNN	Convolutional neural network
BoW	Bag-of-Words
MAV	Micro Aerial Vehicle
ROS	Robot Operating System
RPG	Robotics and Perception Group
SLAM	Simultaneous Localization and Mapping
VO	Visual Odometry

Chapter 1

Introduction

1.1 Motivation

Visual Odometry systems accumulate drift over time. This is a result of small computation errors, coupled with the approximation error resulting from linearizations. Due to this drift, a pure odometry system does not provide a consistent global map [11]. To address problems where metric precision is required across extended distances, a globally consistent map is indispensable.

One method to reduce the error drift, without using other sensors but a camera, is to recognize previously visited places, also referred to as place recognition and loop closure [5]. Detecting loops and using the constraints to perform global bundle adjustments will consecutively reduce the error in both the map and the trajectory of the camera, resulting in a globally consistent map. The terms place recognition and loop closure are not distinctively defined in literature. In this thesis we use the term place recognition for detecting a previous observed image scene and loop closure for adding the calculated constraint to the pose graph. Adding place recognition and loop closure to an odometry system turns it into a SLAM (Simultaneous localization and mapping) system. The RPG SVO (Fast semi-direct monocular visual odometry) [10] is a sound VO system and produces a locally consistent map. The objective of this thesis is to implement place recognition and loop closure on top of SVO, to obtain a globally consistent map. The place recognition part of this thesis is inspired the ORB-SLAM [20], which addressed this issue under related constraints.

1.2 Related Work

Both place recognition and loop closure are topics on which a lot of research has been done in the past two to three decades. These subjects are central topics within SLAM, used in robotics. Papers which address these problems are: Fast and incremental method for loop-closure detection using bags of visual words

done by Angli et al. [4], Placeless place-recognition [19], Loop closure detection in SLAM by combining visual and spatial appearance by Ho et al. [17], Robust place recognition for 3D range data based on point features by Steder et al.[25] and Bags of binary words for fast place recognition in image sequences by Gálvez-López et al. [12]. To just mention a few.
The main challenge of this thesis was to find a way to use SVO [10] as a basis and implement place recognition and loop closure on top of it. Our place recognition was inspired by the work of Mur-Artal et al. [20], which addresses the same problem setting in similar circumstances. Mur-Artal et al. are using a Bag of Words approach for place recognition. Another approach is to use a convolutional neural network like Gomez-Ojeda et al. [14]

1.3 Contribution

In this thesis, we were able to successfully implement a robust offline place recognition, loop closure and bundle adjustment, on top of the RPG SVO system [10]. Using the geometric verification detailed in Section 6.2 our method barely yields false positives. Moreover, we were able to find ways to keep the real time advantages of SVO in tact. This means we did not have to increase the number of image features tracked by SVO, to get place recognition results. See Section 5.2 for more details. Furthermore we explored the impact of various parameters and methods on the precision and recall of the place recognition. The contributions provide a basis for a future online implementation.

Chapter 2

Preliminaries

In this chapter we give some basic information about the core methods of this thesis. These methods were implemented on top of SVO [10].

2.1 Place Recognition

Place recognition is is about redetecting previously observed image scenes. It consists of the detection and verification of previous visited places. In Figure 2.1 is an example of a trajectory. If a camera travels along the yellow path, a loop detection system should rerecognize the previously visited places within the red square.

Figure 2.1: Google Maps image from the Malaga 7 dataset.

In this thesis we use place recognition based on images and its features. When an image scene of frame A is recognized in frame B, we obtain a relationship between two frames in the pose graph. This relation between the two frames is displayed as two green dots in the Figure 2.2.
The approach we used in this thesis to rerecognize a place is known as the bag of words method [24].

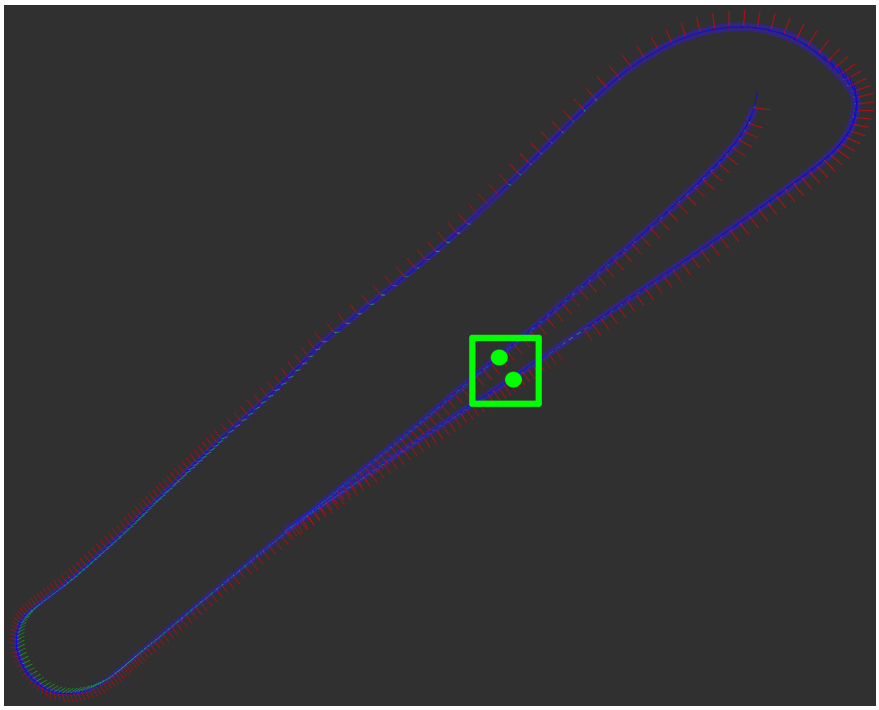


Figure 2.2: SVO pose graph of the Malaga 7 dataset visualized in RVIZ.

2.2 Place Recognition using the Bag of Words Model

One of the first steps of our place recognition pipeline is to determine BoW matches. BoW is short for Bag of Words. The BoW model consists of the following components:

- Images
- Features
- Descriptors
- Visual Words
- Vocabulary
- BoW Vector
- Score

The general goal of the Bag of Words Model is to classify images. First visual features are detected within an image [18]. From these features descriptors are extracted. Then the descriptors are organized into so called visual words. A visual word is defined as a subset of the descriptor space, where descriptors are considered to be similar. Words are defined by nearby descriptors taken from large datasets. These nearby regions can be found by using k-means clustering

[16] or similar methods. All the words together represent the vocabulary.
For a given image, the features are detected and described with a suitable description method. In our implementation we use ORB descriptors [22]. With the help of the vocabulary one can look up what word the descriptor corresponds to. For each frame a so called BoW vector is created which is a histogram of the words within one image.
To compare two frames for a potential scene redetection, we use the TF-IDF weighting [24] of their BoW vectors. TF-IDF is a product of the word frequency and the inverse document frequency. The similarity score of a frame pair is high when the TF-IDF weights correspond to each other.
Like ORB-SLAM [20] we use a so called reference score, which is the minimum score obtained from consecutive frames observing at least one common landmark. This reference score acts as a flexible threshold. In our implementation, the score between a potential match needs to be above the reference score. The benefit of using a reference score is the threshold is adopted to a certain image scene. This way the threshold is customized to the current location.

More information about the Bag of Words Model can be found here: [26].

2.3 Loop Closure

When a scene has been rerecognized, we want to use the resulting constraint in the pose graph to reduce the drift error. This constraint is obtained by calculating the translation and rotation between the two matched frames and adding it to the pose graph. After adding a correct constraint to the pose graph, one can run a global bundle adjustment, which will reduce the overall error of the map and the trajectory of the camera.

Chapter 3

System Overview

In this chapter we give an overview of the different technical components relevant in this thesis, including the components used to test a wide variation of parameter settings. The colors used in the figures of this chapter, illustrate the relation between the components.

3.1 SVO System and Data Extractor

Figure 3.1 illustrates the pipeline of SVO [10]. The Data Extractor represented in blue color, is used to extract the needed information form SVO for further evaluation, such as:

- The frame images
- The timestamp of the image
- The frame id
- The feature coordinates
- The track id of each feature, if tracked
- 3D coordinates of the landmarks
- The descriptor of the extracted feature
- The global pose estimate of each frame, with drift

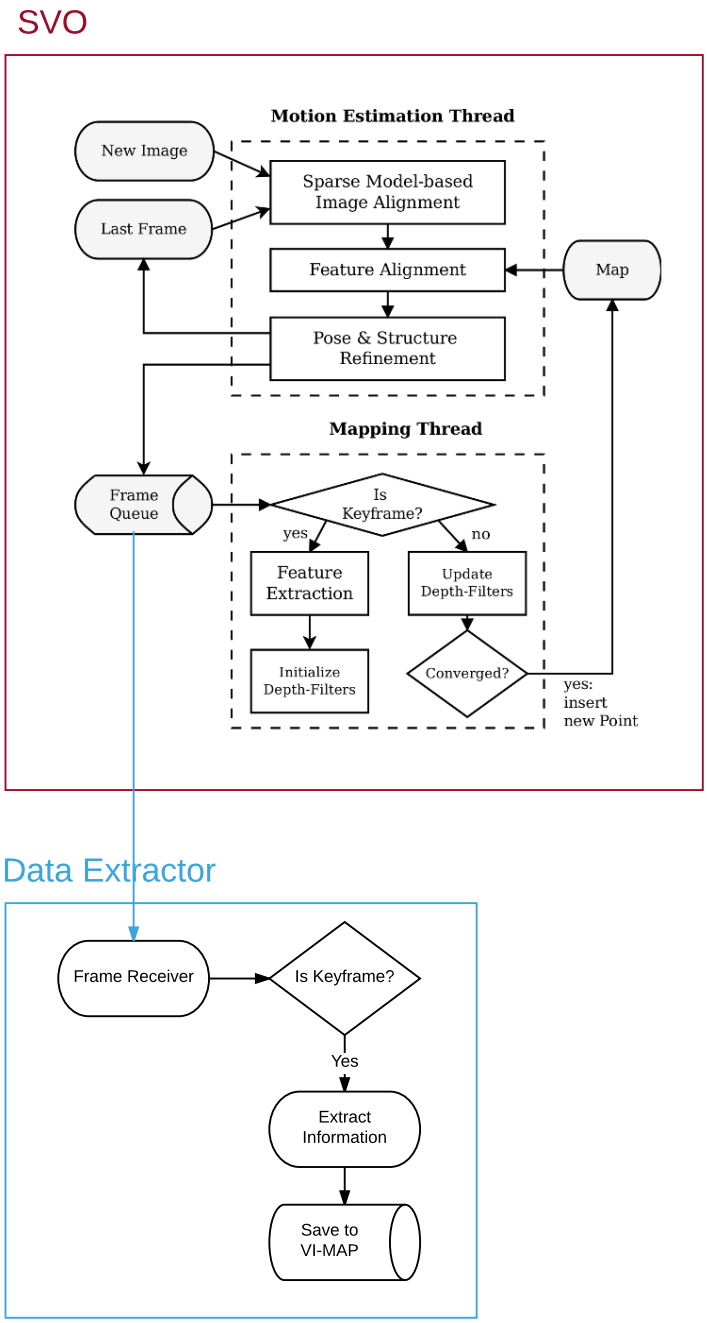


Figure 3.1: SVO Tracking and Mapping Pipeline in the red container (Figure from the original paper) [10]. The data extractor,in the blue container retrieves the data, necessary for place recognition, from SVO.

The information retrived from the data extractor in Figure 3.1 gets stored in the VI-MAP, which is a mapping data structure developed by the ASL group [1]. The VI-MAP is used for further offline processing, see Section 3.2.

3.2 Place Recognition

In Figure 3.2 the process of the offline place recognition is illustrated. Essentially, feature descriptors are matched with the BoW method to other relevant feature descriptors. Frame pairs which contain enough matches are checked for geometric consistency using Ransac, Random sample consensus. Matches which pass Ransac are accepted as a valid place recognition. To see the impact which the geometric verification has on the matching precision, see Subsection 6.2.

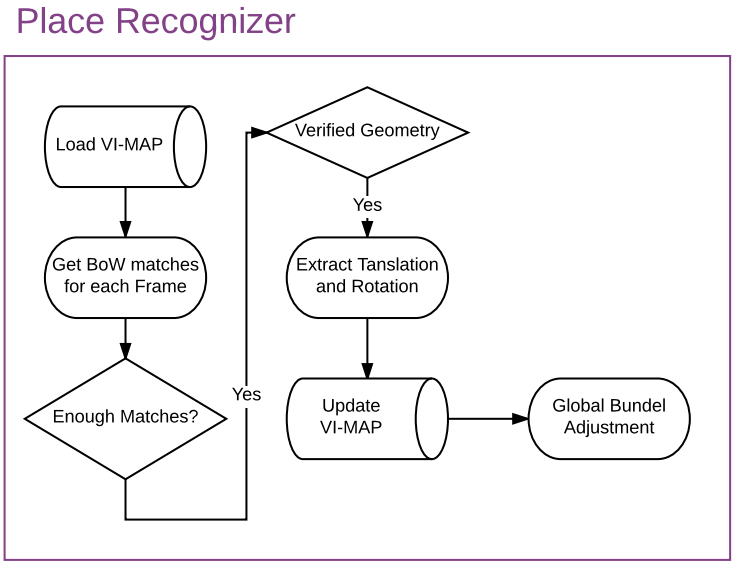


Figure 3.2: This Figure illustrates the logic of the place recognizer implementation. This process is based on the data stored in the VI-Map see Figure 3.1

.

3.3 Evaluation Pipeline

To be able to make a lot of tests with numerous parameters, we built an evaluation pipeline, as can be see in Figure 3.3. This pipeline retrieves the data output from the subprocesses, namely the SVO Data Extractor component in Figure 3.1 and the Place Recognizer component Figure 3.2. The data output is then evaluated and visualized through various plots.

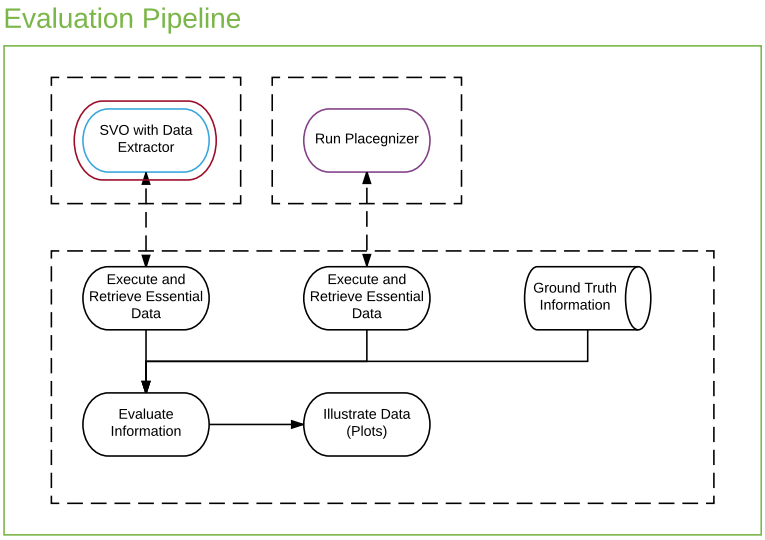


Figure 3.3: The Evaluation Pipeline is used to trigger the components with the given parameter and store the outcomes as plots and other files.

To evaluate and compare the quality of the a method and the set of parameters used, the evaluation pipeline produces performance plots and the ground truth information provided by the datasets. Our evaluation method is explained in more detail in Chapter 4 and the emerged results in Chapter 6.

Chapter 4

Evaluation Methods

In this chapter we discuss how we evaluated the quality of our place recognition. We go through what datasets we use and which we had to dismiss. Finally we explain what methods we use to illustrate our findings.

4.1 Evaluation Pipeline

The evaluation pipeline shown in Figure 3.3 is used to automating the test runs and the following evaluation. Basically, it spawns all the component processes and evaluates the resulting data. This pipeline was crucial, because some of the test series were time intensive. For example, to produce a precision recall curve as described in Subsection 4.5.3, where we varied the tolerance for the geometric verification 6.2, we ran the place recognition up to 30 times on large data sets with up to 3000 keyframes. Having an evaluation pipeline was very convenient and gave us insights hard to get without it.

4.2 Ground Truth for Place Recognition

To evaluate the quality of the place recognition, we picked datasets which contained a position ground truth, see Section 4.3. For each query frame, we need to evaluate if the potential match is a false negative, true negative, false positive or true positive. A common practice to evaluate place recognition, is to use a time and a distance constraint on the frames, extracted from the ground truth data [7].
We illustrate these constraints in Figure 4.1. The green line segments represent trajectories from the past, where place recognition is possible, see line B and C in Figure 4.1. The red line segment is the trajectory from the immediate past, see line A in Figure 4.1. In this context, immediate is defined by a certain time duration t_Δ, see Table 4.1. t_Δ prevents matches to the immediate past which should be not considered as a rerecognition. The arrow head represents the current position. The green circle is an introduced radius r_f in which passed

frames need to redetect previous image scenes. Otherwise this will result in a false negative, see line C in Figure 4.1. The blue circle is another introduced radius, see r_o in Table 4.1, in which redetection is optional, see line B in Figure 4.1. Optional frames will be evaluated in favor of the evaluated system.

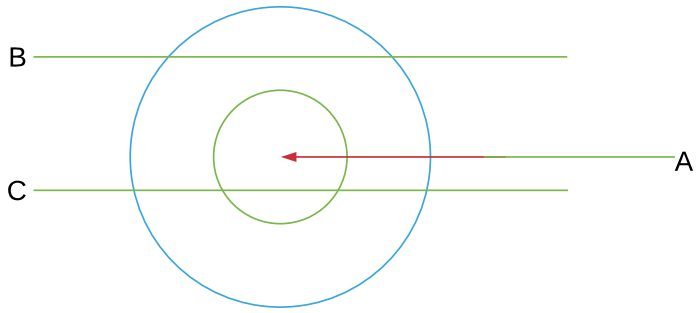


Figure 4.1: In this figure we demonstrate how the ground truth is evaluated. By two distance constraints and one time constraint. Illustrated by the two circles and the lines respectively.

To express this formally we introduce some variables in Table 4.1. Additionally we have the constraints depicted in the Equations 4.1.

Symbol	Description
$\vec{p}_q$	Position of query place
$\vec{p}_d$	Position of database place
t_q	Time of query place
t_d	Tim eof database place
r_f	Forced radius
r_o	Optional radius
t_Δ	Time ignored

Table 4.1: This table introduces the variables used within Section 4.2.

$$\begin{aligned} \vec{p}_q &\neq \vec{p}_d \\ r_f &< r_o \\ t_\Delta &> 0 \\ t_q &> t_d + t_\Delta \end{aligned} \tag{4.1}$$

Case	tp	fp	tn	fn
If $\vec{p}_q$ was matched to $\vec{p}_q$				
$\lvert\vec{p}_q - \vec{p}_d\rvert \leq r_o \wedge t_q - t_d > t_\Delta$	1	0	0	0
$\lvert\vec{p}_q - \vec{p}_d\rvert > r_o \vee t_q - t_d < t_\Delta$	0	1	0	0
If $\vec{p}_q$ did NOT match to any database place				
$\forall\vec{p}$ w.r.t 4.1, $\lvert\vec{p}_q - \vec{p}\rvert > r_f$	0	0	1	0
$\forall\vec{p}$ w.r.t 4.1, $\lvert\vec{p}_q - \vec{p}\rvert \leq r_f$	0	0	0	1

Table 4.2: Here we depict the possible cases of our evaluation. Given the prevailing Constraints (4.1).

The dataset ground truth containes a GPS coordinate for each second. We used a weighted average to approximate the position for the frames in between two ground truth positions. The resulting inaccuracy is neglectable.

4.3 Used Datasets

In general we were looking for datasets which have an adequate frame rate and contain one or more loop closures. Depending on the speed of movement, monocular SVO need a frame rate above 15 to keep track of the image features. Moreover we need a ground truth to be able to evaluate our results. Furthermore we preferred datasets on which similar work has published their results. So we can compare and contrast our results to others.

4.3.1 Malaga 7

Malaga 7 is one of 15 datasets provided in Malaga dataset [6]. The 7th part contains a loop. This dataset was convenient to test our system in the developing phase. Convenient with respect to the manageable size, which made it easier to test changes of our components. We collect the basic information about this dataset in Table 4.3. The trajectory of the path is depicted in Figure 4.2.

Topic	**Value**
Duration	106 s
Distance	$\sim$ 0.7 km
Frame Rate	$\sim$ 20
Description	Around a small avenue.

Table 4.3: Malaga 7 Metrics

Figure 4.2: Trajectory of Malaga 7

4.3.2 Malaga 10

Malaga 10 is a relatively big data set with many loop closures and changing light conditions. We collect the basic information about this dataset in Table 4.4. The trajectory of the path is depicted in Figure 4.3.

Topic	Value
Duration	865 s
Distance	∼ 5.7 km
Frame Rate	∼ 20
Description	Multiple loop closures in a suburb area.

Table 4.4: Malaga 10 Metrics

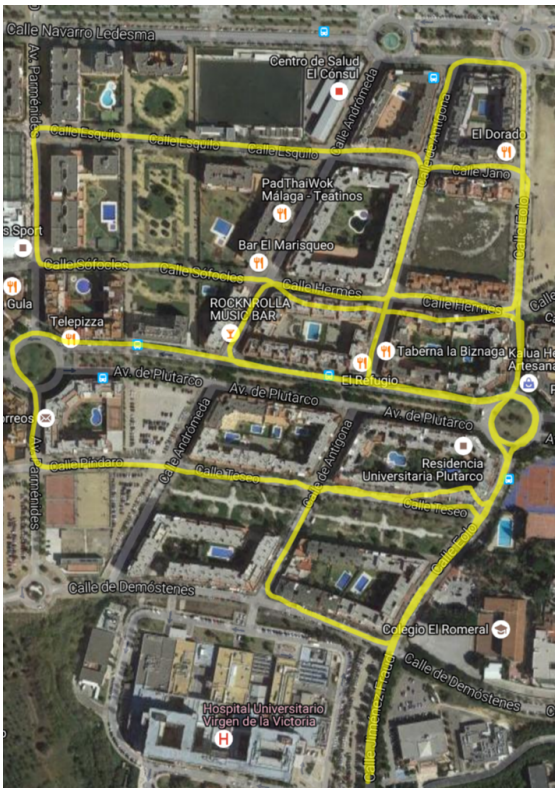


Figure 4.3: Trajectory of Malaga 10

4.3.3 Malaga 6

Malaga 6 is dataset part is similar to Malaga 7, although a slightly longer distance is traveled. Many image scenes in this dataset are similar to each other, even being from two separate places. This can be seen in Figure 4.4, along with camera trajectory. We collect the basic information about this dataset part in Table 4.5.

Topic	Value
Duration	230 s
Distance	∼ 1.2 km
Frame Rate	∼ 20
Description	Around building blocks.

Table 4.5: Malaga 6 Metrics

Figure 4.4: Trajectory of Malaga 6

4.4 Evaluated Datasets

This section is about datasets we evluated, but did not use because of an insufficient frame rate relative to the movement. The problem is that SVO can lose track if it does not redetect enough previous detected features.

- The St. Lucia dataset [2]
- The Kitti dataset [13]

SVO [10] has been run on Kitti in the past, but this only works using the stereo images. Possibly, the St. Lucia dataset would also be trackable using the stereo images. For this Thesis we used SVO in monocular mode only.

4.5 Illustration Methods

In this section we present and explain the used illustration methods. The illustration methods are mainly usend in the Chapter 6 and Chapter 4. Instead of explaining the first plot or each plot throughout this report, we choose to centralize this here and referring to this description throughout the thesis.

4.5.1 Relevant Set

In the relevant set plot we illustrate when the system redetects a place in the upper plot with the title "Proposed Matched". In the lower two plots we com-

pare this to the ground truth matches. Where the "Ground Truth Compulsory Match" are the matches according to the r_f radius described in 4.2. The "Ground Truth Feasible Match" are derived from the r_o radius accordingly.

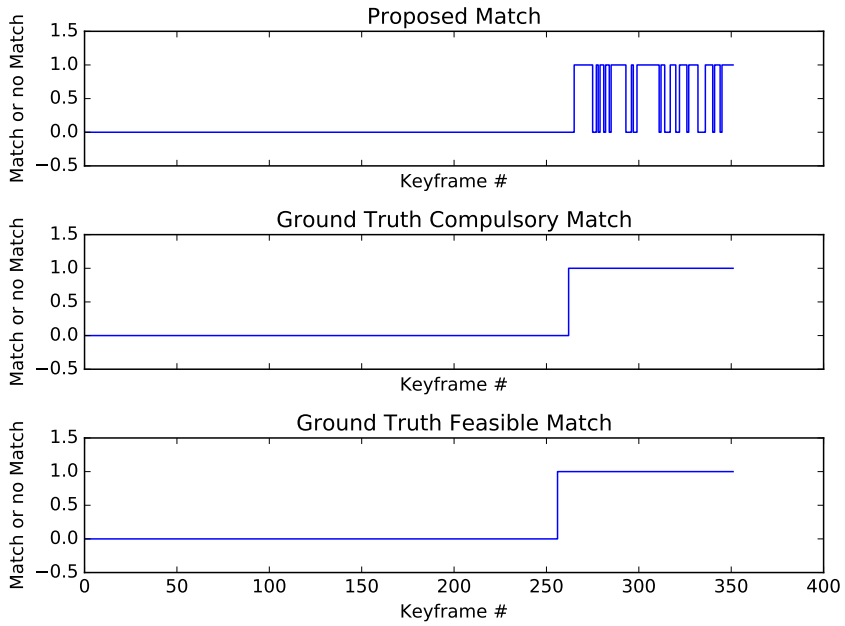


Figure 4.5: Example relevant set plot.

This gives us a visual impression about the quality and quantity of the matches and mismatches. On the x-axis we represent all the keyframes. The y-axis represents a match with the value 1 and no match with the value 0.

4.5.2 Confusion Matrix

A confusion matrix, seen in 4.6, is one way to illustrate what keyframes are matched to which other keyframes. Both x- and y-axis represents the the indexes of the ordered keyframes. A dot in the 2D plane stands for a match between the two corresponding keyframes. We choose only to represent a match from the query frame to the database frame and not vice versa. This is why our confusion matrices are asymmetric.

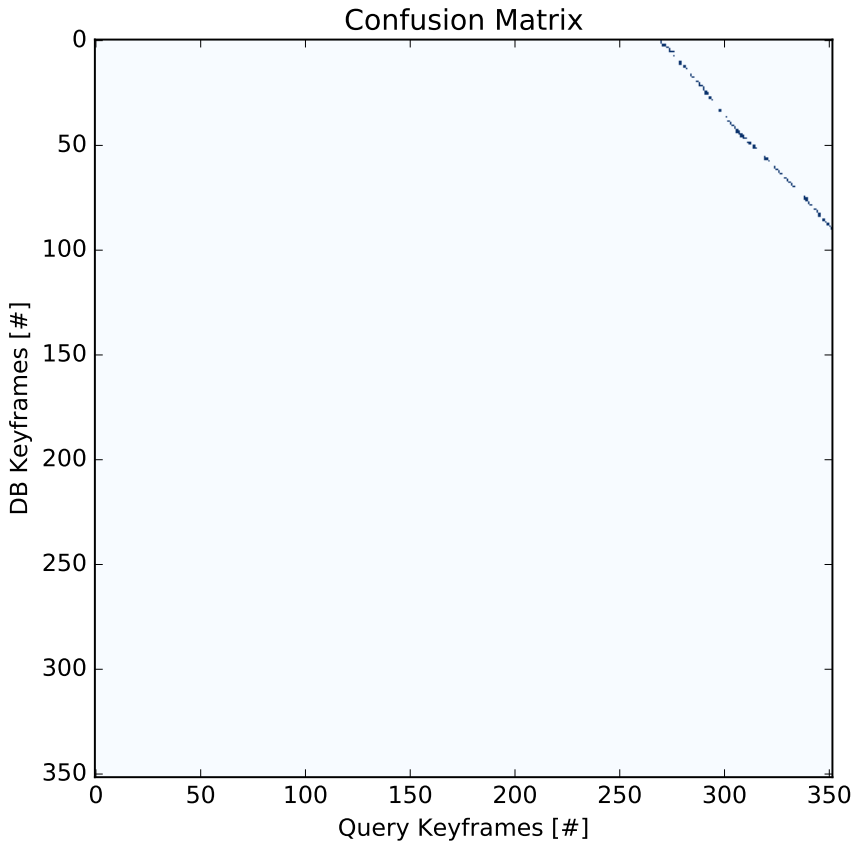


Figure 4.6: Example confusion matrix plot.

4.5.3 Precision Recall Curve

The precision recall curve is a nice way to visualize the trade off between precision and recall based on the choice of one parameter. The precision represents the ratio between true positives and the total detected positives:

$$\frac{tp}{tp + fp} \tag{4.2}$$

The recall is the ratio between the true positives and all the potentially positive matches:

$$\frac{tp}{tp + fn} \tag{4.3}$$

See the sets in Figure 4.7.

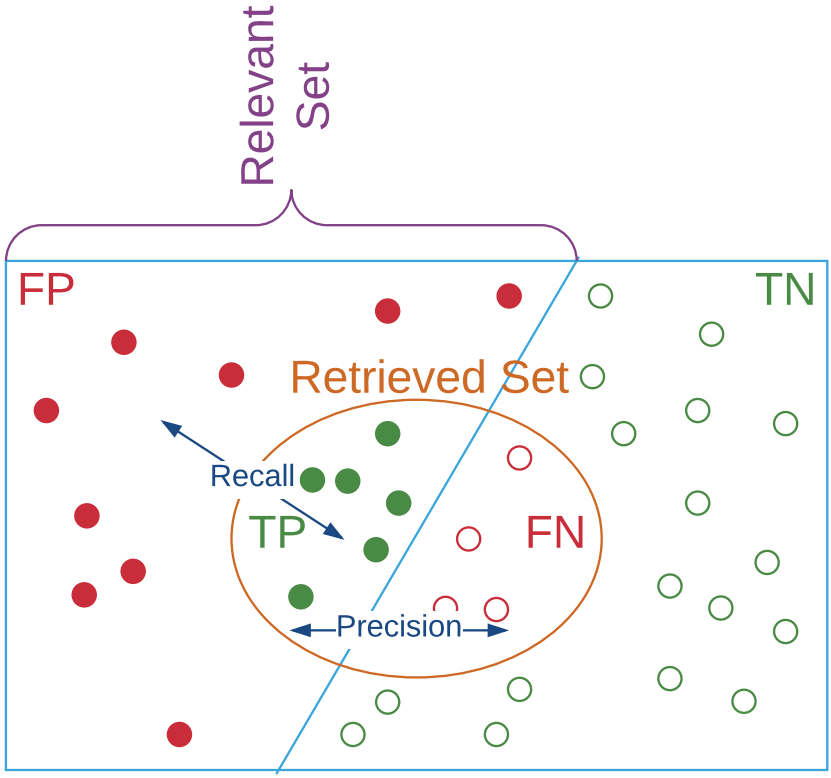


Figure 4.7: Illustration of TP, TN, FN, TN, sets and the sets used for the precision recall curve. Furthermore, we illustrated the relevant and retrieved set.

Let us assume that we change a threshold which determines if a score of a potential match is counted as a match or not. If this threshold is high, the recall value decreases but the precision increases. With a precision recall curve over various tolerances, one can find a well suited tolerance threshold. If we have a look at Figure 4.8, we can see that there is a threshold which leads to a 0.9 precision and a 0.75 recall. Depending on the application this could be a good threshold which results in a high recall and has a decent precision.

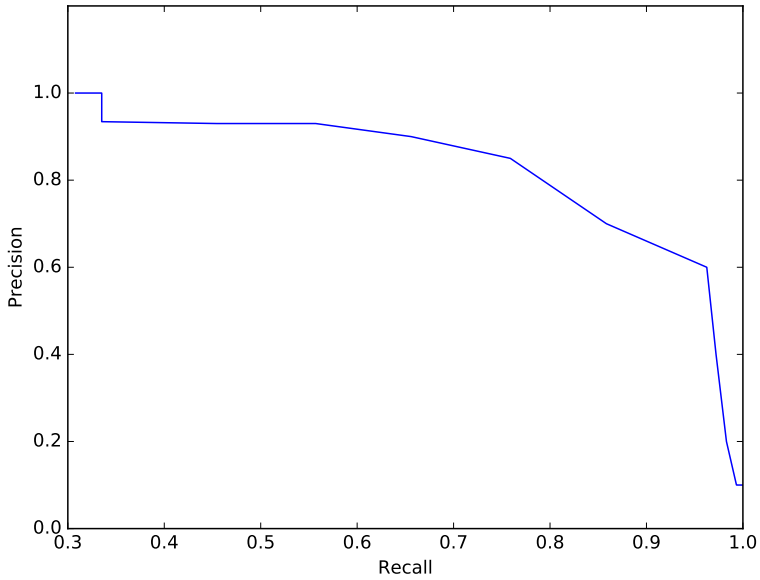


Figure 4.8: Example precision recall curve plot.

4.5.4 Tolerance Recall Curve

After geometric verification the overall precision is usually one or close to one. Consequently, recall is the changing factor in the precision recall curve, see Subsection 4.5.3. Within the process of geometric verification, we use a tolerance which represents the square of the pixel radius in which the projection needs to be, to count as an inlier. We write more about the tolerance in Subsection 6.3. To illustrate what tolerance leads to what recall we created the tolerance recall curve. See the example Figure 4.9.

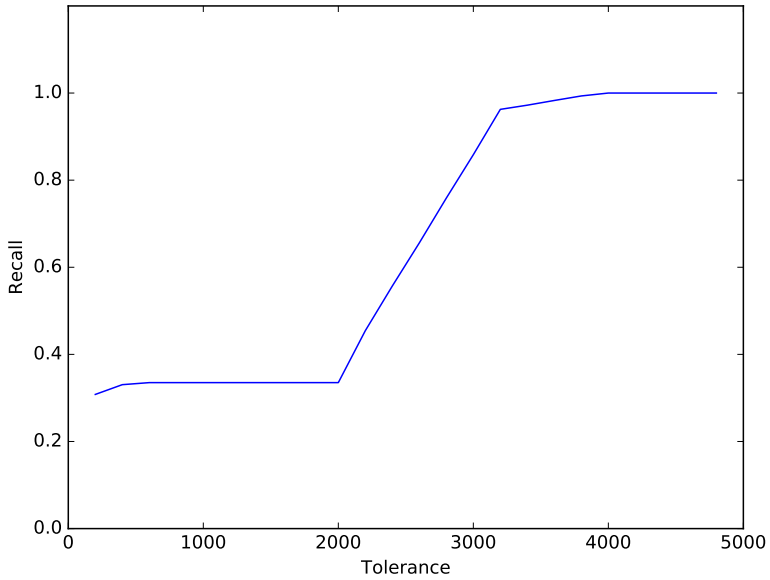


Figure 4.9: Example tolerance recall curve plot.

4.5.5 Orb Vocabulary Scores of Keyframes

Figure 4.10 illustrates different score values for each keyframe. The "best" score is the highest BoW vector score of all other keyframes, except the ones which share landmarks with the current one. The "min" score is simply 75% of the best score. At last, the "reference" score is an interesting approach which we adapted from ORB-SLAM [20]. The reference score is the minimum BoW vector score to all the keyframes with shared landmarks. This results in a varying threshold for each frame. The reference score is later used as a threshold for BoW vector matches with this keyframe.

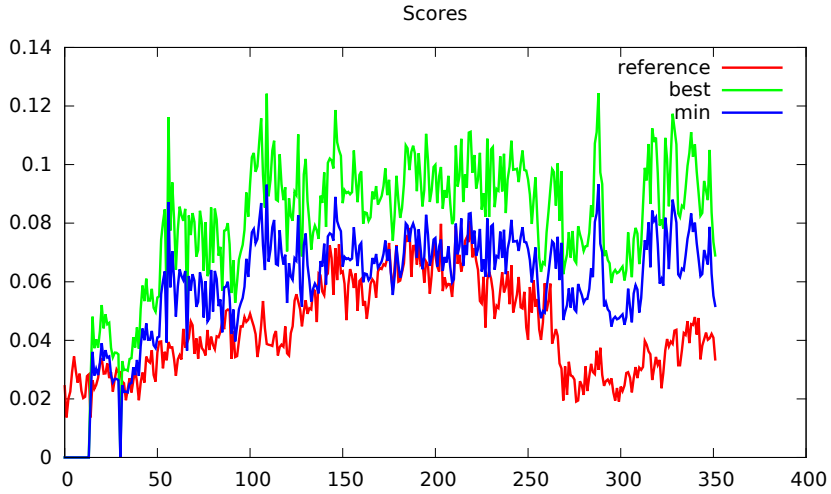


Figure 4.10: Match scores achieved by each query.

4.5.6 Common Word Count of Keyframes

In Figure 4.11 we have the max number of common words, or common ORB descriptors, to one other keyframe. The min number of common words is set to $0.8 * min\_common\_words$.

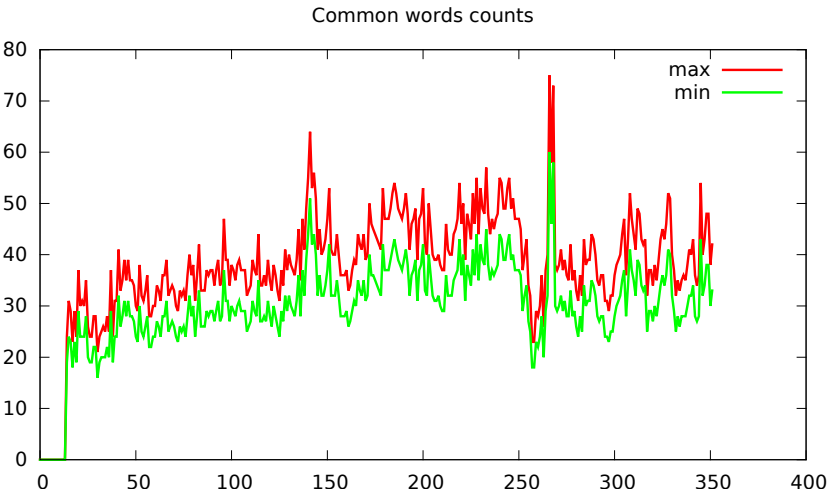


Figure 4.11: Common word count for each keyframe.

4.5.7 Amount of Frames passing hurdles

Figure 4.12 illustrates how many keyframes passes certain loop closure criteria. "Min common words" shows how many frames pass the minimum common words threshold. Respectively the "reference score" and the "min score" shows how many frames passed these thresholds. See Subsection 4.5.5 for more information about the scores.
The "ORB matches" line shows the number of keyframes which have at least shared similar ORB descriptors, which also correspond to 3D landmarks. The highest and most resource consuming criteria, is the geometric verification of the BoW mathces, displayed as "inliers".

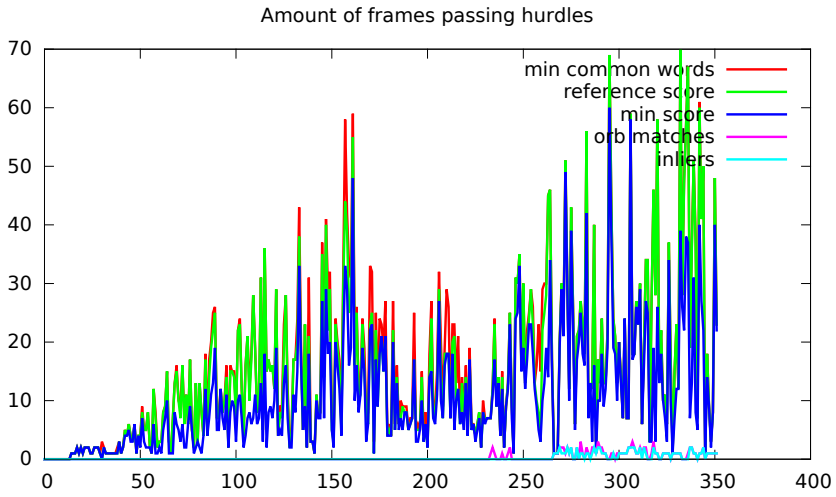


Figure 4.12: Amount of database frames which matches the different criteria for the keyframes.

4.5.8 Feature Observations from Keyframes

In Figure 4.13 we display the observation metrics. "Total" is the number of keypoints in each keyframe. "Associated" represents the keypoints in each keyframe which have been observed by other keyframes, or in other words the number of landmarks in each keyframe. Moreover, "matched" depicts the maximum number of BoW matches for each keyframe. At last "inliers" show the maxiumum number of geometric keypoint inliers of each keyframe. Inlier counts below the threshold are shown as 0.

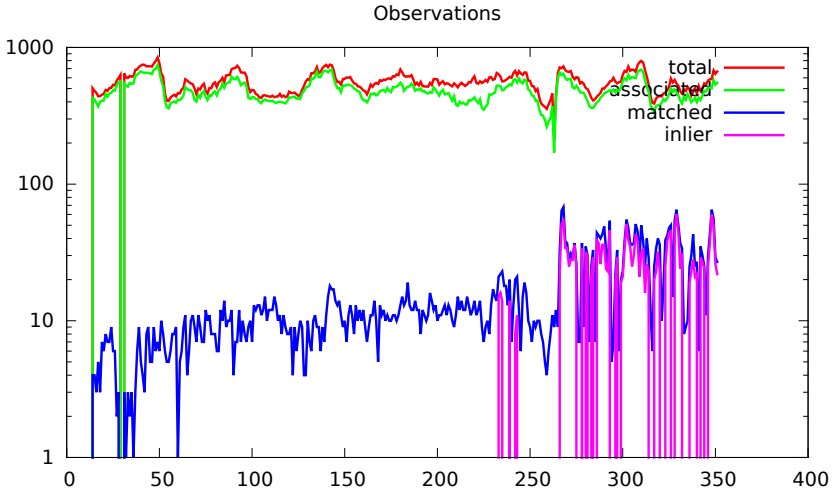


Figure 4.13: Amount of features observed from the keyframes.

Chapter 5

Implemented Approaches

In this chapter we present two implemented approaches. In the first approach we based our place recognition on the landmarks provided by SVO [10]. We needed to adapt the SVO parameters to get enough landmarks to perform place recognition, see Section 5.1.
In our second approach we ran SVO with its default parameter and extracted additional features from the keyframes. Based on these additional features we did place recognition and loop closure with the help of the 3D information provided by the SVO landmarks. See Section 5.2.

5.1 Using SVO Landmarks for Place Recognition

In this approach we use the landmarks of SVO, which are features that have been observed and redetected in several consecutive frames. Based on the multiple observations, SVO calculates the relative 3D positions of the features turning them into so called landmarks.
In this section we will discuss the changes we did in this approach. The results are documented in Chapter 6.

5.1.1 Modification of SVO

Performing place recognition on the SVO landmarks after running SVO with its default parameters, shown in Table 5.2, does not result in any inlier matches. As can be seen in Figure 5.1, no keyframe passes the orb matches hurdle. The plot is explained in more detail in Subsection 4.5.7. The light blue graph represents the frames which were geometrically verified as matches. The reason for the lack of orb matches when running SVO on its default parameters are the number of observations in each frame.

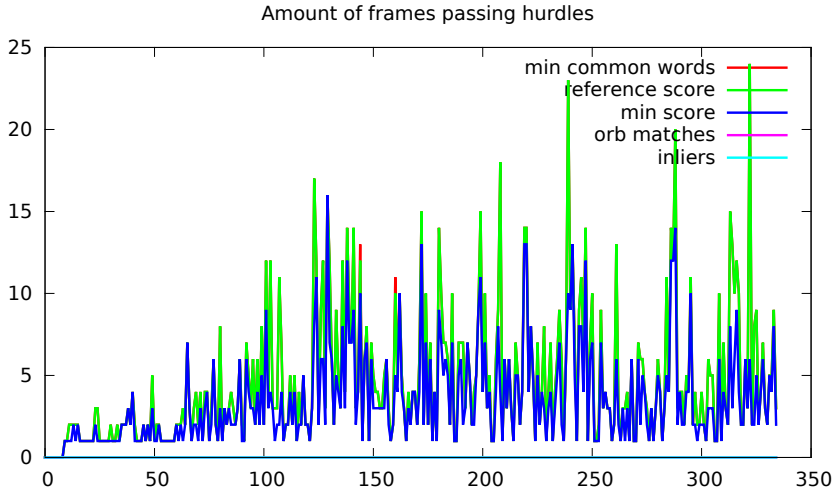


Figure 5.1: Here we see the result of running place recognition on the landmarks extracted from SVO. Running SVO on its default features. No keyframes passes the inlier hurdle, which means we do not get a match.

The default SVO parameters are optimized to perform a robust visual odometry, with as little resources as possible. That is, the numbers of tracked features in SVO are kept as low as possible, to save computational resources. The performance of SVO is one of the main advantages of SVO compared with other visual odometry systems. When it comes to place recognition with BoW, a minimum number of features are needed to redetect a certain scene [8]. With the default parameters of SVO, around 120 features are tracked at all times, this is not enough to reliable redetect enough landmarks for place recognition. Consequently we needed to change the SVO parameters to increase the number of tracked features. Our parameter choices are shown in Table 5.2. They were made based on the criteria explained in Table 5.1.

Parameters	**Description**
grid_size	The frame is divided into patches. This parameter determines the size of these patches. For each patch, the best feature is used by SVO to track the movements. This guarantees a certain distribution of the tracked features which is more robust than having a local heap of features.
kfselect_numkfs_upper_thresh	If at least this number of features are tracked, the current frame will not be considered as a keyframe.
kfselect_min_num_frames_between_kfs	If the last keyframe was selected less than kfselect_min_num_frames_between_kfs frames ago, this frame will not be considered as a keyframe.
kfselect_numkfs_lower_thresh	If less than kfselect_numkfs_lower_thresh features are tracked, the current frame will be marked as a keyframe.
kfselect_min_angle	If the angle of the camera has not changed at least kfselect_min_angle degrees since the last keyframe, the current frame will not be considered as a keyframe.

Table 5.1: In this table we describe the impact of the parameters. The order of the parameter is essential. However there are a few more parameters which determine the selection of a keyframe. If one of the criteria is met the function returns the corresponding boolean value. If none of the kfselect criteria is met a new keyframe is selected.

Parameters	**SVO Values**	**Proposed Values**
grid_size	35	13
kfselect_numkfs_upper_thresh	120	2000
kfselect_min_num_frames_between_kfs	2	5
kfselect_numkfs_lower_thresh	70	500
kfselect_min_angle	20	3

Table 5.2: This table shows the proposed and default SVO parameter values.

The parameters with the prefix *kfselect_* are used to decide if a new keyframe is picked or not. See Table 5.1. With the help of a callback function set in SVO, we extracted all the frames along with its SVO information. Information like

3D coordinates of the landmarks, 2D coordinates of the landmarks projected into the keyframe image. For more information take a look at Section 3.1.

5.1.2 Retrieval of Rotation and Translation

Running 2D to 3D Ransac successfully, the rotation and translation. The scale of the retrieved translation $\vec{t}$ does correspond to the scale used in SVO. The scale is preserved, since the 3D coodinates are given by SVO. Therefore the loop closure constraint can be added to the pose graph, as explained in 2.3.

This method is also used by the ORB-SLAM implementation [20]. Since they perform place recognition on the tracked features, this makes sense.

5.2 Using Additional Keyframe Features for Place Recognition

In Malaga 7 we have 2121 frames, around 350 are getting defined as keyframes by SVO. This means that the additional features, used in the approach discussed in Section 5.1, are tracked throughout all the frames. However, since we do place recognition only on the keyframes we only need the additional landmarks in the keyframes, which is roughly $\frac{1}{6}th$ of all the frames. To not waste that computation power, we came up with an alternative method, where we extracted additional features on the keyframes and used 2D to 2D Ransac to verify the geometry of the BoW matches. Using this approach we can run SVO with default parameters and still redetect image scenes. The results are documented in Chapter 6.

5.2.1 Modification of the Data Extractor

In this approach we run SVO on its default parameters. On the keyframes retrieved by the data extractor 3.1, we perform feature detection and save their descriptors in addition to the landmarks detected by SVO. We use these additional features for place recognition and to perform 2D to 2D Ransac for geometric verification of the matched features.

5.2.2 Retrievement of Rotation, Translation and Scale

After performing BoW matches like we did in the approach discussed in Section 5.1, we perform a 2D to 2D geometric verification. This is a consequence of not having 3D information on the added features, discussed in Subsection 5.2.1.
If Ransac verifies the geometry of the features we can use the fundamental matrix to retrieve the rotation matrix and the translation vector like this:

$$E = K'^{T} F K \tag{5.1}$$

K and K' in (5.1) being the intrinsic calibration matrices of the two images involved. In our case $K = K'$, since the two images where taken by the same camera. E is called the essential matrix, which we will use for further computation.

$$[U, S, V] = SVD(E) \tag{5.2}$$

Where SVD in (5.2) is the singular value decomposition [9]. Now we will define some additional matrices in (5.3):

$$\begin{aligned} B &= \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ L &= U \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} U^T \\ M &= -U \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} U^T \end{aligned} \tag{5.3}$$

Now we can compute two candidates for the rotation matrix R_1 and R_2 and two candidates for the translation vector t_1 and t_2 in (5.4).

$$\begin{aligned} R_1 &= det(UV^T)UBV^T \\ R_2 &= det(UV^T)UB^TV^T \\ \vec{t_1} &= \frac{[L_{32}L_{13}L_{21}]^T}{||[L_{32}L_{13}L_{21}]\,||} \\ \vec{t_2} &= \frac{[M_{32}M_{13}M_{21}]^T}{||[M_{32}M_{13}M_{21}]\,||} \end{aligned} \tag{5.4}$$

Only one of the rotation and one of the translation solutions are feasible. We can find the unique solution, by using the cheirality constraint [21].
The derivation of this is taken from the book Autonomous Mobile Robots by Siegwart et al. [23].

Now we have the rotation and the translation up to a scale. To retrieve the scale used in SVO, we use the 3D landmark informations from SVO. In Figure 5.2 we got two lengths denoted by d_l and d_f. d_f is the distance of an inlier landmark calculated based on the 2D-2D fundamental matrix. d_l is the distance of the landmark given by the SVO coordinate. Now we have the coefficient $\frac{d_l}{d_f}$ used to obtain the approximate SVO scale of the translation vector.

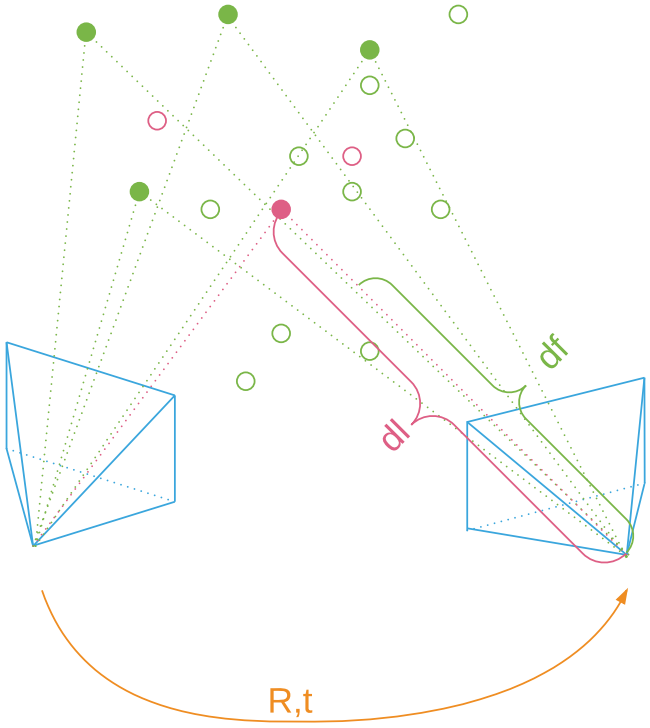


Figure 5.2: Here we illustrate how we retrieve the scale of the translation, by using the 3D information of the SVO inlier landmarks. The filled dots represents the minimum amount of information needed to perform the 5-point algorithm [21] and calculate the fundamental matrix. A higher accuracy is achieved by having more information, displayed as hollow dots. In general the dots represents matched 2D image features. For the red dots we have 3D information, from SVO. The blue pyramids depict the camera and the yellow arrow the rotation and the translation between the two poses.

For each frame tracked by SVO, we have multiple landmarks. So using the average distance of geometric verified landmarks will result in a more robust coefficient:

$$c = \frac{\sum_{i=1}^{n} \frac{d_{l_i}}{d_{f_i}}}{n} \tag{5.5}$$

Now we can use $c\vec{t}$ and R to insert the loop closing constraint in our pose graph. Addressed in Section 2.3.

Chapter 6

Results

In this chapter we compare the results of our approaches. Furthermore we analyze the effects of certain parameters.

6.1 Place Recognition Results

In this section we present and compare the results of our implemented approaches. The first approach was where we run the place recognition on the landmarks of SVO. In the second approach we added additional features to the keyframes. See Chapter 5 for more details about the approaches.

6.1.1 Results in Numbers

The details of the datasets can be found in Section 4.3. The numbers presented in this subsection are the best results achieved by numerous parameter settings. From our perspective the best result is the result where the highest precision was achieved and recall acted as the second criterion.

Malaga 7

Property	**Landmark Approach**	**Added Features Approach**
Keyframes	352	343
Total Positives	82	148
TP	82	148
FP	0	0
Precision	1	1
Recall	0.69	0.96

Table 6.1: Results for the runs on Malaga 7.

Malaga 6

Property	Landmark Approach	Added Features Approach
Keyframes	615	530
Total Positives	0	1034
TP	0	107
FP	0	20
Precision	0	0.84
Recall	0	0.36

Table 6.2: Results for the runs on Malaga 6.

Table 6.2 has a surprising result, the landmark approach does not get any matches. When we compare the candidates of the two approaches, our conclusion is that the Malaga 6 dataset needs way more landmarks than Malaga 7, because there are many similar features within one frame. Even the added feature approach has both lower precision and recall compared with Malaga 7.

Malaga 10

Property	Landmark Approach	Added Features Approach
Keyframes	2738	2427
Total Positives	934	1034
TP	859	960
FP	75	74
Precision	0.92	0.93
Recall	0.36	0.51

Table 6.3: Results for the runs on Malaga 10.

In Table 6.3 one can see that the recall for both approaches is about half of the recall of the recall in Table 6.1. This is caused by place revisitation from the other direction. The image scene do differ a lot, but our evaluation method demands a match.

6.1.2 Relevant Sets

To give a better visual impression of the place recognition, we created the so called relevant set plots. In the following plots you can see the results for Malaga 6, 7 and 10 for both of our approaches. See more details of these plots in Subsection 4.5.1.

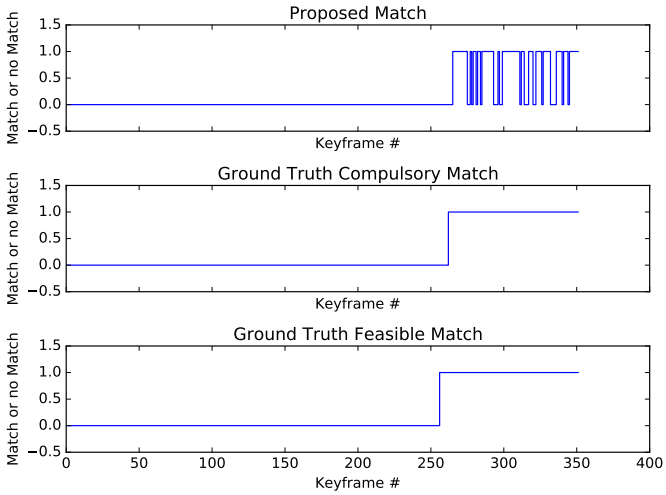


Figure 6.1: Relevant set of Malaga 7, using the landmark approach.

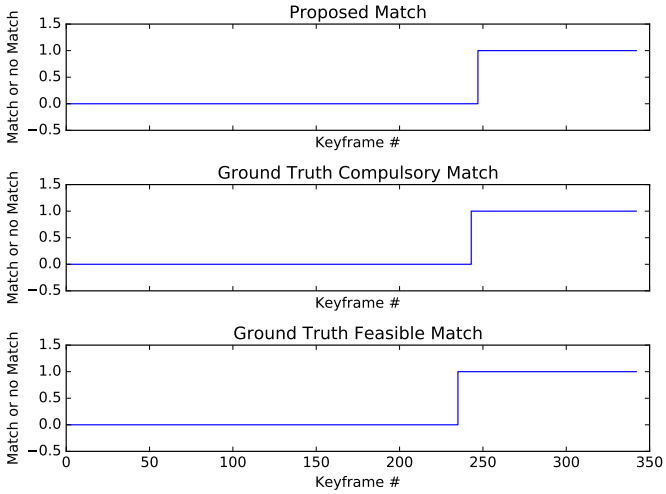


Figure 6.2: Relevant set of Malaga 7, using the additional feature approach.

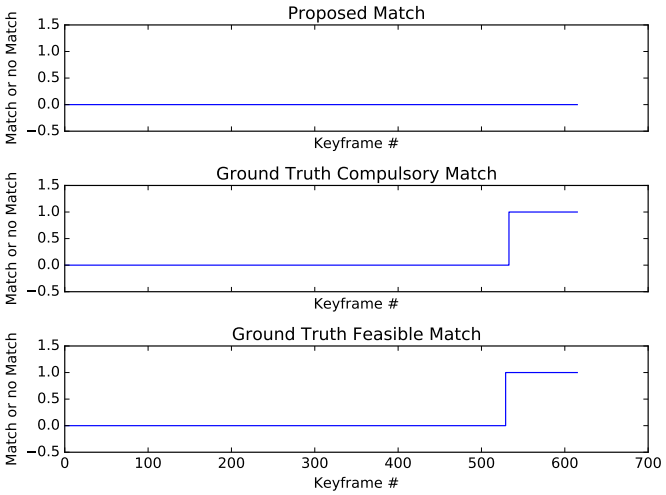


Figure 6.3: Relevant set of Malaga 6, using the landmark approach.

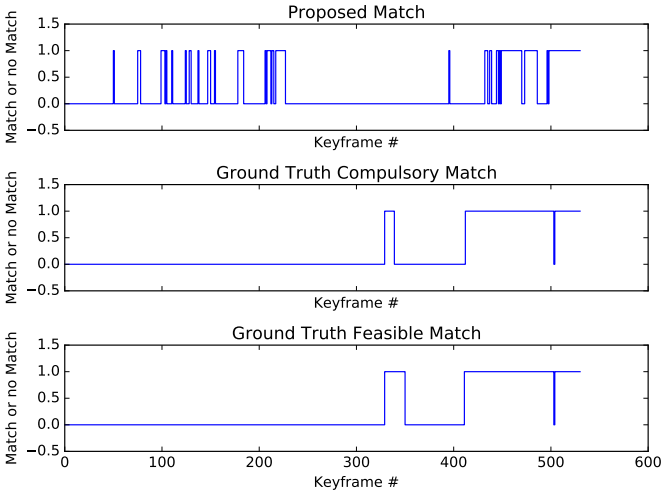


Figure 6.4: Relevant set of Malaga 6, using the additional feature approach.

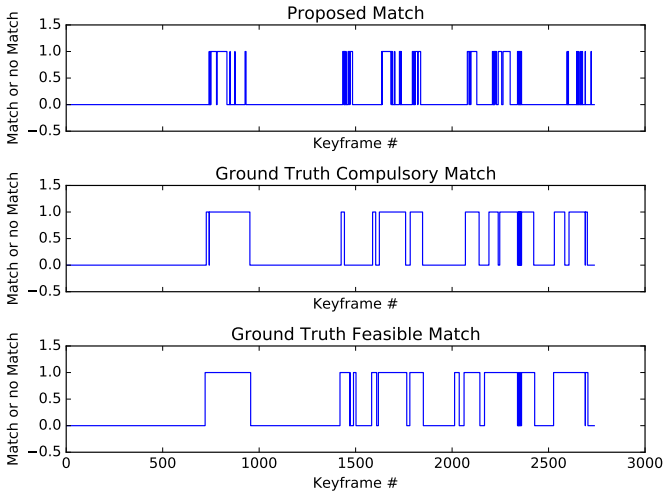


Figure 6.5: Relevant set of Malaga 10, using the landmark approach.

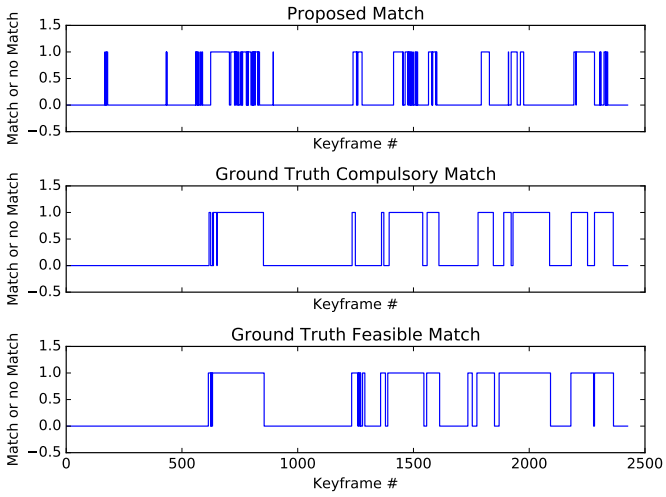


Figure 6.6: Relevant set of Malaga 10, using the additional feature approach.

As we can see in all the Figures 6.1, 6.2, 6.3, 6.4, 6.5 and 6.6, there are almost no false positives. If a wrong place recognition constraint is added to the pose graph, serious inaccuracies could be the consequence.

6.1.3 Confusion Matrix

From confusion matrices seen in the Figures 6.8, 6.10 and 6.12 we can see that the Added Feature Approach gets matches on all the data sets. The precision of the added feature approach used on the Malaga 6 dataset is lower due to very

similar image scenes.
See Subsection 4.5.2 for more detailed information on the confusion matrix plots.

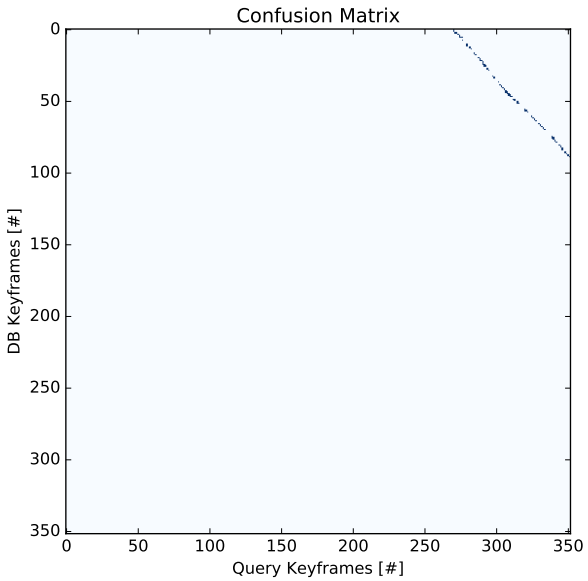


Figure 6.7: Confusion Matrix of Malaga 7, using the landmark approach.

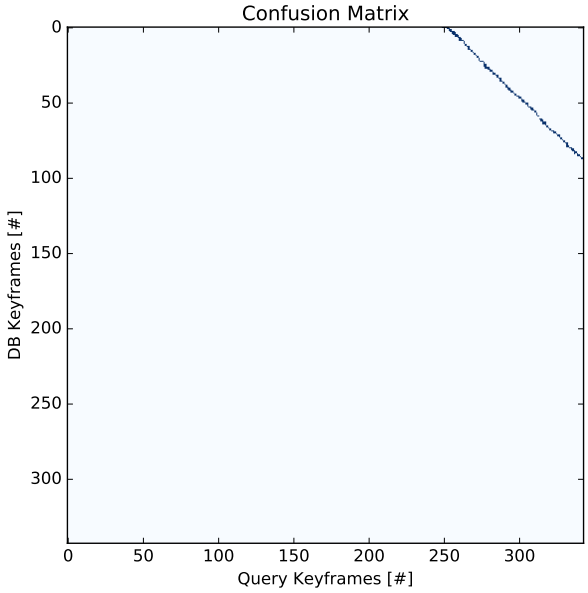


Figure 6.8: Confusion Matrix of Malaga 7, using the additional feature approach.

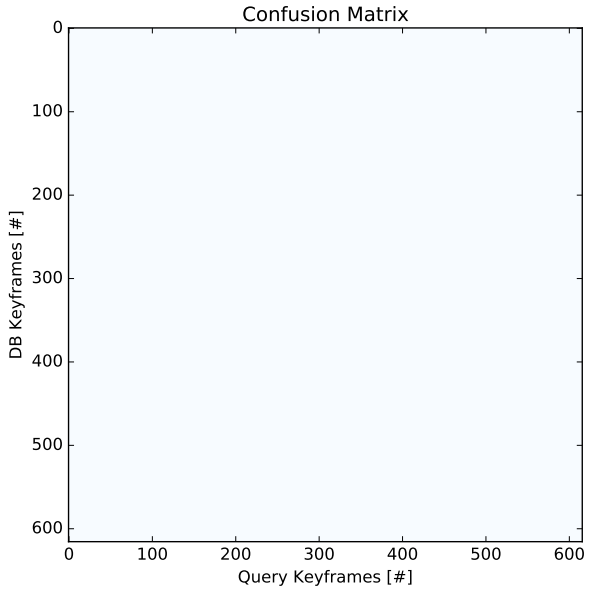


Figure 6.9: Confusion Matrix of Malaga 6, using the landmark approach.

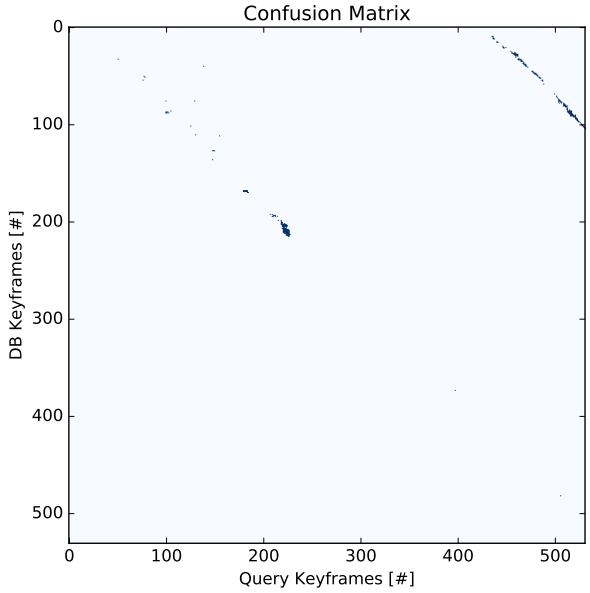


Figure 6.10: Confusion Matrix of Malaga 6, using the additional feature approach.

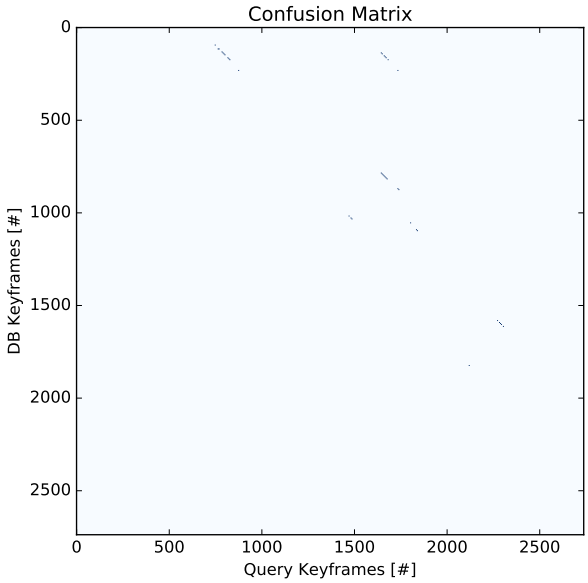


Figure 6.11: Confusion Matrix of Malaga 10, using the landmark approach.

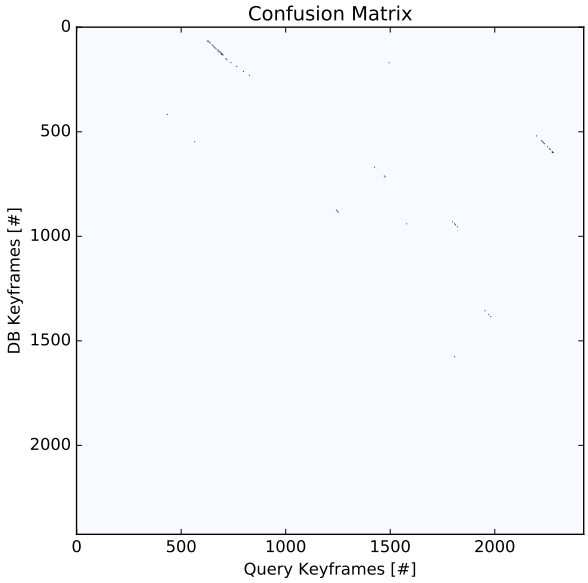


Figure 6.12: Confusion Matrix of Malaga 10, using the additional feature approach.

6.1.4 Observations

The plot details used to illustrate the feature observations is explained in Subsection 4.5.8.

For the additional feature approach we did not extract the number of inliers. This is why there are no inliers illustrated on the plots. However the Ransac probability is set to 0.99 which means for a geometric verified match, a significant amount of inliers need to be given.

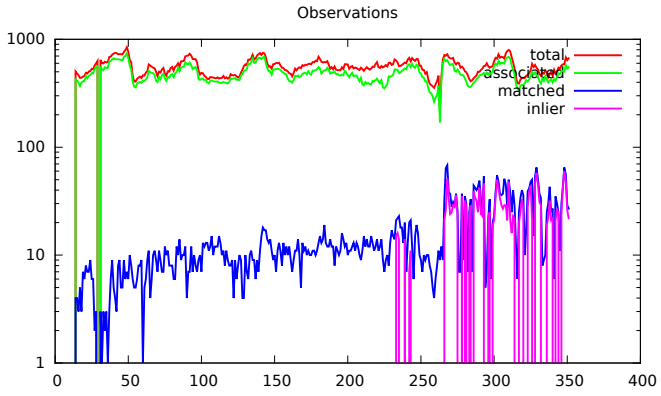


Figure 6.13: Observations of Malaga 7, using the landmark approach.

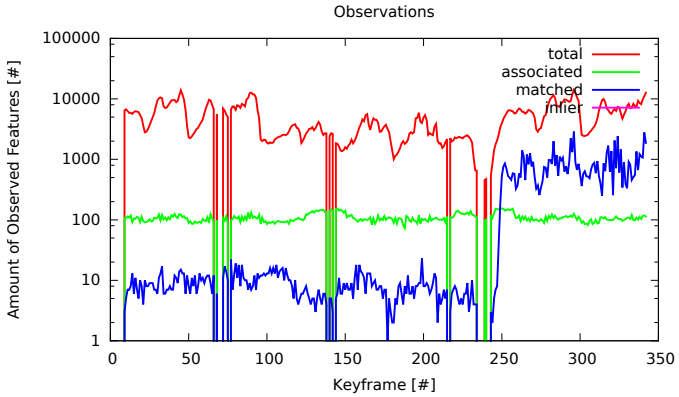


Figure 6.14: Observations of Malaga 7, using the additional feature approach.

In the Figures 6.14, 6.16 and 6.18, where we use the additional feature approach, you can see the big disparity between the total and associated observations. This disparity represents the added features to the keyframes. In contrast to the landmark approach the number of matched features are getting larger than the number of landmarks.

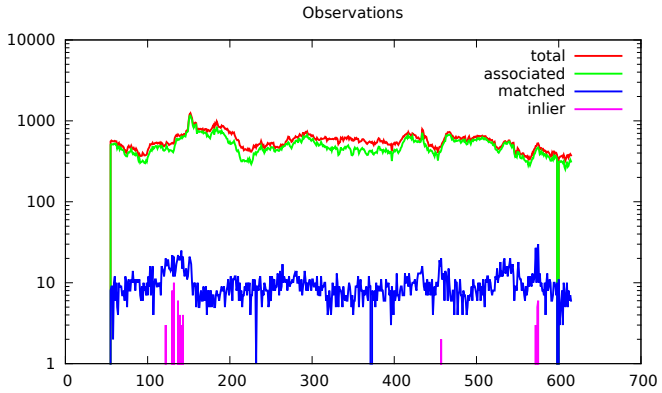


Figure 6.15: Observations of Malaga 6, using the landmark approach.

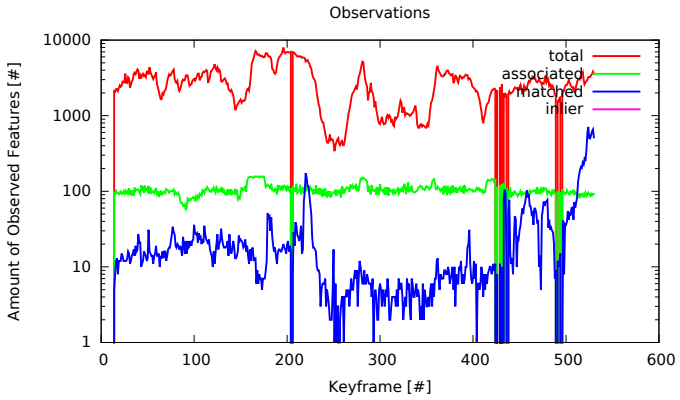


Figure 6.16: Observations of Malaga 6, using the additional feature approach.

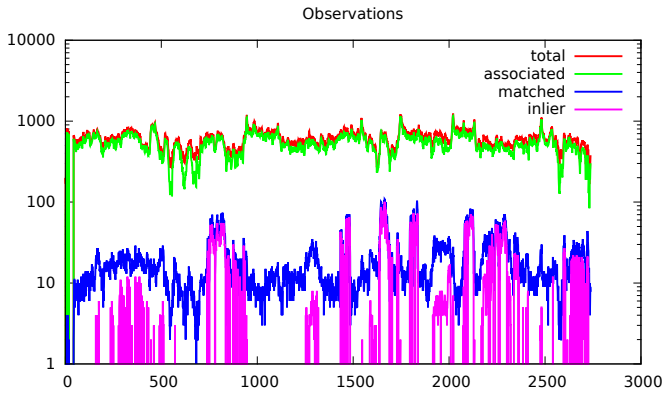


Figure 6.17: Observations of Malaga 10, using the landmark approach.

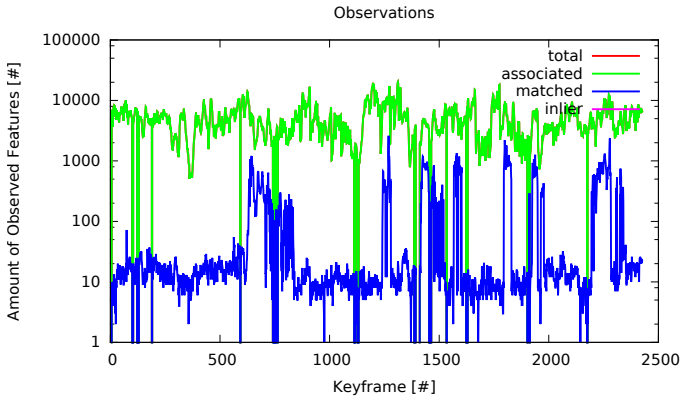


Figure 6.18: Observations of Malaga 10, using the additional feature approach.

6.1.5 Common Words Count

As expected we see larger number of common words for the Figures 6.20, 6.22 and 6.24, where we use the additional feature approach. In contrast we observe smaller numbers of common words in the Figures 6.19, 6.21 and 6.23, where we used the landmark approach.
It is worth pointing out that the common word counts for Figure 6.22, do not have as large spikes as Figure 6.20 and 6.24. This leads to the lower precision and recall values.
For more information on these plots see 4.5.6.

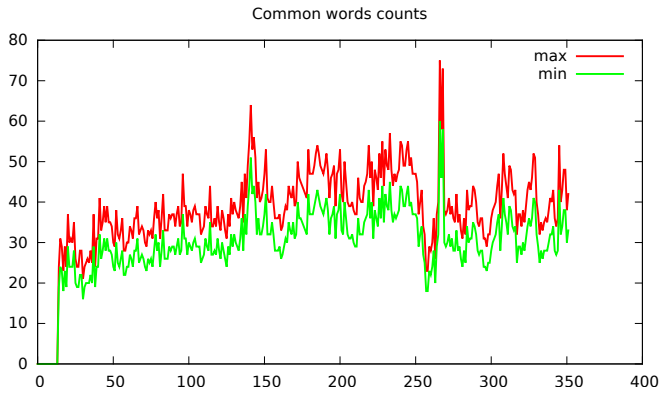


Figure 6.19: Common Words Count of Malaga 7, using the landmark approach.

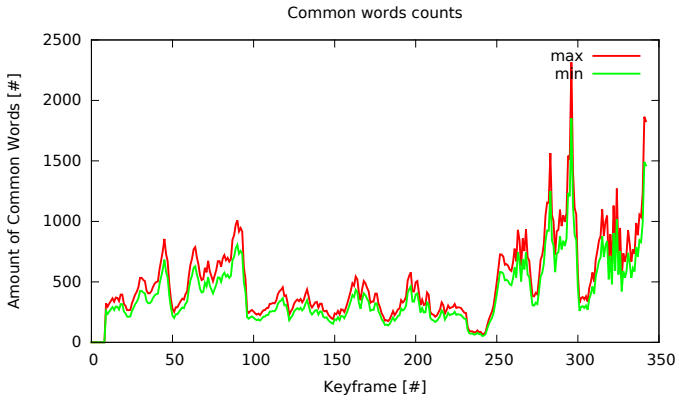


Figure 6.20: Common Words Count of Malaga 7, using the additional feature approach.

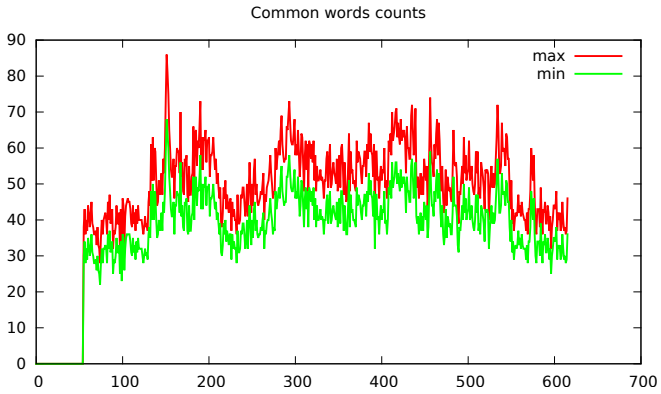


Figure 6.21: Common Words Count of Malaga 6, using the landmark approach.

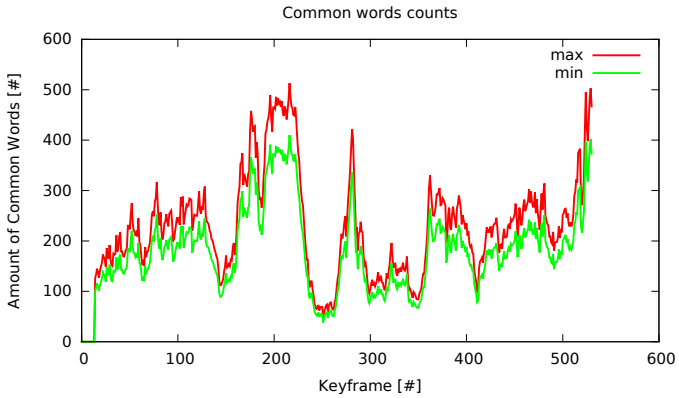


Figure 6.22: Common Words Count of Malaga 6, using the additional feature approach.

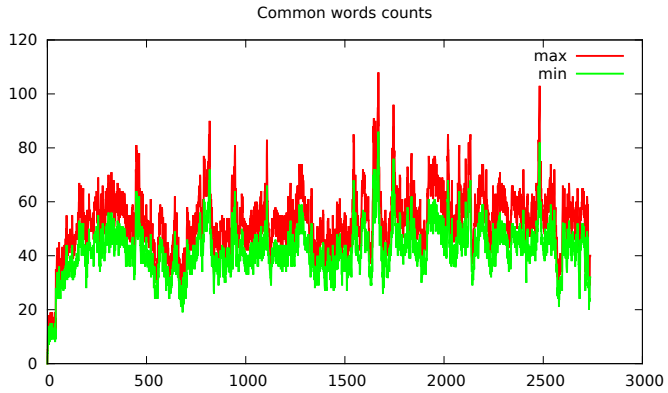


Figure 6.23: Common Words Count of Malaga 10, using the landmark approach.

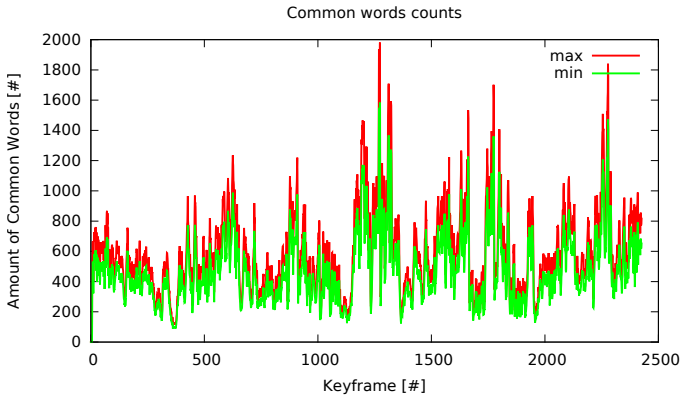


Figure 6.24: Common Words Count of Malaga 10, using the additional feature approach.

6.1.6 Scores

In general we have higher scores where we used the additional feature approach, seen in Figure 6.26, 6.28 and 6.30 than in the landmark approach, seen in Figure 6.25, 6.27 and 6.29. See Subsection 4.5.5, for more details on these plots.

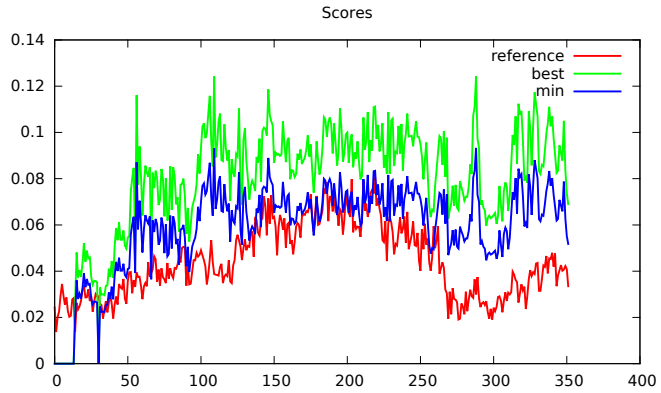


Figure 6.25: Scores of Malaga 7, using the landmark approach.

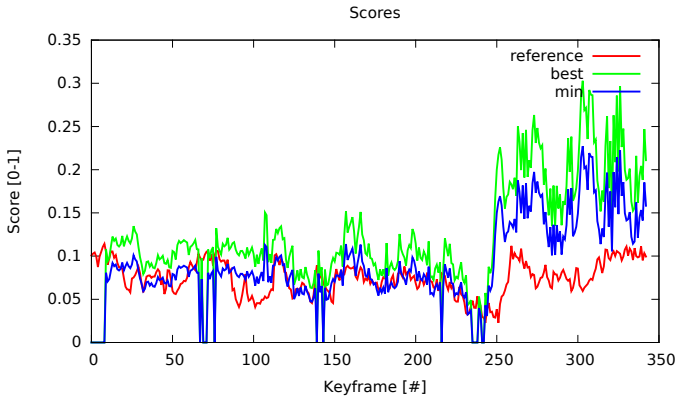


Figure 6.26: Scores of Malaga 7, using the additional feature approach.

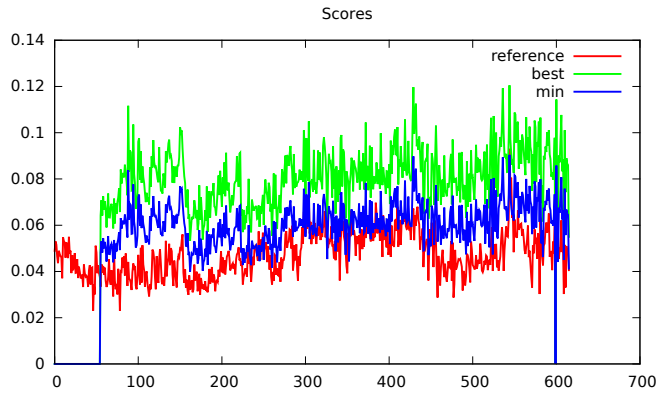


Figure 6.27: Scores of Malaga 6, using the landmark approach.

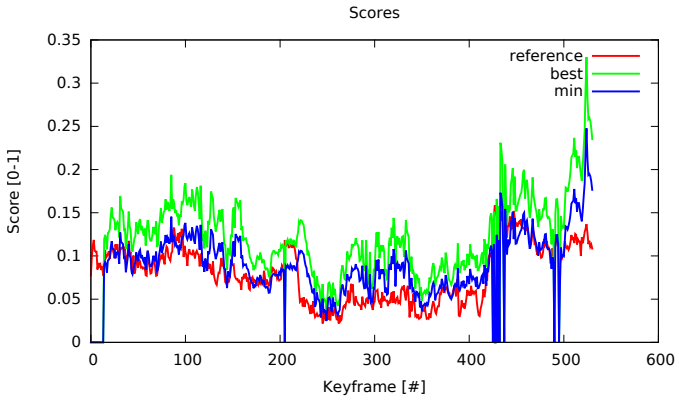


Figure 6.28: Scores of Malaga 6, using the additional feature approach.

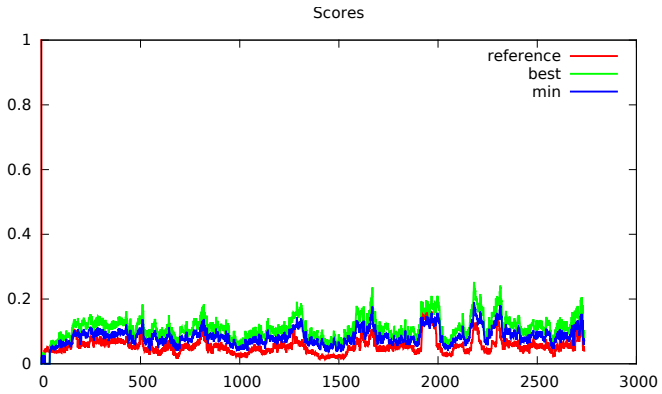


Figure 6.29: Scores of Malaga 10, using the landmark approach.

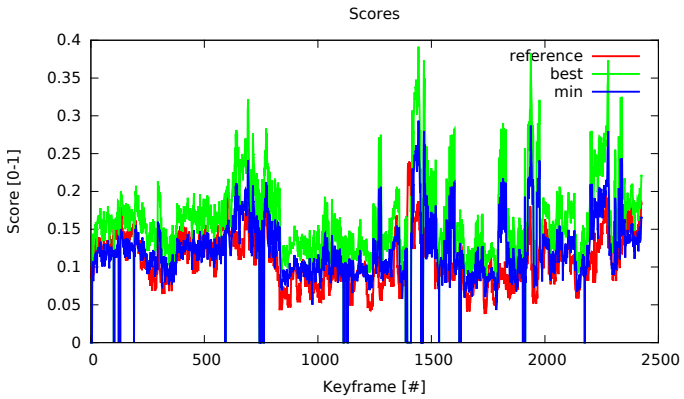


Figure 6.30: Scores of Malaga 10, using the additional feature approach.

6.1.7 Amount of frames passing hurdles

In this subsection we illustrate the Figures 6.31, 6.33, 6.35, 6.32, 6.34 and 6.36, which shows what each keyframe was evaluated to. See Subsection 4.5.7 for more details information on these plots.
Comparing these hurdles to the dataset maps in 4.3, gives you a intuition on why certain values vary at a given keyframe.

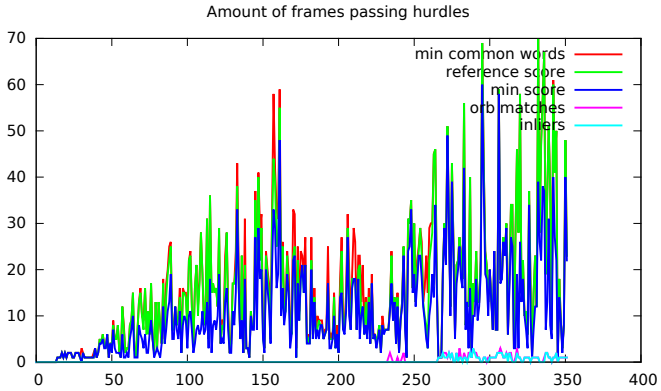


Figure 6.31: Amount of frames passing hurdles of Malaga 7, using the landmark approach.

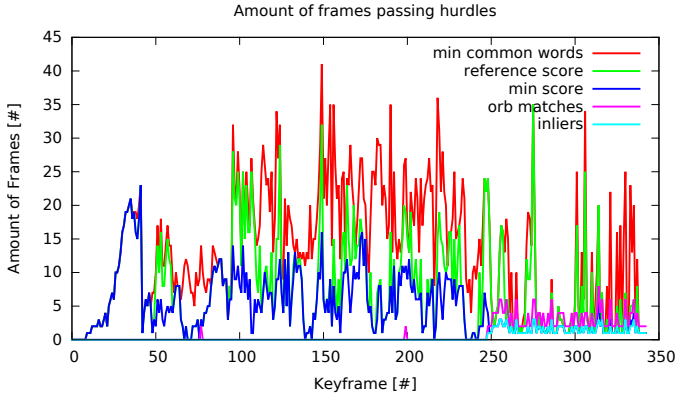


Figure 6.32: Amount of frames passing hurdles of Malaga 7, using the additional feature approach.

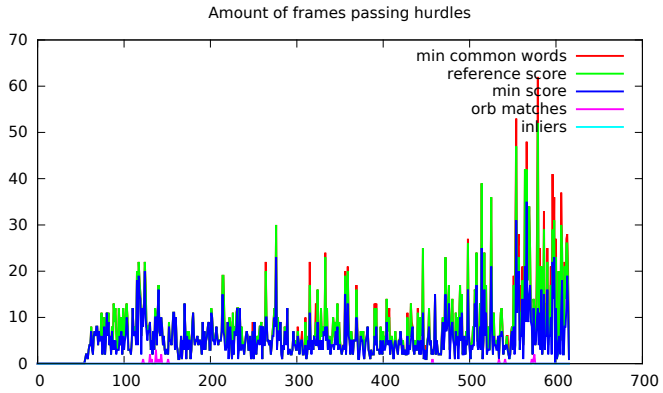


Figure 6.33: Amount of frames passing hurdles of Malaga 6, using the landmark approach.

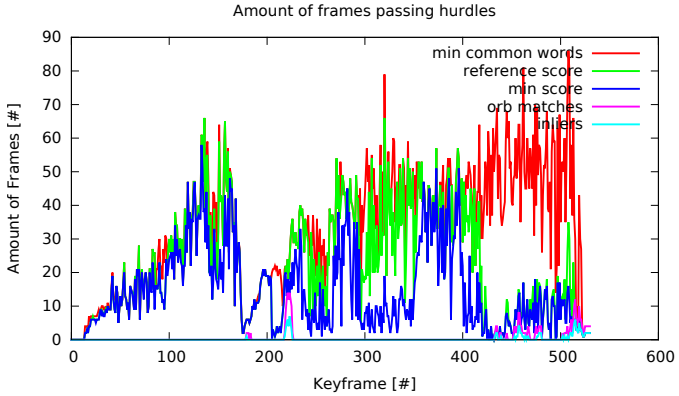


Figure 6.34: Amount of frames passing hurdles of Malaga 6, using the additional feature approach.

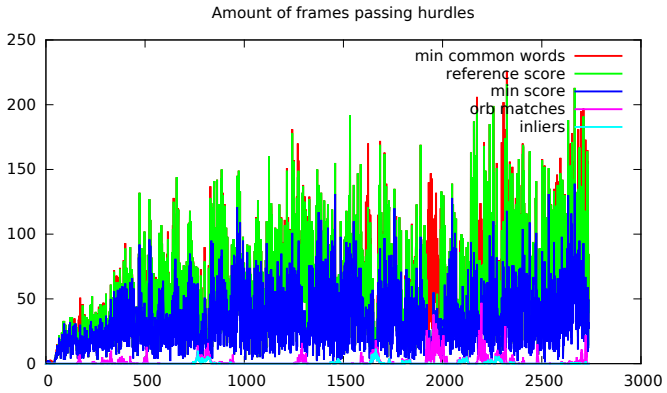


Figure 6.35: Amount of frames passing hurdles of Malaga 10, using the landmark approach.

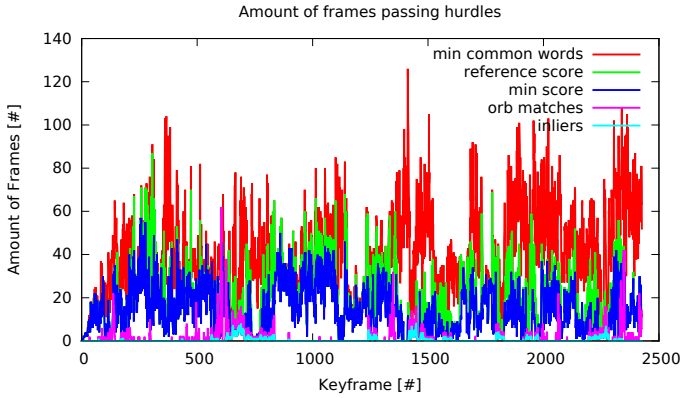


Figure 6.36: Amount of frames passing hurdles of Malaga 10, using the additional feature approach.

6.2 Impact of Geometric Verification

Figure 6.37 represents an example, where two images having a sufficient number of BoW matches. Looking closely at the image pair, we can see that they do not contain the same image scene. Without any further verification this would lead to a false positive. Using geometric verification, where we try to find a translation model of the matched points, we can get rid of almost all the false positives. In Figure 6.38 we illustrate an image pair which contain the same image scene. This BoW match passed the geometric verification in contrast to Figure 6.37.

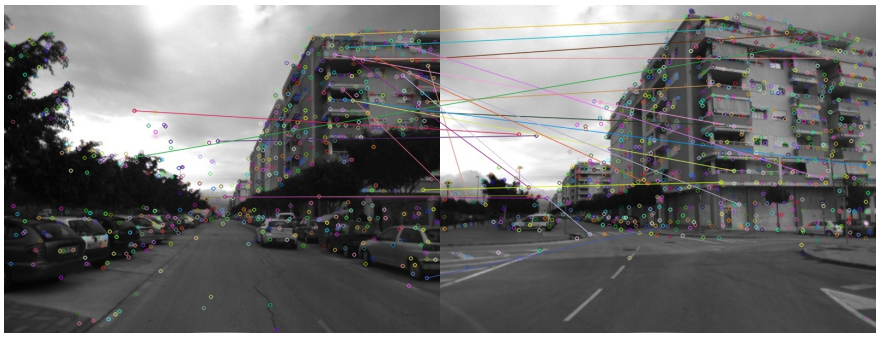


Figure 6.37: An example of a false positive BoW match, which can be detected as an outlier with the help of geometric verification.

Figure 6.38: An example of a true positive BoW match, which gets approved by geometric verification.

On all runs throughout all datasets, see Section 4.3, there were less than 0.25 of the geometric verified matches which were false positives. In Figure 6.39 you can see an example of a run where we did not used geometric verification. Some of the the proposed matches are false. See Subsection 4.5.1 to read more about the details of this plot.

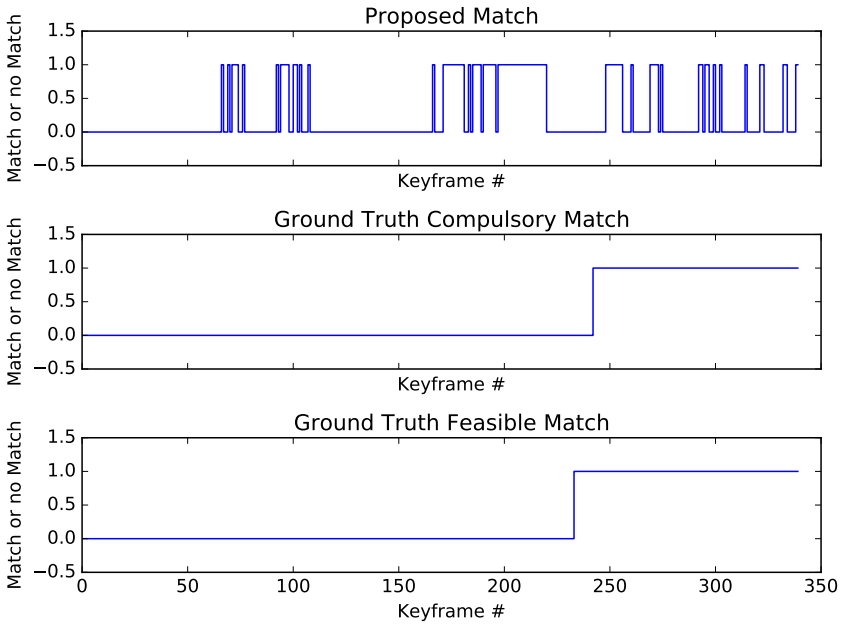


Figure 6.39: Relevant Set without geometric verification.

6.3 Impact of Tolerance used for Geometric Verification

Within the process of the geometric verification 3D to 2D we try to find a feasible translation model. Using this model we can calculate the projection of the query frame 3D features into the database frame and vice versa. If these projections are within a certain tolerance radius of the matched 2D feature we evaluate it as an inlier. We illustrated two different tolerance radii in Figure 6.40 and Figure 6.41. Figure 6.40 and Figure 6.41 has a pixel tolerance radius of $\sqrt{1000}$ and $\sqrt{5000}$ respectively. Within these two figures the colors are essential. The projected and the matched feature do have the same color. The circle represents the tolerance within the projected point has to be, in order to be evaluated as an inlier. When a model has enough inliers, we have verified the geometric correlation between the matched features.
With a bigger tolerance value more matches will pass the geometric verification.

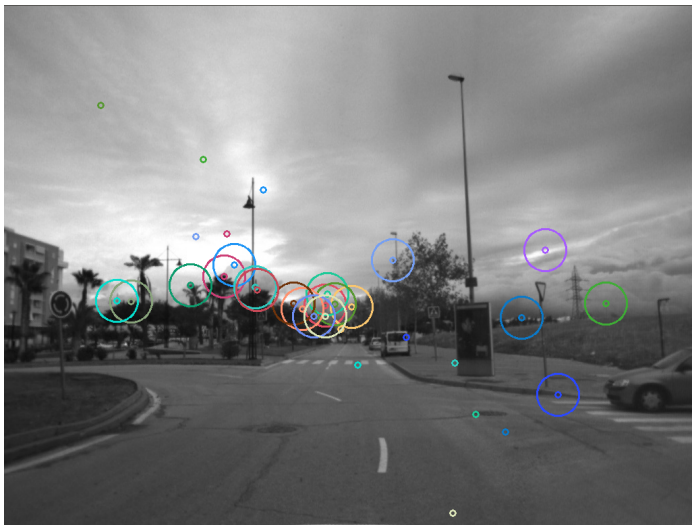


Figure 6.40: Using tolerance radius $\sqrt{1000}$.

Figure 6.41: Using tolerance radius $\sqrt{5000}$.

In Figure 6.42 we see that the precision vary very little, by using different tolerance values. Th recall on the other hand changes from about 0.28 to 0.36. To have a better view on what impact the tolerance has on the recall, we created Figure 6.43.

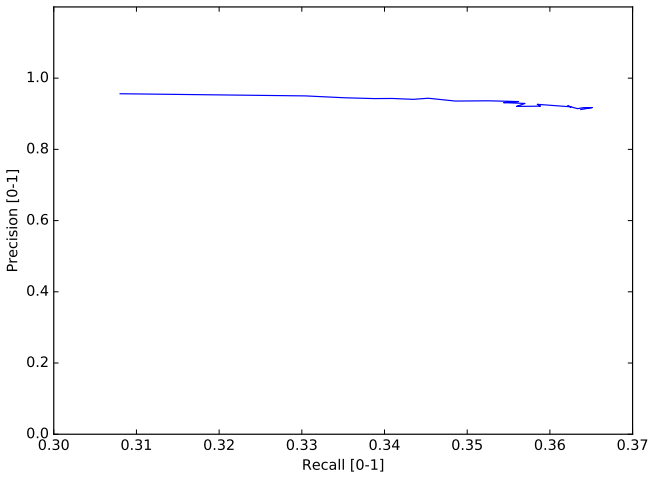


Figure 6.42: This is the precision recall curve resulting from the our landmark approach on Malaga 10. Where we vary the tolerance between $\sqrt{200}$ and $\sqrt{4800}$ pixel.

To evaluate a decent tolerance value, we created a tolerance recall Figure 6.43, explained in detail in Subsection 4.5.4. Based on that Figure we can see that we get the most recall when the tolerance radius is slightly above $\sqrt{2000}$.

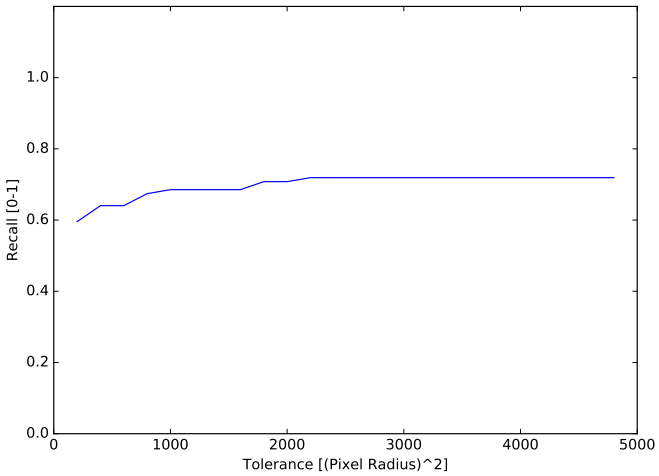


Figure 6.43: Tolerance Recall curve run on the Malaga 7 dataset, see Subsection 4.3.1. The tolerance values of plot are the squared pixel radius

Using a Ransac inlier threshold around 45 leaded to the best results according to Subsection 6.1.1.

6.4 Effect of the Ransac inliers threshold

Looking at the precision recall Figures 6.44 and 6.45, resulting from the added feature approach on Malaga 6. We see that both the precision and the recall value are changing depending on the Ransac Inlier threshold or Ransac threshold.

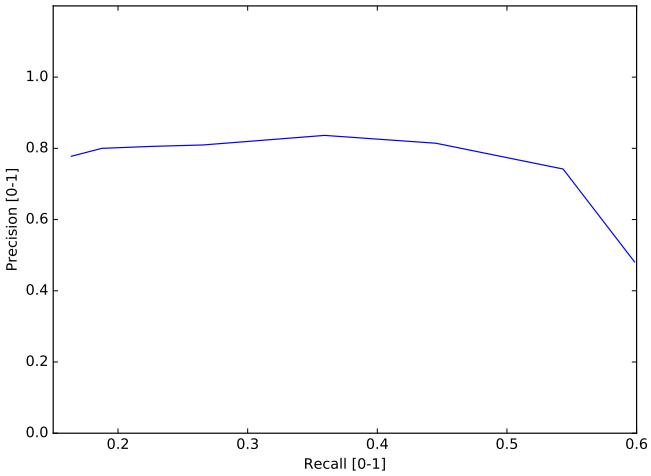


Figure 6.44

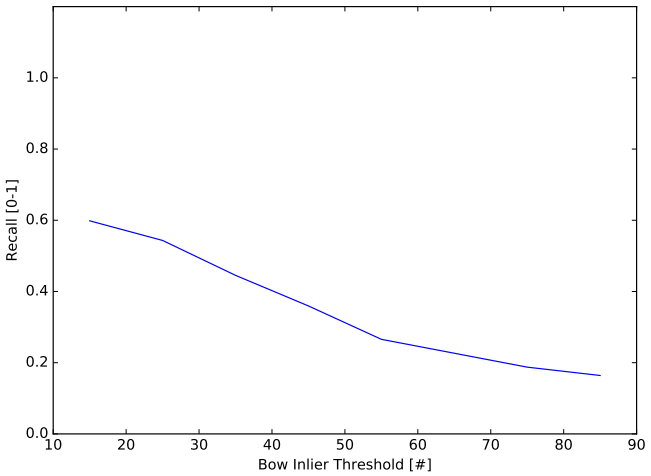


Figure 6.45

Chapter 7

Discussion

In this chapter we summarize the outcome of this thesis and its limitation. Finally we discuss the possible future work based on our effort.

7.1 Conclusion

The main contribution of this thesis is the implementations of an offline place recognition and loop closure system based on SVO [10]. The implementation provides a globally consistent map, which makes it a SLAM system. Our evaluation of the different methods and parameter, pave the way for a future online implementation. A robust visual SLAM with the advantages of SVO, will allow to perform tasks with metric precision. This will in particularly by interesting for long term missions.

7.2 Future Work

The next step will be to take the gained insights, obtained through this thesis and build an online SLAM system. In the online system, the place recognition will be done on the fly. As soon as a place recognition passes all the hurdles, the new constraint will be added to the pose graph. One thread will take care of the global bundle adjustment, reducing the drift in both the map and the camera trajectory. This will require some caution to not end up with race conditions, optimizing the data while SVO uses the data to continuously track the landmarks and updates the pose graph.

In this thesis we used the Bag of Words method for place recognition. It would be very interesting to evaluate how well a convoluted neural network would perform. In terms of precision, recall and the computational resources needed. The results of Gomez-Ojeda et al. [14], show that a CNN approach can be very robust to weather and illumination changes, which would be a huge benefit for

a long term system.

As we have shown in Section 6.2, geometric verification eliminates practically all the false positives. An additional method to reduce the number of false positives we thought about is to use the distance between the camera positions of the query and database frame. Feasible matches are within a certain distance. The place recognition implementation could use this constraint to reduce the search space. This will not only improve the position but save computational resources.

Appendix A

Sources

A.1 Sources

Loop Closure and Evaluation
`https://github.com/uzh-rpg/multiagent_orb`
SVO
`https://github.com/uzh-rpg/rpg_svo_pro`
VI-Map data structure and loop closure
`https://github.com/ethz-asl/multiagent_mapping_basic`

Bibliography

[1] ASL autonomous system lab. `http://www.asl.ethz.ch/`. Accessed: 2016-09-06.

[2] St. Lucia Dataset. `https://wiki.qut.edu.au/display/cyphy/UQ+St+Lucia`. Accessed: 2016-09-09.

[3] St. Lucia Dataset. `https://www.quora.com/What-is-Precision-Recall-PR-curve`. Accessed: 2016-09-18.

[4] Adrien Angeli, David Filliat, Stéphane Doncieux, and Jean-Arcady Meyer. Fast and incremental method for loop-closure detection using bags of visual words. *IEEE Transactions on Robotics*, 24(5):1027–1037, 2008.

[5] Tim Bailey and Hugh Durrant-Whyte. Simultaneous localization and mapping (slam): Part ii. *IEEE Robotics & Automation Magazine*, 13(3):108–117, 2006.

[6] José-Luis Blanco-Claraco, Francisco-Ángel Moreno-Dueñas, and Javier González-Jiménez. The málaga urban dataset: High-rate stereo and lidar in a realistic urban scenario. *The International Journal of Robotics Research*, 33(2):207–214, 2014.

[7] Titus Cieslewski, Elena Stumm, Abel Gawel, Mike Bosse, Simon Lynen, and Roland Siegwart. Point cloud descriptors for place recognition using sparse visual information.

[8] Gabriella Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, volume 1, pages 1–2. Prague, 2004.

[9] L De Lathauwer, B De Moor, J Vandewalle, and Blind Source Separation by Higher-Order. Singular value decomposition. In *Proc. EUSIPCO-94, Edinburgh, Scotland, UK*, volume 1, pages 175–178, 1994.

[10] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. Svo: Fast semi-direct monocular visual odometry. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 15–22. IEEE, 2014.

[11] F Fraundorfer and D Scaramuzza. Visual odometry: Part i: The first 30 years and fundamentals. *IEEE Robotics and Automation Magazine*, 18(4):80–92, 2011.

[12] Dorian Gálvez-López and Juan D Tardos. Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, 28(5):1188–1197, 2012.

[13] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, page 0278364913491297, 2013.

[14] Ruben Gomez-Ojeda, Manuel Lopez-Antequera, Nicolai Petkov, and Javier Gonzalez-Jimenez. Training a convolutional neural network for appearance-invariant place recognition. *arXiv preprint arXiv:1505.07428*, 2015.

[15] Richard W Hamming. Error detecting and error correcting codes. *Bell System technical journal*, 29(2):147–160, 1950.

[16] John A Hartigan and Manchek A Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108, 1979.

[17] Kin Leong Ho and Paul Newman. Loop closure detection in slam by combining visual and spatial appearance. *Robotics and Autonomous Systems*, 54(9):740–749, 2006.

[18] Tony Lindeberg. Feature detection with automatic scale selection. *International journal of computer vision*, 30(2):79–116, 1998.

[19] Simon Lynen, Michael Bosse, Paul Furgale, and Roland Siegwart. Placeless place-recognition. In *2014 2nd International Conference on 3D Vision*, volume 1, pages 303–310. IEEE, 2014.

[20] Raul Mur-Artal, JMM Montiel, and Juan D Tardós. Orb-slam: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

[21] David Nistér. An efficient solution to the five-point relative pose problem. *IEEE transactions on pattern analysis and machine intelligence*, 26(6):756–770, 2004.

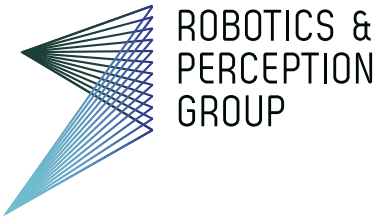
[22] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International conference on computer vision*, pages 2564–2571. IEEE, 2011.

[23] Roland Siegwart, Illah R Nourbakhsh, and Davide Scaramuzza. Autonomous mobile robots. *Massachusetts Institute of Technology*, 2004.

[24] Josef Sivic and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1470–1477. IEEE, 2003.

[25] Bastian Steder, Giorgio Grisetti, and Wolfram Burgard. Robust place recognition for 3d range data based on point features. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 1400–1405. IEEE, 2010.

[26] Yin Zhang, Rong Jin, and Zhi-Hua Zhou. Understanding bag-of-words model: a statistical framework. *International Journal of Machine Learning and Cybernetics*, 1(1-4):43–52, 2010.



Title of work:

Place Recognition and Loop Closure for 3D Mapping

Thesis type and date:

Master Thesis, September 2016

Supervision:

Titus Cieslewski
Prof. Dr. Davide Scaramuzza

Student:

Name:	Mathis Kappeler
E-mail:	mathis.kappeler@uzh.ch
Legi-Nr.:	13-765-482

Statement regarding plagiarism:

By signing this statement, I affirm that I have read the information notice on plagiarism, independently produced this paper, and adhered to the general practice of source citation in this subject-area.

Information notice on plagiarism:

`http://www.lehre.uzh.ch/plagiate/20110314_LK_Plagiarism.pdf`

Zurich, 21. 9. 2016: _____